Large Scale Simulation of Seismic Wave Propagation using GPGPU

S. Aoi
National Research Institute for Earth Science and Disaster Prevention. Tsukuba, Japan

N. Nishizawa
SGI Japan, Ltd, Tokyo, Japan

T. Aoki
Global Scientific Information and Computing Center, Tokyo Institute of Technology, Japan

SUMMARY:
Huge computation resources are required to perform large-scale wave propagation simulations using realistic models for high accuracy assessment. Though the information technology drastically improves the capacity of computer, it is still a bottleneck of hazard calculation because the practical model is rather large scale. To overcome this difficulty, we applied multi GPU technique to the GMS (Ground Motion Simulator) which is a practical tool for wave propagation simulation using discontinuous grid. The performance test for the multi GPU calculation showed almost perfect linearity for the weak scaling test up to the simulation with 1024 GPUs where the model sizes (number of grids) are increased in proportion to the degree of parallelism (number of GPUs). Here we used the model with about 22 million grids as the unit model and the model size for the 1024 GPUs case is about 22 billion grids. The typical turnaround time was less than one hour and the performance of the GMS on GPU is practically satisfactory.

Keywords: GPGPU, ground motion simulation, FDM, GMS

1. INTRODUCTION

Three-dimentional ground motion simulation is one of the key techniques for seismic hazard assessment using the hybrid method. In the method, high frequency component which is rather random process is simulated by the stochastic Green's function method. On the other hand, low frequency component is calculated by deterministic method such as FDM (finite difference method) and FEM (finite element method) assuming source and velocity structure models. The capacity (calculation and memory resources) of recent powerful computer is still not enough for detailed seismic hazard assessment because the practical model is rather large scale.

One of the prevailing approaches to overcome the problem of heavy computational expense is the utilization of the GPGPU (General purpose computing on graphics processing units) (e.g. Aoi et al., 2009; Michéa and Komatitsch, 2010; Okamoto et al., 2010). GPGPU is the technique of using a GPU to perform computation in applications traditionally handled by the CPU. In this study, we develop the FD code for the multi GPU simulation and examine the performance test on TSUBAME2.0 which is one of the Japanese fastest supercomputer operated by Tokyo Institute of Technology.

2. FD SIMULATION OF WAVE PROPAGATION USING GPU

2.1. FD Formulation using Discontinuous Grid

In this study, we employed the solver of the GMS (Ground Motion Simulator; Aoi et al., 2004) as the original code and develop the code for the multi GPU computation. GMS is a total and practical system for seismic wave propagation simulation based on 3-D FDM using discontinuous grids (Aoi and Fujiwara, 1999), which includes the solver as well as the preprocessor tools (parameter generation

tool) and postprocessor tools (filter tool, visualization tool, and so on). The solver of the GMS employs the velocity-stress formulation (e.g. Virieux, 1984) using the FD operator having fourth-order of accuracy in space and second-order in time (e.g. Graves, 1996).

One of the problems in FD modeling using uniform grid is the extra computational requirements which are related with the oversampling of the models. This problem is getting obvious when the model includes the strong velocity contrast, because the grid spacing is determined by the shortest wavelength in the region to be calculated. In order to avoid this oversamping we use a discontinuous grid (Figure 1) that is a kind of a non-uniform grid adapted to the velocity structure.

This grid system consists of two regions having the different grid spacing whose ratio is a factor of three, and the continuity of the wave field are kept by eliminating or inserting grids at the overlapping region of these two regions. The use of discontinuous grid significantly reduces the computational requirements, which is model dependent but typically one-fifth to one-tenth, without a marked loss of accuracy.

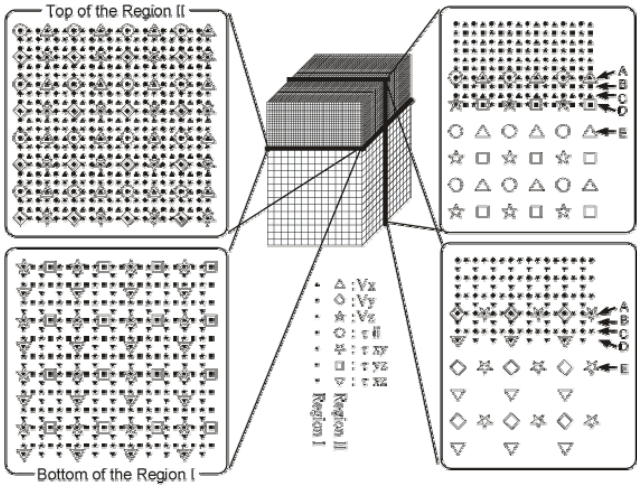

Figure 1. Center: 3-D discontinuous grid system. Left: Two transections on the top and at the bottom of the overlapping region of Regions I and II, where the elimination or the insertion of grid points is necessary. Right: Two profiles of the discontinuous grid. The arrows A-E show the overlapping region of Regions I and II (After Aoi and Fujiwara, 1999).

2.2. Implementation of the Code for Computations on GPU

We develop the FD code using CUDA (Compute Unified Device Architecture). The computational model is decomposed in two horizontal directions and each decomposed model is allocated to a different GPU (Figure 2). Because the values on the grid at the boundary of the neighbor decomposed models are necessary for the calculations, two grids from the boundary are overlapped each other and the values on these grids are exchanged by MPI (Message Passing Interface). Relative time required for the communication compared to the time for the calculation is longer for GPU than for CPU, because the calculation speed is much faster for GPU. Moreover, the overheads for the communication are larger for GPU because direct communications are not available and values are transmitted to the target GPU via CPU using MPI. Therefore the time for the communication is not negligible and the concealment technique of the communication by overlapping the calculation and the communication is essentially important for achieve high performance parallel computation using GPU. Popular technique for concealing the communication is follows (Figure 3a): Values on the overlapped grids are calculated first and then the communication of those values between neighbor decomposed models are performed during the calculation of rest grids. This technique is not efficient because it requires discontinuous memory accesses which are hard for GPU. Considering that our discontinuous grids have two regions having different size of grid spacing, exchanges of the values on the overlapped grid in one region are made during the calculation of another region(Figure 3b). Our concealment technique makes it possible to avoid the discontinuous memory accesses.

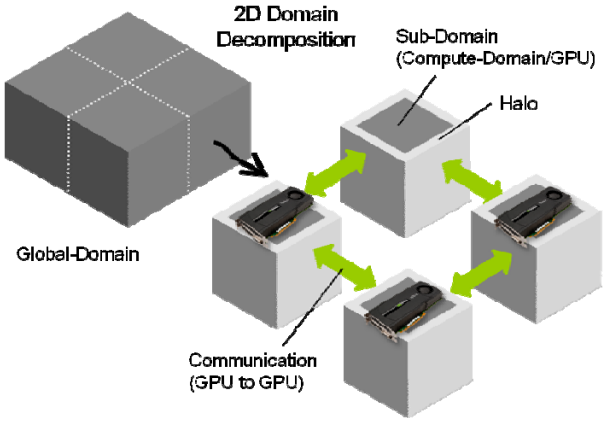

Figure 2. Model decomposition for the parallel simulation. The computational model is decomposed in two horizontal directions and each decomposed model is allocated to a different GPU.

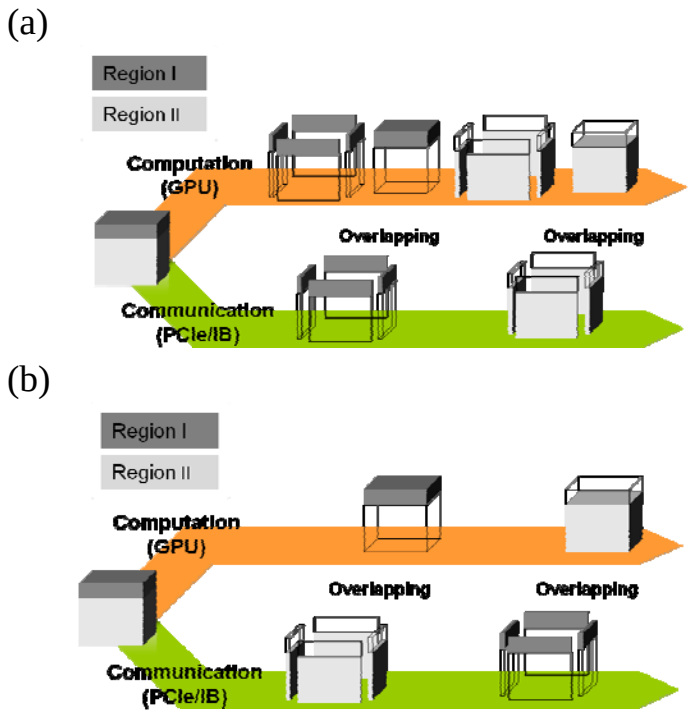

Figure 3. Two methods for concealing the communication. (a) Popular technique: the values on the overlapped grids are antecedently calculated to other grids. (b) Our proposed technique: the values in one region are exchanged during the calculations of another region. This technique is more efficient because it successfully avoid the discontinuous memory accesses.

3. PERFORMANCE TESTS

We examined the two kinds of performance tests for parallel computing; weak and strong scaling tests. For the strong scaling test, we used a small model, *unit420*, which has about 22 million (420 x 420 grids for the two horizontal directions in Region I; Figure 4 and Table 1) and the model size is independent from the degree of parallelism. The speed-up using the *model420* is 3.2 and 7.3 for 4 and 16 GPUs cases, respectively (Figure 5). The reason of the decrease of the parallel performance is that the communication time increases so that it is no longer possible to conceal by calculation time. Moreover, the number of the threads for each GPU decreases because the model size allocated to each GPU becomes too small.

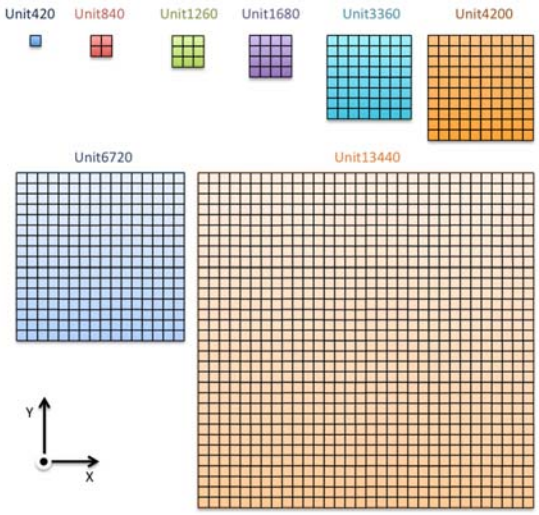

Figure 4. Models used for the performance test.

Table 1. Grid Configuration of *unit420*

	# of grids for each direction			# of grid
	x	y	z	
Region I	420	420	100	17,640,000
Region II	140	140	200	3,920,000
Total				21,560,000

Table 2. Numbers of Total Grids for the Models used for Performance Test

Model Name	# of grids
unit420	21,560,000
unit840	86,240,000
unit1260	194,040,000
unit1680	344,960,000
unit3360	1,379,840,000
unit4200	2,156,000,000
unit6720	5,519,360,000
unit13440	22,077,440,000

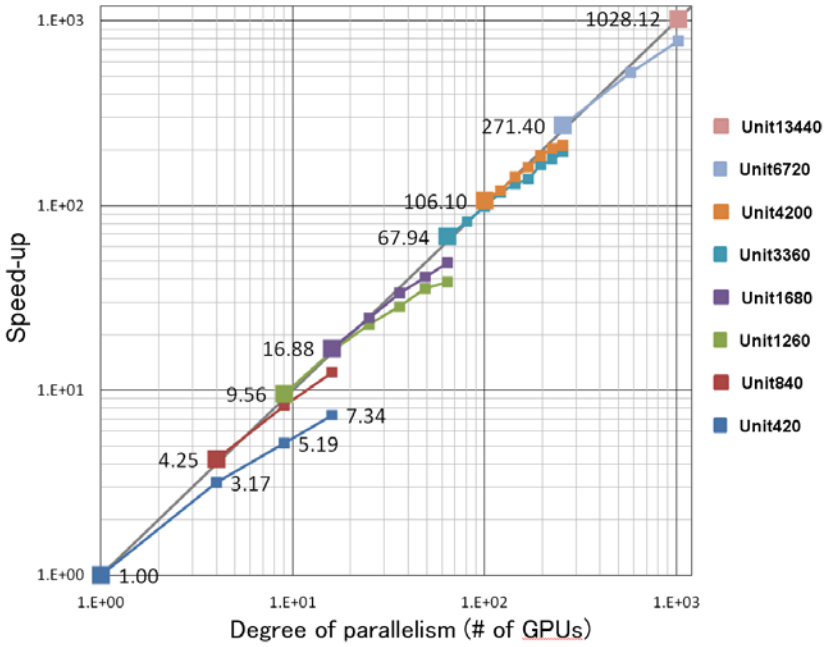

Figure 5. Scalability of the parallel computation on GPU. Each color (which corresponds the color for the model shown in Figure 4) shows the result of performance test using different size of the model.

On the other hand, for the weak scaling test where the model sizes (number of grids) are increased in proportion to the degree of parallelism (number of GPUs), the result showed almost perfect linearity up to the simulation with 1024 GPUs (Figure 5). Here we used the *unit420* as the unit model and the model size for the 1024 GPUs case is about 22 billion grids (*unit13440*; Figure 4 and Table 2).

Considering that the time steps for most model we use for simulation are up to hundreds thousand, the turn around times are several minutes to a few hours when the GPU resources appropriate to the size of the model is available. Thus, the performance of the GMS on GPU was practically satisfactory for most cases.

4. CONCLUSIONS

We have applied the GPGPU computation to a ground motion simulation which requires large computational resources. This is the technique of using a GPU to perform computation in applications traditionally handled by the CPU. We have developed the simulation code for GPGPU using CUDA based on the solver of GMS which is a total system for seismic wave propagation simulation. The computational model is decomposed in two horizontal directions and each decomposed model is allocated to a different GPU. It is substantially important for high performance GPU calculation to conceal the communication between each GPU by overlapping the calculation and the communication. We have proposed a new efficient concealing technique that successfully avoids the discontinuous memory accesses.

First we have performed strong scaling test using the model with about 22 million grids (*unit420*) and achieved 3.2 and 7.3 times of the speed-up by using 4 and 16 GPUs. Next, we have examined the weak scaling test where the model sizes (number of grids) are increased in proportion to the degree of parallelism (number of GPUs). The result showed almost perfect linearity up to the simulation with 1024 GPUs. The typical turn around time is less than one hour and the performance of the GMS on GPU is practically satisfactory.

REFERENCES

Aoi, S. and Fujiwara, H. (1999). 3-D finite difference method using discontinuous grids. *Bulletin of the Seismological Society of America.* **89**, 918-930.

Aoi, S., Hayakawa, T. and Fujiwara, H. (2004). Ground motion simulator: GMS. *Butsuri-Tansa.* **57**, 651– 666 (in Japanese with English abstract).

Aoi, S., Nishizawa, N. and Aoki, T. (2009). 3-D wave propagation simulation using GPGPU. *Programme and Abstracts, Seismological Society of Japan 2009 Fall Meeting*. A12-09 (in Japanese).

Graves, R.W. (1996). Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differenes. *Bulletin of the Seismological Society of America.* **86**, 1091-1106.

Michéa, D. and Komatitsch, D. (2010). Accelerating a three-dimensional finite-difference wave propagation code using GPU graphics cards. *Geophysical Journal International*. **182**, 389-402, doi:10.1111/j.1365-246X.2010.04616.x.

Okamoto, T., Takenaka, H., Nakamura T. and Aoki, T. (2010). Accelerating large-scale simulation of seismic wave propagation by multi-GPUs and three-dimensional domain decomposition. *Earth, Planets and Space*. **62**, 939-942.

Virieux, J. (1984). SH-wave propagation in heterogeneous media: Velocitiy-stress finite-difference method. *Geophysics*. **49**, 1933-1957.