1P2-13

MIC and GPU-based implementation of sound field simulation using parallelized FDTD methods

MIC と GPU による並列化 FDTD 法系音場シミュレーションの実装と評価

Nastuhiko Araki[1†], Ryo Imai[2], Yukihisa Suzuki [3], and Kan Okubo [4] ([1,2,3,4] Tokyo Metropolitan Univ.)

荒木夏彦 [1†]，今井陵 [2]，鈴木敬久 [3] ， 大久保寛 [4] ([1,2,3,4] 首都大学東京)

1. Introduction

In recent years, as processor for parallel computing, meny-core-based archtecture released; e.g., Intel many-integrated core Architectures (MIC). MIC have many cores more than 60 and on-packeged memory for high memory bandwidth. Unlike GPU, we can use program codes written for general CPUs to MIC on parallel computing without APIs like CUDA.

Finite-difference time-domain (FDTD) Schemes[1] are simple to implement as an acoustic simulation. It is suitable for parallel computing, and moreover the performance strongly depends on the memory bandwidth. So the performance of acoustic FDTD simulation on MIC and/or GPU may be effective rather than use of regular DDR memory with common CPUs.

In this study, we evaluated the performance of four kinds of acoustic FDTD schemes on the GPU ,Intel MIC and CPUs. In addition, We applied some software optimizations on MIC and CPU to reveal the best performance on those. Also, we use Fups (fields update per second, or Cups; cells update per second) for evaluation scale.

2. FDTD Schemes

The governing acoustic equation for loss-less media in theree dimensions is shown as the following two equatins.

$$\rho \frac{\partial \vec{v}}{\partial t} = \nabla p \qquad (1)$$

$$\nabla \cdot \vec{v} = \frac{1}{K}\frac{\partial p}{\partial t} \quad (2)$$

where ρ is the density of the medium[kg/m^3], v is the particle velocity[m/s], p is the scalar sound pressure [Pa], K is the bulk modulus[Pa], and t is the time[s]. Then, by approximating Eq.1 and Eq.2 with the second order central difference, FDTD(2,2) scheme is obtained. FDTD(2,4) is obtained similarly by using four-order precision in space domain.

Eliminating $\vec{v}$ using Eq.1 and Eq.2, obtain following acoustic wave equation.

$$\frac{1}{c}\frac{\partial^2 p}{\partial t^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} \quad (3)$$

Similarly, applying second orde central difference for Eq.3, so-called WE-FDTD(2,2) scheme[3] is obtained, wheares approximating Eq.3 with four-order central difference in space domain, we obtain WE-FDTD(2,4) scheme[2].

Although the amount of calculation increase, both FDTD(2,4) and WE-FDTD(2,4) schemes have good performance rather than both FDTD(2,2) and WE-FDTD(2,2) respectively in terms of accuracy of calculation. Because wave equation (WE-) type calculate without the variable of particle velocity, their amount of calculation is less than FDTD(2,2). Both methods provide same calculation result of acoustic pressure. The relationship between WE-FDTD(2,4) and FDTD(2,4) is almost similar.

3. Software Optimization

Basically, we use the intel icc compiler with -O3 flag and the OpenMP for multithreading in this study. We performed one or more following software optimization for computation on MIC and CPU.

- **NOOPT**
  Basic optimization. Using OpenMP and -O3 flag for the icc compiler. This is applied for all otimization patterns.
- **SIMD**
  Insertion SIMD pragma[4] on for-loops. It guide the compiler to vectorize codes inside for-loop.
- **TILE**
  Dividing the calculation domain into small sub-domains to avoid cache-hit miss ratio. In this study, y-direction loop divided two loops.
- **COLLAPSE**
  Insertion collapse(2) pragma[5] on top of the nested for-loop. It transform two nested loop into bigger one. So each thread would works more efficiency.

4. Performance measurement

In this study, we use Fups as an evaluation scale. Fups indicates the number of field points updated per second. Fups is derived from calculation time [s] as follows:

$$\mathrm{Fups} = \frac{Nx \cdot Ny \cdot Nz \cdot Nt}{Calculation\ time\ [\mathrm{s}]}$$

When *Nx,Ny ,Nz* are number of points in the *x*, *y*, *z*-direction in the calculation domain. Nt is the number of time-loop iteration.

5. Result of Performance measurement

Fig.1 shows the results of the performance measurement for Intel Skylake-SP(CPU), KNL (MIC) and P100(GPU) with CUDA-8.0.

According to Fig.1, as compared Fups in scheme types, WE-FDTD(2,2) is larger than WE-FDTD(2,4), FDTD(2,2) and FDTD(2,4). The result of WE-FDTD(2,2) is more than two-times as high as FDTD(2,2). This manner is applicable for (2,4) schemes. Also, FDTD(2,2) scheme is less affected by optimization. In an extreme case, fups at NOOPT is larger than that at other optimization cases on Skylake-SP.

Fups of all FDTD schemes on GPU (P100) is greater than other MIC and CPU. On KNL, TILE+ COLLAPSE optimization is most effective for all schemes, while TILE optimization without COLLAPSE lower fups for NOOPT. On Skylake-SP, most scores are lower in comparison with KNL(MIC) and GPU.

When using WE-FDTD(2,4) with TILE+COLLAPSE optimization gains few speeds up against NOOPT optimization on Skylake-SP. But, in other schemes, this optimization makes fups lower.

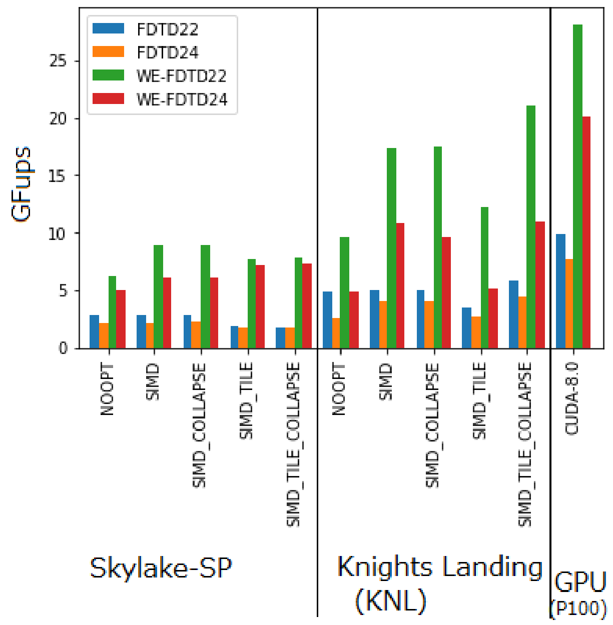

Figure 1 Performance (Giga Fups)

6. Conclusions

In this study, we implemented four kinds of FDTD schemes on MIC, CPU and GPU. Then we performed some software optimizations for CPU, MIC codes.

In for FDTD schemes, GPU (P100) perform better than CPU and MIC. As for types of schemes, WE-FDTD(2,2) scheme is fastest. The sound field calculated without the variable of particle velocity by WE-FDTD(2,2) is almost identical to that of FDTD(2,2). However, FDTD(2,2) requires more than two times calculation time than WE-FDTD(2,2). This manner is applicable for (2,4) schemes.

On KNL, TILE optimization makes best performance for all schemes. It also requires COLLAPSE optimization to resize the for-loop.

References

[1] K. S. YEE. Numerical solution of initial boundary value problems involving maxwell's equation in isotropic media. IEEE Trans. Antennas &Propagat., Vol. 14, No. 3, pp. 302-307, 1966.
[2] Yuuki Sendo, Hironori Kudo, Tatsuya Kashiwa, and Tadao Ohtani. The fdtd(2, 4) method for highly accurate acoustic analysis in three - di-mensional space. Vol. 86, pp. 30 - 37, 11 2003.
[3] Naoki Kawada, Takeshi Yoda, Norio Tagawa, Takao Tsuchiya, and Kan Okubo. Evaluation of acoustic simulation using wave equation finite difference time domain method with compact nite differences. Japanese Journal of Applied Physics, Vol. 51, No. 7S, p. 07GG06, 2012.
[4]https://software.intel.com/en-us/node/524555/. (2018.8.9 access).
[5]https://software.intel.com/en-us/articles/openmp-loop-collapse-directive/. (2018.8.8 access).