データサイズとストリームの利用状況とを考慮したCUDAプログラムの高速化

小野 和馬*，竹嶌 良，津邑 公暁（名古屋工業大学）

A Dynamic Scheduling for CUDA Applications based on Data Size and Stream Usages

Kazuma Ono, Ryo Takeshima, Tomoaki Tsumura (Nagoya Institute of Technology)

1. はじめに

GPU に汎用計算を行わせる GPGPU が注目を集めている．このような GPU プログラミングにも用いることが可能である，GPU 用の並列計算アーキテクチャモデルとして，CUDA が開発されている．この CUDA には，関数を並行実行させるサポートが備わっているが，効率的な関数の並行実行のためには関数の実行順を適切に制御する必要がある．そこで我々は，関数とデータの転送の実行順を自動的に制御する手法として，事前転送方式と直前転送方式が提案している．しかし，事前転送方式はデータ転送に要する時間を考慮していない．さらに 2 つの方式をプログラマ自身が適切に選択し用いる必要がある．本稿では，前者の問題を直前転送方式を改良することで解消し，後者の問題を 2 つの方式の自動的な切り替えによって解消する．

2. 研究背景

CUDA では関数を並行実行させるためのサポートとして Concurrent Kernel Execution が備わっている．これを用いて，効率的な関数の並行実行を行うために，Kernel Reordering[(1)]という関数の発行順序を制御する手法が提案されている．Kernel Reordering では，関数の発行のみを行う専用のスレッドを追加することで，関数の発行順序を制御する．しかし，この手法を用いるには，関数が使用するデータを事前に GPU へ転送する必要があり，データ転送とカーネル関数の並行実行が行えない場合がある．この問題を解決するために我々は，関数の発行だけでなくデータ転送処理の発行も専用のスレッド（Kernel Call Thread）に任せることで，データ転送処理と関数呼出しの発行制御を行う，事前転送方式と直前転送方式の 2 つを提案している．これら 2 つの方式は，処理の発行を Kernel Call Thread に任せてから実際に処理が開始されるまでの時間によって，適する状況がそれぞれ異なる．しかし，2 つの方式が適する状況をプログラマが判断するのは難しい．さらに，直前転送方式はデータ転送に要する時間を考慮していないため，先にサイズの大きいデータを転送してしまい，他のデータ転送の開始が遅れる可能性がある．

3. 提案手法

本稿では，データ転送に要する時間を考慮するよう直前転送方式の改良を提案する．さらに，プログラマが意識せずとも，状況に応じて 2 つの方式を自動的に切り替える手法を提案する．まず，前者については，サイズの小さいデータを先に転送することで，他のデータ転送の開始を早め，そのデータを用いるカーネル関数の実行開始も早める手法を提案する．次に，後者については，処理の発行をスレッドに任せてから実際に発行されるまでの時間によって，2 つの方式のいずれが適するかが異なることから，この時間に影響を与えるストリームの利用状況および，関数が入力とするデータサイズをパラメータとして，いずれの方式を適用すべきかを自動的に判別する手法を提案する．

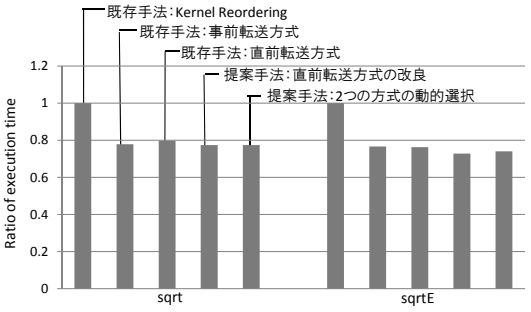

Fig.1　Result

4. 評価

評価結果を Fig.1 に示す．評価には，平方根を計算するプログラムを 2 種類用いた．一方のプログラム（sqrt）は，通信量がスレッド毎に全て同じプログラムで，もう一方のプログラム（sqrtE）では，大小 2 種類の通信量を扱うスレッドが半数ずつ存在している．グラフの縦軸は実行時間を示しており，各手法の実行時間は，既存手法である Kernel Reordering の実行時間を 1 として正規化してある．評価結果から，sqrt では既存の直前転送方式とほぼ結果が変わらないのに対し，sqrtE では，直前転送方式の改良手法が約 4.7%，動的切り替え手法が約 2.7%，それぞれ既存の直前転送方式より実行時間を削減でき，スレッド間に通信量のばらつきがある場合に効果が得られることが確認できた．

5. おわりに

本研究では，既存の直前転送方式に対して，データサイズを考慮する改良を提案した．さらに，事前転送方式と直前転送方式を動的に切り替える手法を提案した．今後の課題としては，動的切り替え手法の実行時間をより短くするよう，切り替えのパラメータを調整したり，評価に用いるベンチマークプログラムを増やすことが挙げられる．

文　献

(1) Florian Wende, et al.: On Improving the Performance of Multi-threaded CUDA Applications with Concurrent Kernel Execution by Kernel Reordering: *Proc, SAAHPC*, pp. 75–83(2012)