

GPU の総和計算アルゴリズムによる FDTD の高速化 Acceleration of FDTD calculation by parallel reduction algorithm in GPU

NTT ナノフォトニクスセンタ¹, NTT 物性基礎研²

○谷山 秀昭^{1,2}, 納富 雅也^{1,2}

NTT Nanophotonics Center¹, NTT Basic Research Lab.²

○Hideaki Taniyama^{1,2}, Masaya Notomi^{1,2}

E-mail: taniyama.hideaki@lab.ntt.co.jp

【はじめに】マクスウェル方程式の数値的な解法である Finite-Difference Time-Domain (FDTD) 法は、光学微細構造における電磁波の振舞いの解析において重要なツールとなっている。しかし、FDTD 計算には大容量のメモリと長時間の計算が必要な点が問題であった。これに対し我々は、GPU の高並列計算と高速なメモリアクセスの特長を利用することで、FDTD 計算の高速化を実現してきた [1]。FDTD では各格子点の値を計算する際、隣接する格子点の値のみを参照するため並列化に適していたと言える。一方、系全体の電磁場エネルギー計算のように、全要素から一つの物理量を求める総和計算を GPU で並列に行うには、GPU の特性を考慮したアルゴリズムの採用が必要である。特に GPU を用いて総和計算を効率的に行うためには、効率低下の原因となるウォープ・ダイバージェントを少なくする必要がある。またメモリアクセスの効率化のためは、バンクコンフリクトを防ぐようにコーディングすることも重要である。

【手法】今回我々は、FDTD 中の総和計算のアルゴリズムとして、スレッド番号を工夫することでウォープ内の分岐を無くすと同時に遠いデータ同士を先に足すなどメモリアクセスの順序を変更し、更にループの最適化を行うことで GPU の特長を活かすという Harris が提案した並列計算の最適化手法 [2] を採用して FDTD 計算全体の高速化を試みた。

【計算結果】評価のため $200 \times 30 \times 325$ のグリッドサイズに対し 196,000 ステップの計算を行った。使用した計算機は、CPU が Xeon E5606 2.13GHz、GPU が GeForce GTX 580 からなる計算ノードが InfiniBand で接続されたクラスタ構成となっており、今回はそのうち 5 ノードを用いた。図 1 に時間ステップ計算のみの場合、および各時間ステップ毎に総和計算を行った場合の計算時間を示す。既に報告したように [1]、時間ステップ計算だけの場合は GPU を用いることで計算時間が大幅に短縮されている。しかし時間ステップ計算を GPU で行っても総和計算を CPU で行った場合はその差が小さくなっている。この速度低下の原因は CPU の遅さだけでなく、CPU-GPU 間のデータ転送時間が大きく影響している。今回は、全計算に GPU を用いることで大幅な計算時間の短縮が達成できた。

[1] H. Taniyama et al., PIERS 2011, 1A9-K-14, 2011 年秋季応物, 31a-ZR-3, 2013 年 5 月電子通信学会エレクトロニクスシミュレーション研究会.

[2] M. Harris: "Optimizing Parallel Reduction in CUDA", http://developer.download.nvidia.com/compute/cuda/1_1/

Website/Data-Parallel_Algorithms.html#reduction. 図 1 FDTD 計算に要した時間。全ての計算を CPU で行った場合 (CPU)、総和計算のみ CPU で行った場合 (CPU+GPU)、全ての計算を GPU で行った場合 (GPU) の結果。

