Performance Comparison of the Parallelized FDTD Scheme with the PML implemented on GPU and MIC Architectures

Ryo Imai, and Yukihisa Suzuki
Department of Electrical and Electronic Engineering
Graduate School of Science and Engineering, Tokyo Metropolitan University
Hachioji-shi, Japan
y_suzuki@tmu.ac.jp

Kan Okubo
Department of Information and Communication Systems
Graduate School of System Design, Tokyo Metropolitan University
Hino-shi, Japan

***Abstract*—In this paper, GPU, MIC and CPU's performances of FDTD scheme with the PML absorbing boundary condition are compared. As results of performance measurement, the combination of GPU+CUDA achieved the highest performance, followed in order by combinations of MIC+OpenMP, GPU+OpenACC and CPU+OpenMP. Also, the performance of the model with the PML was lower than the model without the PML on MIC.**

***Keywords*—FDTD method; GPU; MIC; Xeon Phi**

I. Introduction

There has been a discussion about hardware-accelerations for the large-scale Finite-Difference Time-Domain (FDTD) method using many-core processors [1]. Graphics Processing Unit (GPU) and Intel Many Integrated Core (MIC) architecture have been brought to attention for the reason of their availability and scalability among many kinds of many-core processors, because they are used only by connecting to PCIe Buses. Over the past several years, a number of studies have reported GPU-based acceleration of FDTD scheme [2]–[5] and a few have done with MIC [6], [7]. However, little study has been done to take account of perfectly matched layer (PML) absorbing boundary condition (ABC) [8], which is necessary for practical problems in many cases.

Therefore, the present study was undertaken in order to analyze the performances of the FDTD scheme for electromagnetic analyses with the PML ABC implemented on GPU and MIC. In this paper, GPU, MIC and CPU's performances for the FDTD scheme with the PML ABC are compared. In addition, performance are compared between the model with the PML and without the PML for the MIC architecture.

II. Calculation Model and Computational Environment

Fig. 1 illustrates the schematic view of the numerical model for the performance measurement. The model has a cube domain, and a point wave source is located in the center of the domain. The PML which has a thickness of 16 layers is employed as ABC. The size of calculation domain is scaled from 64x64x64 to 512x512x512 cells.

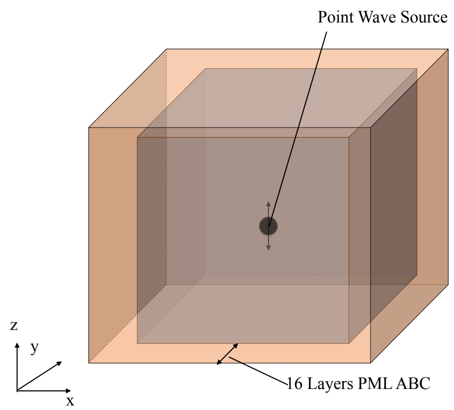

Fig. 1. Schematic view of the numerical model.

Table I shows the types of processors and parallelization methods about the GPU, MIC and CPU we have evaluated. The FDTD scheme for GPU is parallelized by Compute Unified Device Architecture (CUDA) [9] and OpenACC. The FDTD scheme for MIC and CPU is parallelized by OpenMP. Compiler versions used for each parallelization method are CUDA 6.5, PGI Accelerator C Compiler 15.4 included in OpenACC 2.0 and Intel C/C++ Compiler 15.0 included in OpenMP 4.0. All of parallelization methods are naively implemented, without performing optimizations such as cache blocking, padding and explicit vectorization.

TABLE I. Processors and Parallelization Methods about the GPU, MIC, and CPU

Architecture	Processor	Parallelization Method
GPU	NVIDIA Tesla K40	CUDA (ver. 6.5)
		OpenACC (ver. 2.0)
MIC	Intel Xeon Phi 7120P	OpenMP (ver.4.0)
CPU	Dual Intel Xeon E5-2620 v3	

978-1-4673-8801-6/16/$31.00 ©2016 IEEE

III. Measurement Results

Fig. 2 shows dependences of performances (GFlops) on the number of calculation domain cells among four types of combinations of processors and parallelization methods. According to Fig.2, performances increased along with the growth of number of calculation domain cells for GPU and MIC. The combination of GPU+CUDA indicated the highest performance, followed in order by combinations of MIC+OpenMP, GPU+OpenACC and CPU+OpenMP. In the condition of 512x512x512 cells, combinations of GPU+CUDA, MIC+OpenMP, GPU+OpenMP and CPU+OpenMP achieved 55.1GFlops, 26.7GFlops, 13.1GFlops and 2.74GFlops, respectively.

In Fig.3, performances are compared between the model with PML and without PML for MIC. Fig. 3 revealed that the performance of the model with PML was lower than the model without PML. In the condition of 512x512x512 cells, performances of the model with the PML and without the PML were 34.8GFlops and 26.7Gflops, respectively.

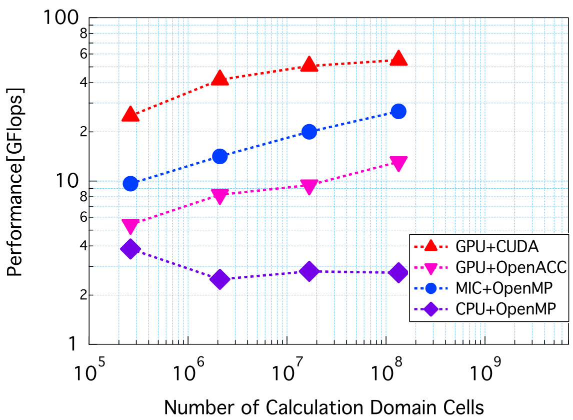

Fig. 1. Dependence of Performances (GFlops) on the Number of Calculation Domain Cells among Four Types of Combinations of Processors and Parallelization Methods.

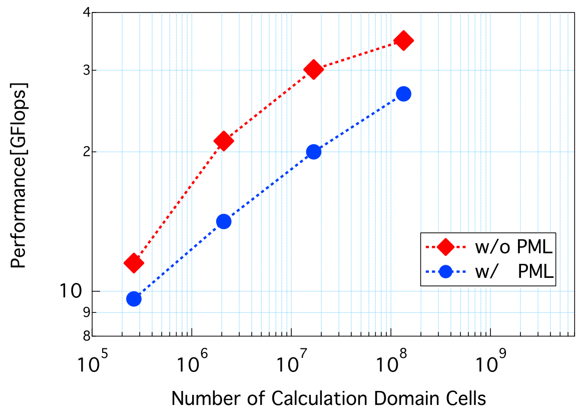

Fig. 3. Performance Comparison between the Model with PML and without PML for MIC.

IV. Conclusion

In this paper, we compared performances of the FDTD scheme with the PML ABC between GPU, MIC and CPU. As results of performance measurement shown in Fig. 2, GPU+CUDA achieved the highest performance, followed in order by MIC+OpenMP, GPU+OpenACC and CPU+OpenMP.

The performance obtained by the model with the PML was lower than that without the PML. In future work, we would like to reveal the cause of lower performance indicated in the FDTD calculation with the PML ABC and to make suitable optimization for that.

References

[1] D. Orozco and G. Gao, "Mapping the FDTD application to Many-Core Chip Architectures," in *Parallel Processing, 2009. ICPP'09. International Conference on*. IEEE, 2009, pp. 309–316.

[2] D. De Donno, A. Esposito, L. Tarricone, and L. Catarinucci, "Introduction to GPU computing and CUDA programming: A case study on FDTD [EM programmer's notebook]," *IEEE Antennas and Propagation Mag.*, vol. 52, no. 3, pp. 116–122, 6 2010.

[3] K.H. Kim, K. Kim, and Q.H. Park, "Performance analysis and optimization of three-dimensional FDTD on GPU using roofline model," *Comput. Phys. Commun.*, vol. 182, no. 6, pp. 1201–1207, 6 2011.

[4] K. Kim and Q. Park, "Overlapping computation and communication of three-dimensional FDTD on a GPU cluster," *Computer Physics Communications*, vol. 183, no. 11, pp. 2364–2369, 11 2012.

[5] T. Nagaoka and S. Watanabe, "Accelerating three-dimensional FDTD calculations on GPU clusters for electromagnetic field simulation," in *Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, San Diego, California USA, 2012, pp. 5691–5694.

[6] Y. Wenhua and Y. Xiaoling, "Novel hardware acceleration techniques for finite difference time domain methods," in *Electromagnetics in Advanced Applications (ICEAA), 2014 Int. Conf. on*, Palm Beach, USA, 8 2014, pp. 167–169.

[7] T. Nagaoka and S. Watanabe, "Efficient three-dimensional FDTD computation with emerging many-core coprocessor for bioelectromagnetic simulation," *Trans. Jpn. Soc. for Medical and Biol. Eng.*, vol. 51, no. Supplement, pp. R–39, 2013.

[8] J. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Computational Phys.*, vol. 114, no. 2, pp. 185–200, 10 1994.

[9] NVIDIA, "CUDA C Programming Guide," 2015. [Online]. Available: http://docs.nvidia.com/cuda/cuda-c-programming-guide