マルチ GPU に対する理論モデル及び GPU アルゴリズムの解析と実現

小山田, 徹 / OYAMADA, Toru

(出版者 / Publisher)
法政大学大学院理工学・工学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 理工学・工学研究科編 / 法政大学大学院紀要. 理工学・工学研究科編
(巻 / Volume)
58
(開始ページ / Start Page)
1
(終了ページ / End Page)
4
(発行年 / Year)
2017-03-31
(URL)
https://doi.org/10.15002/00014211

法政大学大学院理工学・工学研究科紀要 Vol.58(2017 年 3 月）　　法政大学

マルチ GPU に対する理論モデル及び GPU アルゴリズムの解析と実現

Proposal of a theoretical model for multi-GPU and analysis and realization of GPU algorithm

小山田　徹
Toru Oyamada
指導教員　和田　幸一

法政大学大学院理工学研究科応用情報工学専攻修士課程

The GPU (Graphics Processing Unit) has high computing power.The technology called GPGPU which uses GPU for general purpose computation is used in various fields.In this research, the theoretical model of multi GPU with plural GPUs construct, analyze and implement algorithms that operate on multi GPU.
Keyword: multi GPU, theoretical model

1. はじめに

グラフィックの画像処理を行うコンピュータの部品に GPU(Graphics Processing Unit) があり,GPU は強力な計算能力を持っている.GPU を汎用計算に用いる GPGPU と呼ばれる技術は様々な分野で用いられている.本研究では GPU を複数搭載したマルチ GPU の理論モデルを構築し,マルチ GPU 上で動作するアルゴリズムの解析と実装を行う.

2. GPU の特性

GPU にはグローバルメモリとシェアードメモリの 2 つのメモリが存在しそれぞれ特徴がある[1]. グローバルメモリは容量は大きいが，アクセスレイテンシが大きい．シェアードメモリは容量は小さいがアクセスレイテンシも小さい. GPU のプログラミングにおいてはグローバルメモリとシェアードメモリを効率的に使用することが,アプリケーションを高速化するために重要になる.

またコンピュータ 1 台に複数の GPU を搭載し，並列に動作させる技術をマルチ GPU という．このマルチ GPU の実行モデルは，通常時は 1 つのスレッドによって逐次的に実行され，並列処理時に特定の領域(GPU の関数)が 2 つ以上のスレッドにより並列的実行される. これまでのマルチ GPU の研究では,2 枚の GPU を搭載したマルチ GPU を実現し,その効率の分析を行った.また GPU-GPU 間でのデータのやり取りはできなかった．しかし現在扱っているマルチ GPU では GPU を 3 枚搭載し,GPU から GPU へメモリの転送を行うことができるようになっている.

3. 理論モデル

(1)シングル GPU に対する理論モデル

GPU に対する理論モデルには様々なものがある. GPU のグローバルメモリとシェアードメモリの特徴をよく捉えている UMM,DMM と呼ばれるモデルと UMM, DMM を組み合わせて 1 つの

GPU として捉えたモデル[2],多くの GPU に共通する特徴を抽象化した並列計算モデルである AGPU モデル[3],我々の研究室で提案したモデル[4]など様々なモデルが提案されている.本論文で提案するマルチ GPU に対する理論モデルは,以前研究室で我々が提案した GPU に対する理論モデルを発展させたものである.そのためまずは我々の提案した GPU に対する理論モデルを以下の図 1 と表 1 で紹介する.

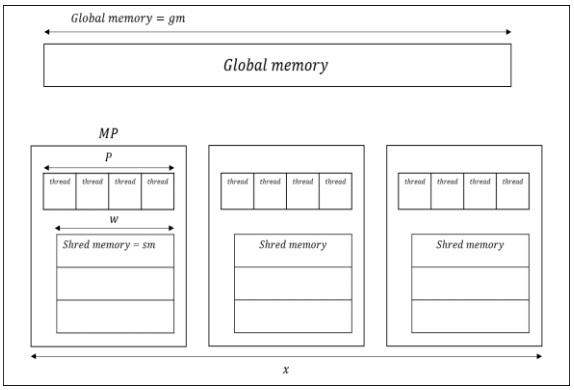

図 1：我々の提案したル GPU の理論モデル

表 1：理論モデルのパラメータ

1MPで使用するスレッド数	p
ワープ数	w
MP数	x
グローバルメモリへのアクセス時間	Lgm
シェアードメモリへのアクセス時間	Lsm
グローバルメモリ数	gm
MPあたりのシェアードメモリ数	sm
要素数	N

(2)理論モデルのメモリアクセス

GPU へのメモリアクセスは 1 ワープ単位で w 個のスレッドが同時にアクセスし,w 個のメモリアクセスを連続して処理する場合はパイプライン方式で処理される.グローバルメモリには GPU のどのスレッドからもアクセスすることができ,シェアードメモリには同じ MP 内にあるスレッドからしかアクセスすることができない.またグローバルメモリへのアクセスはワープの w 個のスレッドが同時にアクセスする.このようなアクセスはコアレッシングアクセスと呼ばれる.この理論モデルではコアレッシングを毎回行うものとする. シェアードメモリは 1 ワープのスレッド数と同じ数の w 個の物理的なバンクに分かれており,すべて異なるバンクへのアクセスならば,すべてのメモリアクセスは 1 度に行われる.同じバンクにアクセスが集中することをバンクコンフリクトと呼び,バンクコンフリクトを起こさないようにアルゴリズムを考える必要がある.

(3) マルチ GPU に対する理論モデル

マルチ GPU に対する理論モデルは,基本的にはシングル GPU に対する理論モデルを組み合わせたものであり,MP(マルチプロセッサ)の数や MP 内で同時に実行できるスレッドの数,グローバルメモリとシェアードメモリの容量やレイテンシなどを考慮している.それに GPU の枚数や CPU-GPU 間,GPU-GPU 間の通信などを考慮した構成になっている. また各 GPU のグローバルメモリに入力を分割して配置できるため,評価するアルゴリズムによって初期入力を変えることができる. マルチ GPU の理論モデルとパラメータを以下の図 4 と表 2 に示す.

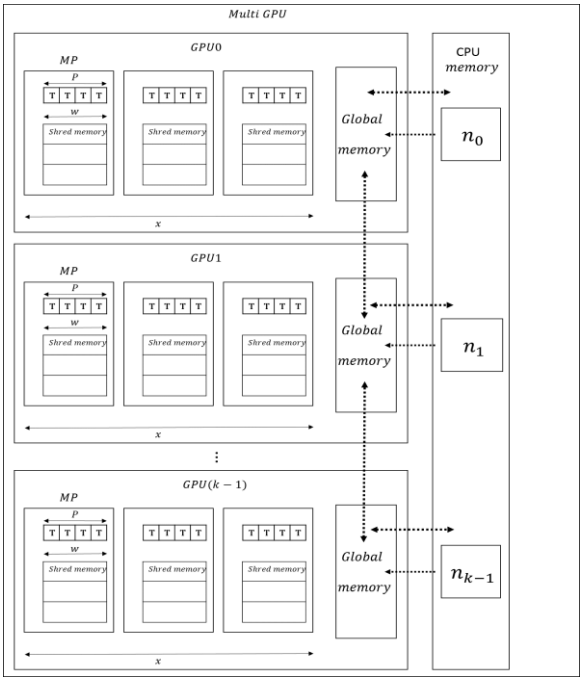

図 4：マルチ GPU に対する理論モデル

表 2：マルチ GPU の理論モデルのパラメータ

1MPで使用するスレッド数	p
ワープ数	w
1GPUのMP数	x
グローバルメモリへのアクセス時間	Lgm
シェアードメモリへのアクセス時間	Lsm
1GPUのグローバルメモリ数	gm
MPあたりのシェアードメモリ数	sm
他のGPUのグローバルメモリへメモリ転送時間	Lgg
CPU･GPU間のメモリ転送時間	Lcg
GPUの数	k
GPUiの要素数	n_i
全体の要素数	n=n_0+n_1+…+n_{k-1}

4. 実装と評価

本研究ではマルチ GPU 上で実装するアルゴリズムとして畳み込み計算を扱う. 畳み込み計算とは. サイズが E の入力配列 x とサイズが F+E-1 の入力配列 y, サイズが N の出力配列 z がある場合,z[i] = x[0]*y[i]+ x[1]*y[i+1]+….+x[m-1]*y[i+m-1] (0≦i≦n-1)を計算する問題である.また F>>E であるとする.

また本研究で行う畳み込み計算の入力サイズは GPU のグローバルメモリに入り切らないほど大きいものとし,CPU-GPU 間のデータの転送と GPU 内部での計算を繰り返す.それをシングル GPU,マルチ GPU でグローバルメモリのみを使用する場合とシェアードメモリを使用する場合の理論値は以下の表 3,表 4 のようになる.

表 3：理論モデルを用いたシングル GPU 上でのグローバルメモリのみを使用した場合の畳み込み計算の理論値

グローバルメモリ	$\left\lceil \frac{F}{f} \right\rceil * (\mathrm{E} * (\frac{f}{wx} + \frac{f}{xp} Lgm - \frac{f}{xp}) + 2*\mathrm{f}*\mathrm{Lcg})$ (gm≧2E+2f)
シェアードメモリ	$\left\lceil \frac{F}{\mathrm{f}} \right\rceil * (\mathrm{E}* (\frac{f}{wx} + \frac{f}{xp} Lsm - \frac{f}{xp}) + \frac{\mathrm{f}}{x*f_s} *(\frac{2sm}{w} + Lgm + Lsm - 1) + 2*\mathrm{f}*\mathrm{Lcg})$ (gm≧2E+2f sm≧2e_s+2f_s)

表 4：理論モデルを用いたマルチ GPU 上でのシェアードメモリを使用した場合の畳み込み計算の理論値

グローバルメモリ	$\left\lceil \frac{F}{f*\mathrm{k}} \right\rceil * (\mathrm{E} * (\frac{f}{wx} + \frac{f}{xp} Lgm - \frac{f}{xp}) + 2*\mathrm{f}*\mathrm{Lcg})$ (gm≧2E+2f)
シェアードメモリ	$\left\lceil \frac{F}{f*\mathrm{k}} \right\rceil * (\mathrm{E} * (\frac{f}{wx} + \frac{f}{xp} Lsm - \frac{f}{xp}) + \frac{\mathrm{f}}{x*f_s} * (\frac{2sm}{w} + Lgm + Lsm - 1) + 2*\mathrm{f}*\mathrm{Lcg})$ (gm≧2E+2f sm≧2e_s+2f_s)

シングル GPU 上,マルチ GPU 上で実際に畳み込み計算アルゴリズムを動かしたときのグローバルメモリのみを使用する場合と,シェアードメモリを使用する場合の理論値と実測値は以下の図 5,図 6 のようになった.

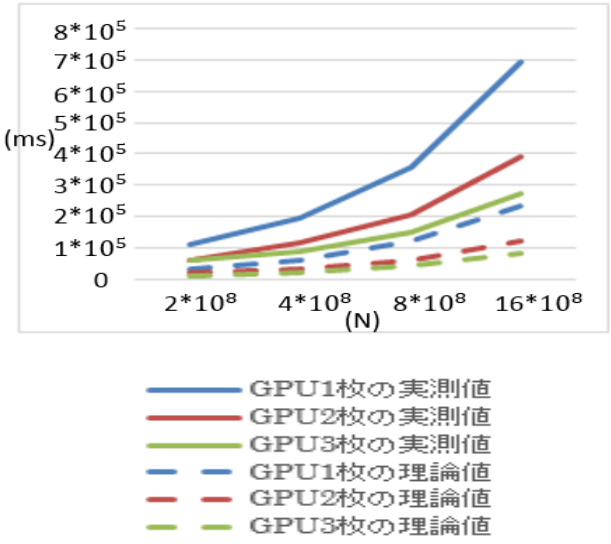

図 5：グローバルメモリのみを使用した場合の畳み込み計算の理論値と実測値の比較

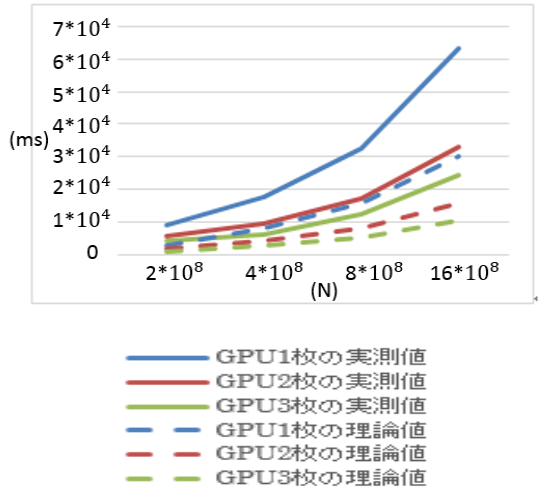

図 6：シェアードメモリを使用した場合の畳み込み計算の理論値と実測値の比較

理論値の計算では GPU でのメモリアクセスに焦点を当てて導出していて,またアルゴリズムの性質上 GPU の関数をも呼び出す毎にレイテンシが発生するため,理論値と実測値で開きがでた.シェアードメモリを使用する場合の理論値と実測値の差も同様の理由で発生していると思われる.GPU の枚数が 2,3 枚と増えるほど,計算時間もほぼ$\frac{1}{2}$, $\frac{1}{3}$になっている.しかし GPU が増えるほどレイテンシも増えるので GPU1 枚あたりの効率は GPU が増える毎に少しづつ下がっていくと思われる.

5. 結論

本研究のマルチ GPU では GPU の搭載数を増やし,また GPU-GPU 間でのメモリ転送が行えるようになり,以前のマルチ GPU よりも効率的な計算ができるようになった.

また,GPU-GPU 間通信には他の方法があり,また今後も発表されると思うので,そういった技術を組み込むことで,理論モデルの更なる改良をすることができると考えられる.

謝辞：本研究を進めるにあたり,ご指導を頂きました和田幸一教授および研究室の皆様に感謝致します.また,大阪府立大学　の藤本典幸教授に心より御礼申し上げます.

参考文献

1) Jason Sanders, Edward Kandrot：CUDA by example 汎用プログラミング,出版, インプレスジャパン(2011)

2) Koji Nakano：The Hierachical Memory Machine Model for GPUs, 2013 IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum (2013-5)

3) 小池敦, 定兼邦彦 : GPU を用いた並列ソートアルゴリズム, 第 10 回情報科学ワークショップ (2014-8)

4) 鈴木, 遠藤, 和田 : GPGPU に対する理論モデル, 2013 電子情報通信学会総合大会, ISS 特別企画, 学生ポスターセッション, DK-1 (2013-03)

発表学会

小山田,和田,藤本：マルチ GPU に対する理論モデル及び GPU アルゴリズムの解析と実現, 2017 年電子情報通信学会 ISS 特別企画 学生ポスターセッション(2017-3 予定)