| | |
|---|---|
| 氏　　名 | 小池 敦 |
| 学位(専攻分野) | 博士(情報学) |
| 学 位 記 番 号 | 総研大甲第 1797 号 |
| 学位授与の日付 | 平成27年9月28日 |
| 学位授与の要件 | 複合科学研究科 情報学専攻<br>学位規則第6条第1項該当 |
| 学 位 論 文 題 目 | A Computational Model and Algorithms to Utilize GPUs for Discrete Problems |
| 論 文 審 査 委 員 | 主　査　教授 胡 振江<br>教授 宇野 毅明<br>教授 河原林 健一<br>教授 合田 憲人<br>教授 定兼 邦彦 東京大学大学院<br>教授 青木 尊之 東京工業大学 |

(別紙様式 2)
(Separate Form 2)

# 論文内容の要旨
## Summary of thesis contents

プロセッサの動作周波数の向上は限界を迎えており，周波数向上に代わるパフォーマンス向上の手段として，並列アーキテクチャが注目されている [1]．パフォーマンス向上が周波数向上により行われる場合，多くのアプリケーションはソフトウェアの修正を行うことなくその恩恵を受けることができる．しかし，並列アーキテクチャを活用する場合には，新しい並列アルゴリズムが必要となる．

GPU は元々はグラフィック処理のための専用プロセッサとして開発された．しかし，高い並列性と広いメモリバンド幅を備えていることにより，グラフィック処理以外の汎用の処理に GPU が使われ始めている．汎用処理への GPU 活用は GPGPU または GPU コンピューティングと呼ばれており，多くの分野で注目を集めている．NVIDIA 社の最新の商用 GPU では，コア数は 3072，メモリバンド幅は 337GB/s に達している．また，GPU は多くのスーパーコンピューターで活用されている．2012 年 11 月の Top500 で 1 位となった Titan もその一つである．さらに，GPU は非常に省電力で動作する．2014 年 11 月の Green500 [2]では上位 10 マシンのうち，9 マシンが GPU を活用している．

GPU は多数のコアを用いて効率よく処理を行うため，特殊なアーキテクチャとなっている．GPU プログラミングにおいては，このアーキテクチャを適切に考慮する必要がある．NVIDIA 社は GPGPU のための開発環境として，CUDA [3]を提供しており，CUDA 上で開発を行うことにより，様々な GPU アーキテクチャ上で動作するプログラムを実装することができる．しかし，最適なパフォーマンスを得るためには，GPU アーキテクチャを適切に考慮してアルゴリズムを設計する必要がある．

逐次アルゴリズムの評価では，RAM(Random Access Machine)モデル上での漸近解析が一般的に行われている．漸近解析により得られた計算量からは実計算時間はわからないにもかかわらず，以下の理由で非常に有益である．第一に RAM モデルはすべての逐次実行マシンに対する抽象化となっており，RAM モデルを用いて漸近解析を行うことで，デバイスの仕様に依らない汎用的なアルゴリズムの性能を知ることができる．さらに計算量の下界が存在し，アルゴリズムの計算量との比較を行うことができる．第二に実際にプログラムを実行することなくアルゴリズムの性能を評価できる．第三にパラメーターの値がパフォーマンスにどのように影響するかを見積もることができる．これらの理由により，RAM モデルを用いることで，アルゴリズムの設計を実装やコードのチューニングとは独立して行うことができる．

一方，並列実行マシンには，RAM モデルのような共通の抽象化が存在しない．並列アルゴリズムの漸近解析に一般的に使用されているモデルに PRAM モデル [4]がある．PRAM は複数のコアと一つの共有メモリからなる．しかし，PRAM は GPU アーキテクチャとは大きく異なっており，GPU 向けアルゴリズムの性能を正しく評価できない．例えば，GPU のマルチプロセッサ内の b 個のコアはメモリ上の b 個の連続した要素を並列に読み取ることができる．つまり，計算量は 1 であり，これは PRAM モデルでの解析結果と一致する．しかし，b 個の要素が連続的に配置されていない場合，計算量は 1 から b の間で変化する．このような場合には GPU に特化した計算モデルが必要となる．

GPU に特化した計算モデルは幾つか提案されている [5, 6]が，これらでは，GPU は SIMD アーキテクチャを持つと仮定されている．それゆえ，これらのモデルでは，GPU の

最も重要な特徴の一つであるマルチスレッディングについて考慮することができない．さらにこれらのモデルと他の既存計算モデルとの関係も不明瞭である．

一方で, GPU 上でのプログラムの実行時間を高精度に見積もるためのモデルも提案されている [7, 8]．これらのモデルを用いることで，実 GPU 上でプログラムを実行することなく，計算時間を見積もることができる．これはコードをチューニングする際に有効である．

本博士論文では，まず，AGPU モデルと呼ばれる並列計算モデルを提案する．AGPU モデルは 3 つのパラメーターのみを用いて GPU の SIMT アーキテクチャを抽象化している．AGPU モデルは GPU 向けアルゴリズムの漸近解析をすることに特化している．AGPU モデルではアルゴリズムの時間計算量，I/O 計算量，多重度の 3 つについて漸近解析を行うことにより，アルゴリズムの計算時間を評価する．AGPU モデルを用いて漸近解析を行う目的はパフォーマンスのボトルネックを見つけることである．そのためにモデルはシンプルながらもコアレッシングやバンクコンフリクト，マルチスレッディングなどのパフォーマンスに影響を与える要因について考察できるようになっている. GPU 上の計算量はデバイス仕様に依存するが，AGPU モデル上の計算量の定数倍に抑えられる．さらに，AGPU モデルは他の多くのモデルと関係を持っており，アルゴリズムを設計する際に他のモデルでの知見を活用することができる．

次に，離散問題に対する基本的なアルゴリズムとしてリストに対するリダクションアルゴリズム，プレフィックススキャンアルゴリズム，比較ソートアルゴリズムを扱う．効率のよいアルゴリズムを提案するだけでなく，AGPU モデルを用いたアルゴリズムの開発方法についても示す．すなわち，既存アルゴリズムの計算量と計算量の下界を比較することにより，パフォーマンスのボトルネックを発見し，それを取り除くことで新しいアルゴリズムを提案する．

リダクションアルゴリズムに関しては，まず，既存の標準アルゴリズム [9]である Tree-based アルゴリズム，Cascading アルゴリズムについて解析を行い，Cascading アルゴリズムが Tree-based アルゴリズムよりも高速であることの証拠を与える．すなわち Tree-based アルゴリズムの時間計算量は Cascading アルゴリズムの時間計算量よりも大きくなっている．Cascading アルゴリズムは最適な時間計算量，I/O 計算量を持つが，演算子が非可換の場合には使用することができない．そこで，非可換な演算子に対応するアルゴリズムとして Pipeline アルゴリズムを提案する. Pipeline アルゴリズムも最適な時間計算量，I/O 計算量を持っており，GPU 上での処理速度も Tree-based アルゴリズムよりも高速である．多くの問題が非可換演算子でのリダクションとして解くことができることが知られている．最大部分和問題もその一つであり，この問題は多くの実用的な応用を持っている．最大部分和問題の比較実験を行い，Pipeline アルゴリズムは Tree-based アルゴリズムよりも最大 3.9 倍高速であり，また，CPU を用いた逐次アルゴリズムよりも最大 29 倍高速であることを示す．さらに, AGPU モデルの既存モデルに対する優位性を示すために Matrix-based アルゴリズムについても解析を行う．Matrix-based アルゴリズムは SIMD アーキテクチャ上で最適な時間計算量，I/O 計算量を持つものの，GPU 上では非常に低速である. AGPU モデルを使うことでその原因を説明できる. すなわち, Matrix-based アルゴリズムでは多重度が小さくなっている．

次に, GPU 上で最も高速なプレフィックススキャンアルゴリズム [10]について, AGPU モデルを用いて解析を行う．このアルゴリズムはチューニングパラメーター $\alpha$ を持ってい

る．AGPU モデルを用いてアルゴリズムを解析することで，$\alpha$ の値に関して，時間計算量と多重度のトレードオフがあることを示す．そして，様々な $\alpha$ の値を用いて実験を行い，計算時間が AGPU モデルでの解析結果と同様の傾向となることを示す．

最後に，比較ソートアルゴリズムを扱う．GPU-Warpsort [11]は最高速のアルゴリズムの一つであるが，AGPU モデルを用いて計算量と計算量下界の解析を行うと， I/O 計算量は最適ではないことがわかる．そこで，新しいアルゴリズムを提案する．提案アルゴリズムは最適な I/O 計算量を持つだけでなく，実際にも高速である．実験を行い，提案アルゴリズムは GPU-Warpsort よりも最大で 1.9 倍高速であることを示す．

## References

[1] Herb Sutter. The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software. Dr. Dobb's Journal, 30(3), March 2005.

[2] Sushant Sharma, Chung-Hsing Hsu, and Wu chun Feng. Making a case for a green500 list. In IEEE International Parallel and Distributed Processing Symposium (IPDPS 2006)/ Workshop on High Performance - Power Aware Computing, 2006.

[3] NVIDIA Corporation. NVIDIA CUDA C programming guide version 4.2, 2012.

[4] Steven Fortune and James Wyllie. Parallelism in random access machines. In Proceedings of the tenth annual ACM symposium on Theory of computing, STOC '78, pages 114–118, New York, NY, USA, 1978. ACM.

[5] K. Nakano. The hierarchical memory machine model for gpus. In Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2013 IEEE 27th International, pages 591–600, 2013.

[6] Nodari Sitchinava and Volker Weichert. Provably efficient gpu algorithms. CoRR, abs/1306.5076, 2013.

[7] K. Kothapalli, R. Mukherjee, M.S. Rehman, S. Patidar, P.J. Narayanan, and K. Srinathan. A performance prediction model for the cuda gpgpu platform. In High Performance Computing (HiPC), 2009 International Conference on, pages 463 –472, dec. 2009.

[8] Sunpyo Hong and Hyesoon Kim. An analytical model for a gpu architecture with memory-level and thread-level parallelism awareness. In Proceedings of the 36th annual international symposium on Computer architecture, ISCA '09, pages 152–163, New York, NY, USA, 2009. ACM.

[9] Mark Harris. Optimizing parallel reduction in cuda, 2008.

[10] Yuri Dotsenko, Naga K. Govindaraju, Peter-Pike Sloan, Charles Boyd, and John Manferdelli. Fast scan algorithms on graphics processors. In Proceedings of the 22nd annual international conference on Supercomputing, ICS '08, pages 205–213, New York, NY, USA, 2008. ACM.

[11] Xiaochun Ye, Dongrui Fan, Wei Lin, Nan Yuan, and P. Ienne. High performance comparison-based sorting algorithm on many-core gpus. In Parallel Distributed Processing (IPDPS), 2010 IEEE International Symposium on, pages 1–10, 2010.

(別紙様式 3)
(Separate Form 3)

# 博士論文の審査結果の要旨
# Summary of the results of the doctoral thesis screening

審査委員全員出席の下、博士学位請求論文(以下、「論文」)の内容についてスライドを用いて約 45 分間の口頭発表を行った後、質疑応答を 25 分間行った。

GPU はグラフィック処理のための専用プロセッサとして開発されたが、高い並列性と広いメモリバンド幅を備えていることにより、汎用処理への GPU 活用は GPGPU または GPU コンピューティングと呼ばれており、多くの分野で注目を集めている。GPU は多数のコアを用いて効率よく処理を行うため、特殊なアーキテクチャとなっているので、GPU プログラミングにおいてはこのアーキテクチャを適切に考慮する必要がある。今まで GPU に特化した計算モデルは幾つか提案されているが、それらの提案においては GPU が SIMD アーキテクチャを持つと仮定されているので、GPU の最も重要な特徴の一つであるマルチスレッディングについて考慮することができない上、これらのモデルと他の既存計算モデルとの関係も不明瞭である。本博士論文はその問題を解決するため AGPU モデルと呼ばれる並列計算モデルを提案した。AGPU モデルは 3 つのパラメーターのみを用いて GPU の SIMT アーキテクチャを抽象化しており、アルゴリズムの時間計算量, I/O 計算量, 多重度の 3 つについて漸近解析を行うことにより, アルゴリズムの計算時間を評価できる。そして、離散問題に対する基本的なアルゴリズムとしてリストに対するリダクションアルゴリズム、プレフィックススキャンアルゴリズム、比較ソートアルゴリズムなどを扱えることを示し、効率のよいアルゴリズムを提案するだけでなく、AGPU モデルを用いたアルゴリズムの開発方法についても与えている。

本博士論文は英語で記述されており、全 7 章から構成されている。

第 1 章と第 2 章は、研究の背景、研究目的、主要な貢献など論文全体の構成、予備知識を述べている。

第 3 章は、AGPU モデルと呼ばれる並列計算モデルを提案している。AGPU モデルは GPU 向けアルゴリズムの漸近解析をすることに特化しており、アルゴリズムの時間計算量、I/O 計算量、多重度の 3 つについて漸近解析を行うことにより, アルゴリズムの計算時間を評価することになっている。GPU 上の計算量はデバイス仕様に依存するが、AGPU モデル上の計算量の定数倍に抑えられ、さらに、AGPU モデルは他の多くのモデルと関係を持っており、アルゴリズムを設計する際に他のモデルでの知見を活用することができるようになっている。

第 4 -6 章は、離散問題に対する基本的なアルゴリズムとしてリストに対するリダクションアルゴリズム、プレフィックススキャンアルゴリズム、比較ソートアルゴリズムを扱っており、効率のよいアルゴリズムを提案するだけでなく、AGPU モデルを用いたアルゴリズムの開発方法についても示している。すなわち、既存アルゴリズムの計算量と計算量の下界を比較することにより、パフォーマンスのボトルネックを発見し、それを取り除くことで新しいアルゴリズムを提案している。

第 4 章は、リダクションアルゴリズムについて述べている。まず、既存の標準アルゴリズムである Tree-based アルゴリズム、Cascading アルゴリズムについて解析を行い、Cascading アルゴリズムが Tree-based アルゴリズムよりも高速であることの証拠を与えている。Cascading アルゴリズムは最適な時間計算量、I/O 計算量を持っているが、演算子が非可換の場合には使用することができない。そこで、非可換な演算子に対応するアルゴリズムとして Pipeline アルゴリズムを提案している。Pipeline アルゴリズムも最適な時間

(別紙様式 3)
(Separate Form 3)

計算量として I/O 計算量を持っており、GPU 上での処理速度も Tree-based アルゴリズムよりも高速である。また多くの問題を非可換演算子でのリダクションとして解くことができることが知られている。最大部分和問題もその一つであり、この問題は多くの実用的な応用を持っている。ここでは最大部分和問題の比較実験を行い、Pipeline アルゴリズムは Tree-based アルゴリズムよりも最大 3.9 倍高速であり、また、CPU を用いた逐次アルゴリズムよりも最大 29 倍高速であることを示している。さらに、AGPU モデルの既存モデルに対する優位性を示すために Matrix-based アルゴリズムについても解析を行っている。Matrix-based アルゴリズムは SIMD アーキテクチャ上で最適な時間計算量、I/O 計算量を持つものの、GPU 上では非常に低速であるが、AGPU モデルを使うことでその原因を説明できる。すなわち，Matrix-based アルゴリズムでは多重度が小さくなっている。

第 5 章は、GPU 上で最も高速なプレフィックススキャンアルゴリズムについて、AGPU モデルを用いて解析を行っている。このアルゴリズムはチューニングパラメーター $\alpha$ を持っており、AGPU モデルを用いてアルゴリズムを解析することで $\alpha$ の値に関して、時間計算量と多重度のトレードオフがあることを示している。そして、様々な $\alpha$ の値を用いて実験を行い、計算時間が AGPU モデルでの解析結果と同様の傾向となることを示している。

第 6 章は、比較ソートアルゴリズムを扱っている。GPU-Warpsort は最高速のアルゴリズムの一つであるが、AGPU モデルを用いて計算量と計算量下界の解析を行うと、 I/O 計算量は最適ではないことがわかる。そこで、新しいアルゴリズムを提案し、最適な I/O 計算量を持つだけでなく実際に高速であり、実験で提案したアルゴリズムは GPU-Warpsort よりも最大で 1.9 倍高速であることを示した。

第 7 章は論文のまとめと今後の課題である。

なお、論文成果は海外学会論文誌 1 件、国際会議論文 2 件として公開されている。

審査会において、出願者は上述の内容に沿って説明を行い、そのあと審査委員との質疑応答を行った。質疑応答では、論文及び口頭発表の内容に関して、AGPU モデルの適用範囲と拡張方法、今後の課題を中心に質問があり、的確な回答がなされた。

以上に基づき審査した結果、6 名の審査委員全員一致で,本学位請求論文は学位を授与するのに十分なレベルであるものと判定された。