GTC Workshop Japan 2011 テクニカルセッション

発表申込書

発表者の氏名：**福田　圭祐、丸山　直也、松岡　聡**

発表者の所属：東京工業大学　数理計算科学専攻

連絡先：　　住所　東京都目黒区大岡山2-12-1

メール　fukuda@matsulab.is.titech.ac.jp

電話番号　03-5734-3881

■　口頭発表を希望する。

■　ポスター発表に回った場合に、ポスター発表を行う。

□　ポスター発表は行わない。

□　ポスター発表を希望する。

（■：希望する。　□：希望しない。）

GTC Workshop Japan 2011 (2011 年 7 月 22 日)

CPU/GPU ヘテロジニアス環境における FMM の最適化

Optimization of FMM on CPU/GPU heterogeneous environment

福田　圭祐 [1)], 丸山　直也 [1)], 松岡　聡 [1),2)]
Keisuke Fukuda, Naoya Maruyama and Satoshi Matsuoka

1) 東京工業大学　数理計算科学専攻 （〒 152-8550 東京都目黒区大岡山 2-12-1）
2) 国立情報学研究所 （〒 101-8430　東京都千代田区一ツ橋 2-1-2）

Key Words: *CUDA, Heterogeneous, FMM*

1　はじめに

CPU と GPU の両方のプロセッサを搭載したヘテロジニアス環境が一般的になりつつある。スーパーコンピューターの性能を示す Top500 ランキングでは、2010 年 11 月時点のランキングで上位 5 サイトのうち 3 サイトがこのようなヘテロジニアス環境であった。これは、GPU の持つ高いメモリバンド幅や電力効率が評価されているためである。

一方で、GPU に適したアルゴリズムの開発は発展途上であり、さまざまなアルゴリズムの GPU での効率的な実装が研究されている。本稿で着目する Fast Multipole Method(FMM) は、N 体問題向けの近似計算アルゴリズムである。O(N) の計算量をもち、スケーラブルであることから分子動力学などの分野で注目されている。FMM の粒子間力の評価は 6 つの計算フェーズからなり、それぞれが計算量や計算特性において異なる。

本稿では Fast Multipole Method を対象として CUDA を用いた GPU 実装を行い、CPU での逐次および OpenMP による並列実装に比べて大幅な高速化が達成されたことを示す。

2　実装

本稿では、Ying ら [1] による FMM の派生アルゴリズムである KIFMM の参照実装をベースとし、評価フェーズのうち、Upward フェーズの一部と U-list フェーズについて CUDA を用いて GPU 実装を作成した。また X,W を除く全てのフェーズについて OpenMP を用いて CPU 上で並列化も行った (今回の評価に用いた粒子分散では X,W-list は計算時間ゼロのため)。

U-list フェーズは、従来の N 体問題の直接計算（計算量が $O(N^2)$ である）を局所的に行うフェーズである。この部分は計算密度が高く、GPU に適した計算であると考えられる。また Upward フェーズは、全空間を分割した部分空間において数十　1900 個程度の粒子の情報を 1 つのベクトル（長さ数十　150 程度）に集約する操作であり、粒子数程度の並列度を持つ。U-list フェーズほどではないものの、GPU に適した計算フェーズであり高速化が見込まれる。

それ以外のフェーズについては OpenMP 版と同じ物を利用し、「CUDA+OpenMP 実装」とした。

また、OpenMP 実装については、#omp parallel for ディレクティブを用いて for 文の最適化を行った。

表 1: 表 1 評価環境

	CPU	GPU
Type	Intel 6 core Xeon X5670 × 2	NVIDIA Tesla M2050 × 3
Freq	2.93GHz	1.15GHz
Cores	6 x 2 (with HT)	14SM/448 cores
Memory	54GB	3GB

3　評価

評価環境は表 1 に示した TSUBAME2.0 の単ノードで、CPU1 ソケット、GPU1 ソケットを用いた。

また、比較のための逐次実装として、われわれの OpenMP 実装を 1 スレッドで実行した物を用いた。評価結果を図 1 に示した。逐次実装に対して、GPU+OpenMP 実装は 13.8 倍、OpenMP 実装は 5.5 倍の高速化を実現した。またフェーズごとの比較としては、CUDA+OpenMP 実装は U-list フェーズが 117 倍、Upward フェーズが 8.7 倍、OpenMP 実装はそれぞれ 5.8 倍、5.0 倍であった。

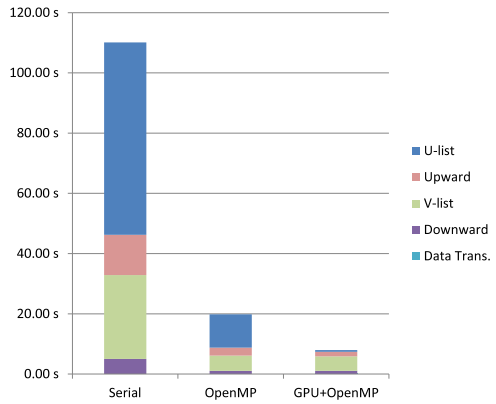

図 1 評価結果

参考文献

[1] A kernel-independent adaptive fast multipole algorithm in two and three dimensions, J. of Computational Physics,2004