講演番号：44
D-1

関数実行順序制御による並行計算の高速化

Accelerate of Concurrent Computing by Function Execution Scheduling

髙松 健 † 黒木 啓之 †
Ken Takamatsu† Takashi Kuroki†
† 東京都立産業技術高等専門学校
†Tokyo Metropolitan College of Industrial Technology

1 はじめに

近年，理論演算性能 (FLOPS) の高い GPU を用いて汎用計算を行うことが注目を集めている．GPU を用いて計算することで単一の大きなデータを複数のスレッドで分割して計算する並列実行，処理の順番に依存がない異なる処理などを並行実行することができる．

しかし，実際に GPU を用いて計算する場合，実際の演算性能が理論演算性能を大きく下回ってしまうことがある．これは主に GPU で計算をするために必要なデータ転送にかかる時間が原因である．この問題を回避するためには，データ転送のスケジューリングを行うことが必要である．

本研究では，データの転送順序と関数実行順序のスケジューリングに注目して並行計算の高速化手法することを目的とする．

2 データ転送とスケジューリング

GPU を用いて計算を行う際に用いるメモリは CPU で利用されるメモリと異なるため，計算をする際にはデータを転送する必要がある．この際，GPU 上で関数を効率的に計算させるためには，関数の実行中に次に行う処理をオーバーラップさせる，関数の実行順序のスケジューリングを行う等の工夫が必要である．

3 提案手法

既存手法 [1] ではスケジューリングの方式として事前転送方式と直前転送方式という 2 つの手法を提案している．本研究では特に直前転送方式に注目して改良を行った．改良点として，既存手法では関数の実行順序は関数実行を発行した順番で実行をしているが，本研究では転送されるデータのサイズでソートを行ってから実行している．

4 研究環境

表 1 に本研究の研究環境を示す．

表 1 研究環境

OS	Ubuntu 18.04, CentOS 7
GPU	GTX680Ti, GTX980
ライブラリ	OpenMP 4.5, CUDA 10.1

5 研究結果

サイズの異なる配列を同一の関数で並行実行した際の処理時間の計測を行った． 図 1 に評価結果を示す．縦軸は実行時間を表しており，既存手法を 1 として正規化している．また左から GPU での逐次実行，既存手法のスケジュー リング，提案手法のスケジューリングとなっている．測定の結果，既存手法に比べて提案手法が 1% の高速化ができていることがわかった．

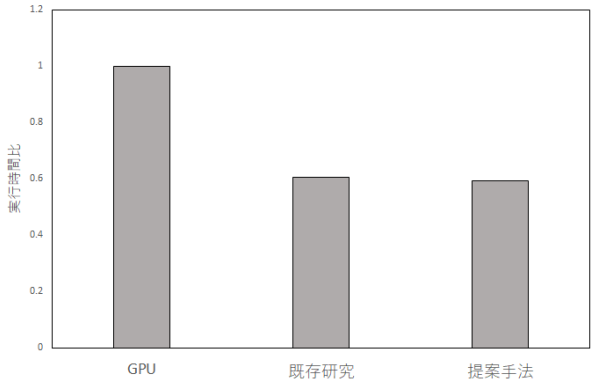

図 1 評価結果

6 おわりに

本研究では，既存手法を改良し，データサイズによる関数実行のスケジューリングをすることで並行計算が高速化できることがわかった．

今後の課題として，複数のベンチマークプログラムを利用して総合的に実行時間を測定する，また様々な関数実行順序を試し，さらなる高速化を目指す．

参考文献

[1] 小野和馬ら，”データ転送の発行順序制御による GPU プログラムの高速化”, 信学技報, vol. 115, no. 174, CPSY2015-23, pp. 85-90, 2015 年 8 月.