

ステンシル計算における通信の自動最適化に向けた性能モデルの評価

河村 知輝[†] 丸山 直也^{†,††,†††} 松岡 聡^{†,†††}

1. はじめに

ステンシル計算は流体運動や熱伝導などの偏微分方程式を数値計算で解く際に現れる。ステンシル計算を解く際に GPU を用いることで高速化されることはよく知られており、大規模に並列実行を行うと隣接通信がボトルネックとなる。この隣接通信コストを抑える手法としてテンポラルブロッキング¹⁾と呼ばれる既存手法がある。この手法には「1 通信あたりの計算回数(段数)」が存在するため、最適段数を選ぶ必要が出てくる。

そこで、本研究ではテンポラルブロッキングの自動チューニングを提案する。本稿では自動チューニングのための準備として、手動で最適化を行い性能評価をし、最適段数を求めるための性能モデルを構築し実測値と比較することにより評価を行った。

2. 性能モデル構築

手動での最適化をする際にはテンポラルブロッキングの他に MPI による 2 次元分割、各種オーバーラップや境界領域計算の並列化も行っている。実装の流れは図 1 のようになる。GPU カーネルは事前に計測したメモリバンド幅の値と計算領域から実行時間を算出するモデルを構築し、CPU-GPU 通信と MPI 通信は別途用意したベンチマークから得られた実行時間を用いることで、各段数における全体の実行時間を算出する性能モデルを構築した。

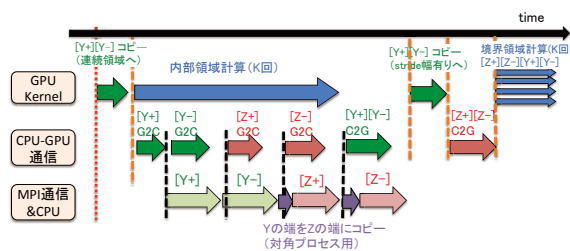


図 1 実装の概要

[†] 東京工業大学
^{††} 理化学研究所
^{†††} 独立行政法人科学技術振興機構 CREST

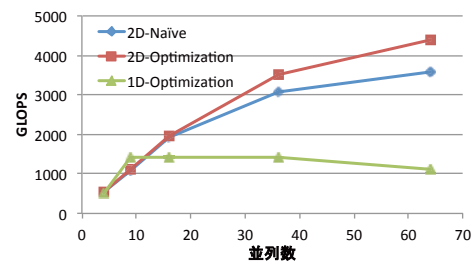


図 2 7 点ステンシル (512x768x768, Strong Scaling)

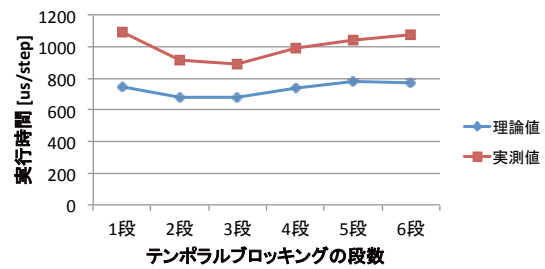


図 3 2D 分割版の性能モデル評価 (512x768x768, 64 並列)

3. 実験・評価

TSUBAME2.0 上で手動最適化の性能評価を行った結果が図 2 となる。最大で 35% 性能向上をし、並列数の増加に伴い性能向上度も増加することが確認できる。性能モデルと実測値を比較したグラフが図 3 となる。実行時間の誤差が 20% 程度あるが、最適段数の導出は可能といえる。

4. 終わりに

本稿ではテンポラルブロッキングを手動で実装をし性能評価を行った。さらに最適段数を導出のための性能モデルを構築・評価を行い有効性を確認した。

参考文献

1) M. Wittmann, G. Hager, and G. Wellein. Multicore-aware parallel temporal blocking of stencil codes for shared and distributed memory. april 2010.