

## 科学研究費助成事業（学術研究助成基金助成金）研究成果報告書

平成 25 年 5 月 20 日現在

機関番号：12102
研究種目：挑戦的萌芽研究
研究期間：2011～2012
課題番号：23650022
研究課題名（和文） データ駆動的待ち合わせ機構による GPGPU の非定型的処理への応用に関する研究
研究課題名（英文） Research on the mechanism for handling non-uniform GPGPU applications by data-driven matching
研究代表者 山口 喜教（YAMAGUCHI YOSHINORI） 筑波大学・システム情報系・教授 研究者番号：00312827

研究成果の概要（和文）：GPGPU をより汎用的で非定型的な処理へ応用するために、タスクベースの細粒度実行機構を提案し、それによって GPU 内および複数の GPU 間における処理の負荷を動的にバランスさせる手法に関して研究を行った。この手法によって、スレッドカーネルの処理の粒度がアンバランスである場合において、GPGPU 内の各ブロックに割り当てられる負荷をバランスさせることが可能であることが確かめられた。

研究成果の概要（英文）：Task-based fine-grain execution mechanism is proposed in order to use GPGPUs for non-uniform applications. It can balance the processing load on intra GPU and inter GPUs. We can find out that this method can improve total balance of load which is assigned to the block of GPGPU, even when the grains of parallel execution unit in a thread kernel are not uniformly balanced.

## 交付決定額

（金額単位：円）

	直接経費	間接経費	合計
交付決定額	2,800,000	840,000	3,640,000

研究分野：総合領域

科研費の分科・細目：情報学・計算機システム・ネットワーク

キーワード：GPGPU、非定型処理、スーパースレッドカーネル

## 1. 研究開始当初の背景

多数の計算ノードを有するグラフィック処理ユニット（GPU）が、多くの科学技術分野における計算のプラットフォームとして重要な位置を占めるようになってきている。しかしながら、負荷分散や GPU が本来持っている膨大な計算リソースをいかに有効に使うかという点において、現状での GPU のプログラミング環境やシステム環境は必ずしも満足すべきものではなく、負荷のバランスがうまくとれていない計算を効率的に処理することができないし、十分な並列性を表現できない非定型的な問題に対しては、その計算パワーを有効に活用することができないという問題がある。

また、例えば、NVIDIA 社の GPGPU の開発環境 CUDA を用いた処理などにおいては、汎用プロセッサを用いたホスト側から GPGPU をデバイスとして呼び出して処理を依頼するという形態をとっており、その間の呼び出しやデータ転送のオーバーヘッドをいかに削減するかという点も全体の処理性能の向上にとって重要なテーマとなっている。

## 2. 研究の目的

本研究の目的は、GPGPU をより汎用的で非定型的な処理へ応用するために、従来のスレッドカーネルではなく、タスクベースの細粒度実行機構を提案し、それによって GPU

内および複数の GPU 間における処理の負荷を動的にバランスさせる手法に関して研究を行うものである。また、そのことによってホストと GPGPU 間のデータのやりとりなども削減することが可能となり、処理の効率を高めることができる。

CUDA などにおける一般的な GPGPU のプログラム実行方式は、下の図に示すように、ホストプロセスが GPGPU の計算に必要な前処理を行い、計算に必要なデータを転送し、

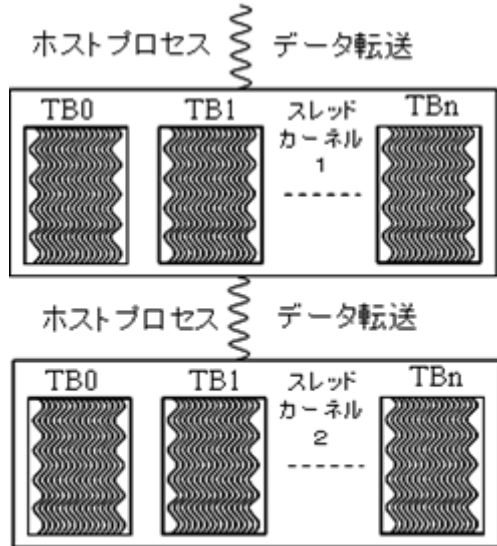


図1. 一般的なGPGPUにおけるプログラム実行方式

GPGPU 内のスレッドカーネルを呼び出し、その結果を受けて、別のスレッドカーネルを呼び出すという形をとっている。このような形態は、非常に多くのデータ並列性が引き出せるアプリケーションには適しているが、タスク並列的なプログラミングには適していない。

そこで、本研究では、下の図に示すように、複数のスレッドカーネルを束ねた統合したカーネル（スーパースレッドカーネルと呼ぶ）を作成し、ホストプロセスとのデータ転送を極力行わずに、スレッド実行の並列性を維持しつつメモリ転送のオーバーヘッドを抑えた実行方式に関して研究を行う。これを外

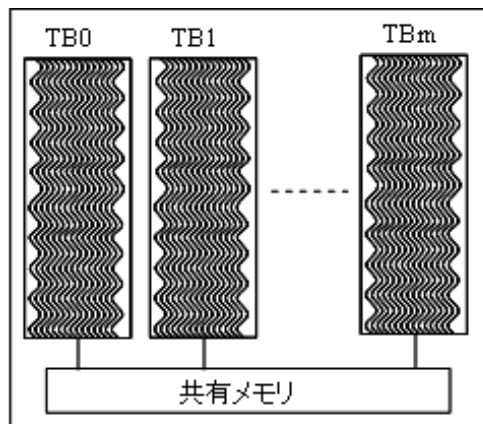


図2. 本提案におけるプログラム実行方式

部から制御するものが、タスク制御機構である。これによって、GPGPU 上での実行において汎用的なタスク並列プログラムからも、十分な並列性を引き出すことが可能となる。

### 3. 研究の方法

新たに、スーパースレッドカーネルという考え方を考案し、これを外部からのタスク制御機構によって制御することによって GPGPU におけるデータ並列処理を生かしつつ、全体的な処理効率を高める。そのために、つぎの3つを考案した。

- (1) GPGPU 内で実行される、複数のスレッドカーネルをまとめて、統合したカーネルとして再構成する。
- (2) 統合したスレッドを外部からのタスク制御機構によって効率的に制御する。
- (3) タスク制御機構に、データ駆動的な待ち合わせ機構を導入する。

これらのメカニズムを利用することにより、まず、複数のカーネルスレッドをまとめて1つの大きな仮想的なカーネルスレッドとして、再構成するための方式について研究を行う。さらに、統合された仮想スレッド内に埋め込まれた複数のスレッド断片を効率的に実行させる制御方式について研究を行う。ここでは、データ駆動的な待ち合わせ機構を用いることにより、仮想スレッドと外部のタスク制御機構との間で制御信号をやりとりすることにより、効率的なスレッド実行をサポートする機構を開発しその性能を評価する。最後に、これらのメカニズムを CUDA の開発環境に組み込むかあるいはプリプロセッサとして実現し、いくつかのベンチマークプログラムによってその実行性能を評価する。

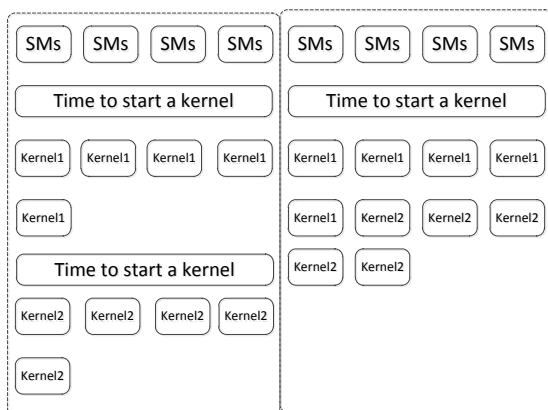
また、非定型的なアプリケーションの1つとして、ソーティングや Packrat Parsing をとりあげ、その並列処理の方策に関して研究をすすめる。

### 4. 研究成果

本研究では、タスクベースの細粒度実行機構によって GPU 内および複数の GPU 間における処理の負荷を動的にバランスさせる手法に関して研究を行い、次のような成果が得られた。

- (1) 複数のカーネルスレッドをまとめて1つの大きな仮想的なカーネルスレッド（スーパースレッドカーネル）として、再構成するための方式についての研究

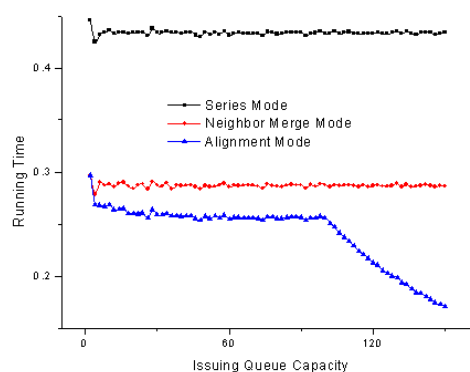
この研究においては、具体的には、まず、複数のスレッドカーネルをまとめて1つの大きな仮想的なスレッドカーネルである、スーパースレッドカーネルとして再構成するための方式について研究を行った。これによって、通常のCUDAにおける処理方式に見られるような、CPUとGPU間での不必要なデータ転送を削減することが可能となる。このとき、スーパースレッドカーネル内の細粒度マイクロスレッド同士の間でのデータ転送を最小化するように、複数のスレッドカーネルを選択する必要がある、このための具体的な戦略や評価手法を研究した。スレッドカーネルを結合するための戦略に関して、いくつかの方式を考案し、その効果について実際にGPGPU上で実験を行った。行列積の計算を行うプログラムを題材として、スレッドカーネルの処理の粒度がアンバランスである場合においても、スレッドカーネルを適切に結合することによって、GPGPU内の各ブロックに割り当てられる負荷がバランスすることを確かめた。



上の図の左に示すように、実験に用いたNVIDIAのGPGPUのプロセッサ単位であるSMのワークロードがアンバランスの場合、すべてのカーネルスレッドが終了してから、次のカーネルスレッドを起動すると、SMの無駄が生じるが、右に示すように各SMで実行されるカーネルスレッドをマージしてスーパーカーネルとすることにより、効率的な並列処理が実現できる。これを確かめるために、様々なワークロードを行列積の計算の大きさで代用し、それらを1つのカーネルスレッドとみなし、それら2つずつをマージしてNVIDIA TESLA C1060上で実行した結果を次の図に示す。

この図において、横軸はマージ可能な全体として処理すべきスレッドの数を表しており、縦軸は実際に実行された処理時間をスレッド単位で正規化したものである。この図に

おいて、Cudaの本来の実行モードはsequential modeであり、Neighbor Merge Modeは、ただ単に隣り合うスレッドをマージし、Alignment modeは適切なペアを選択するものである。この結果に示されているように、実行時間の削減にかなりの効果があることがわかった。



## (2) スーパースレッドカーネルの実行制御に関する研究

この研究では、スーパースレッドカーネルの実行制御に関して研究を進めた。具体的には、スーパースレッドカーネル内の細粒度マイクロスレッド同士の間でのデータ転送を最小化するための、複数のスレッドカーネルの選択に関する評価手法を研究した。また、GPGPUにおける非定型的なプログラムにおける各スレッドの動的な実行方式に関して評価を行うために、クイックソートのプログラムを取り上げ、プロセッサからの制御によらずに、GPGPU単独で、動的なスレッドの制御を行う手法を開発した。しかしながら、グローバルメモリにおける排他制御のオーバーヘッドなどにより、実行効率としては期待した性能が得られていない。これを解決するための課題としては、メモリ階層を生かしたメモリアクセスの効率的な使用方法を開発するとともに、メモリを介さないスレッドの制御方法を導入する必要がある。本研究によって、スレッドカーネルの処理の粒度がアンバランスである場合において、GPGPU内の各ブロックに割り当てられる負荷をバランスさせることが可能であることが確かめられた。

## (3) Packrat Parserにおける並列処理に関する研究

本研究では、非定型的なアプリケーションの1つの例として、Packrat Parsingを取り上げ、その並列処理手法などに関して研究を

行った。まず、仮想マシン方式によって Packrat Parsing のパーザジェネレータを実装した。このとき、Packrat Parsing の根幹要素であるメモ化処理をマルチスレッドで実装することにより並列処理を行い、実際に使われているプログラム言語を構文解析して処理性能を評価した。さらに、本研究では、この手法の問題点である、解析対象文字列のサイズに比例してメモ化領域が肥大化する点を改善する研究を行った。解析時のメモアクセス回数を用いて、動的にメモ領域の大きさを変化させ解析時間を抑える手法などによって、メモ領域の自動調整を行った結果、バックトラックが頻発している対象に対して、メモサイズと解析時間を抑えることができることを示した。

## 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[学会発表] (計 2 件)

- ① 森永 孝仁, 前田 敦司, 山口 喜教, Packrat Parser のメモ化領域の自動調整手法, 情報処理学会プログラミング研究会(2013年01月15日), AiAi ひろば(鹿児島県奄美市)
- ② 遠藤 仁, 前田 敦司, 山口 喜教, メモ化表へのプリフェッチによる Packrat Parser の並列化, 情報処理学会プログラミング研究会(2011年4月25日), 京都大学

## 6. 研究組織

### (1) 研究代表者

山口 喜教 (YAMAGUCHI YOSHINORI)  
筑波大学・システム情報系・教授  
研究者番号：00312827

### (2) 研究分担者

前田 敦司 (MAEDA ATSUSHI)  
筑波大学・システム情報系・准教授  
研究者番号：50293139