

科学研究費助成事業 研究成果報告書

平成 27 年 5 月 28 日現在

機関番号：12102

研究種目：基盤研究(B)

研究期間：2012～2014

課題番号：24300020

研究課題名(和文) ストリーム指向プログラムのマクロ並列化の研究

研究課題名(英文) Research and development of macro parallelization methods targeted to stream-based programs

研究代表者

山際 伸一 (Yamagiwa, Shinichi)

筑波大学・システム情報系・准教授

研究者番号：10574725

交付決定額(研究期間全体)：(直接経費) 13,800,000円

研究成果の概要(和文)：ストリーム指向プログラムはGPUといったメニーコアアクセラレータの普及によって、科学技術計算から産業用製品にまで利用されている。その単体性能は、チップ内における密並列によるプログラム実行により高い性能を示す。しかし、複数のアクセラレータを使った超並列計算を考慮すると、タスクの分割と通信タイミングを配慮したプログラム開発が必要になり、性能をスケーラブルに維持したままの開発が困難となる。本研究では、このようなGPUでのストリーム指向プログラムを対容積・対電力での計算能力の高密度化をねらい、自動的に複数のGPUで並列化し、スケーラブルに性能向上が可能なプログラミング基盤技術を開発する。

研究成果の概要(英文)：The stream-based program has been applied among scientific and industrial computations due to the widely populated manycore accelerators such as GPU. The performance of single GPU is high due to the massively parallel execution of the program. However, considering parallel computing using multiple accelerators, programmers must consider a method for parallelization of the program and its communication timings of the divided programs. Therefore, it is hard to maintain the performance from the parallelized program. This research project aims to develop a programming foundation that can parallelize the stream-based program into multiple GPUs automatically and achieves a scalable performance from the parallel program for GPU.

研究分野：並列分散処理

キーワード：ハイパフォーマンスコンピューティング GPUコンピューティング ストリーム指向

1. 研究開始当初の背景

メモリをランダムアクセスする従来からのプログラミング手法に代わる新たなパラダイムとして、ストリーム指向プログラムが利用されている。特に GPU の数値計算への利用が普及し、OpenCL や CUDA といったプログラミング環境の拡充が後押しする形で科学技術計算から産業用製品にまで利用されている[引用文献]。中でも科学技術計算に関しては、ストリーム指向プログラムのデータ並列性を生かし GPU 内部の小さなプロセッサによる密並列実行により、従来のフォンノイマン型 CPU の性能に比べ、10 - 100 倍の性能を実現している。最近では、複数 GPU をネットワークで接続した GPU クラスタ環境が注目されている。

GPU クラスタでは複数 GPU 間で、ストリーム指向プログラムのマクロな並列化が考慮される。これは、メモリアクセス範囲を分割する従来からのベクトル化とは異なり、データストリームを分割することで並列実行を狙う。この時、GPU 内部のミクロ並列化では必要がない GPU チップ間での通信が必要となる。プログラムは通信ライブラリを使って、GPU 間のデータストリームの移動を実装する。このプログラム開発には、GPU での「ストリーム指向プログラムの実行」と、「GPU 間通信」が混在し、互いの実行タイミングの最適化は困難を極める。通信の最適化を含めた並列化を行い、GPU の潜在性能の 100% 近い実効性能を引き出す新技術が開発できれば、小規模 GPU クラスタ上でさえペタ FLOPS スケールの実効性能を実現するプログラム開発が可能になり、対容積・対電力での計算能力の高密度化が可能になる。

2. 研究の目的

以上の背景から、本研究では、以下の項目を目的とする。

(1) 単体の GPU 向けストリーム指向プログラムをマクロ並列化するコンパイラの開発
GPU 単体の潜在性能を極限まで引き出すことを目的とし、後述の複数の GPU を利用する際にも全体の計算性能が極限まで高められる基盤技術を開発する。

(2) 複数 GPU の利用を可能にする通信コードを自動的に生成するプログラム変換技術
複数の GPU で並列計算をするために、逐次的に記述されたプログラムを並列実行可能な形に変換するための基盤技術を開発する。

(3) 上記 2 つの基盤技術を用い、自然科学等のアプリケーションに応用する

3. 研究の方法

本研究の目的を達成するために、以下の方法で研究を実施した。

(1) 単体 GPU の潜在性能を引き出す技術

GPU と CPU 間のデータ交換に伴うオーバーヘッドの削減方法の開発

GPU と CPU は PCI バスといった周辺バスによって接続されており、CPU が GPU を制御することで、並列計算プログラムとデータをダウンロードして、それを実行する。このデータを再度、GPU 側に転送するオーバーヘッドは GPU でのプログラム実行時間に対して大きく、オーバーヘッドになる。このデータ転送を削減するための方策を開発すると共に、その機能を容易に利用できるプログラミング方式を開発する。

GPU におけるプログラム交換オーバーヘッドを削減するための実行方式の開発

上記の のデータ交換に伴う問題とともに、GPU で異なるプログラムを実行する際にも CPU 側からプログラムをダウンロードし直す必要があり、オーバーヘッドとなっている。このオーバーヘッドを削減するための方法、および、その機能を容易に使えるプログラミングインタフェースを開発する。

(2) 複数 GPU を並列利用するためのプログラム変換方式の開発

プログラムを部分に区切った、または、並列実行するためにスケジューリングされた場合、それらを複数の GPU でパイプライン的に処理する機構を開発する。

(3) 計算物理学への応用

上記の性能改善策を利用し、計算物性物理学のシミュレーションに応用する。

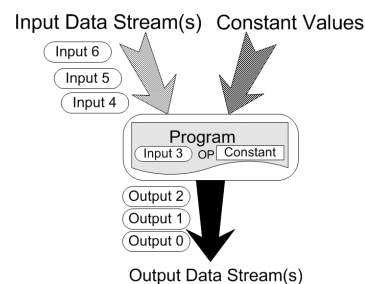


図 1 :Caravela の flow-model

4. 研究成果

本研究から以下の成果が得られた。

本研究では[引用文献]で得ていた研究成果を元にしてさらなる飛躍を目指した。この成果は図 1 に示す、flow-model と呼ばれる GPU 向けプログラムとその入出力を定義したデータ構造を、GPU にマップすることにより並列計算を容易に実行する Caravela と呼ばれるプログラミング環境を開発していた。Flow-model を複数つなげ合わせることでアプリケーションのアルゴリズムを構成し、それぞれの flow-model が単体 GPU にマップされる。この研究基盤を利用し、以下の成果を

得ることができた。

(1) GPU 単体性能におけるホスト CPU との無通信でのプログラム実行方式

Swap 方式と呼ばれる GPU への入出力バッファを、GPU がプログラム実行前後でそのポイント操作のみで入れ替えが可能な方法を使うと、出力データを再利用するような再帰的なプログラムの性能を劇的に向上することができる[引用文献]。このデータ移動に関する最適化のみならず、本研究では、複数の GPU のプログラムをまとめ、スワップ方式と組み合わせることによって、それを GPU 側で連続実行するとプログラムの入れ替えをすることなく、アプリケーションの実行が可能な Scenario-based Execution 方式を開発した[学会発表]。スワップ方式と Scenario-based Execution 方式を使うと、ホスト CPU からのデータやプログラムの移動を一切、行わない、無通信実行が可能になる。これら方式を用い、flow-model の入出力のコネクションを、従来のシェルプログラムのようにスクリプトで記述しておけば、スワップ方式も含め、自動実行し、flow-model を GPU にマップして実行してくれる簡易なプログラム環境 CarSh[学会発表]を開発した。CarSh は GPU といったアクセラレータとされる CPU と協調動作する並列プロセッサを持ったシステムに全て適用でき、さらに、GPU 等をスクリプトだけで実行できるシステムである。CarSh は図 2 に示すようなシステム階層の上に実装され、GPU における詳細なプログラム方法を知らないプログラマでも、GPU のプログラムを用意して、flow-model を定義し、スワップ方式を暗に利用しながら、GPU の潜在性能を引き出した並列計算が可能になった新しいプログラミングインタフェースを提供することができた。

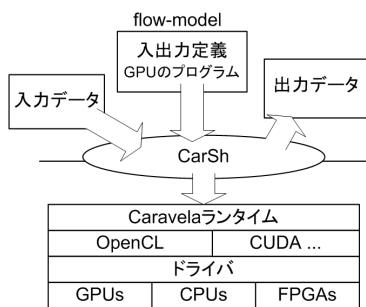


図 2 : CarSh のシステム構成

(2) 複数 GPU での並列実行パイプラインを自動生成するコンパイラ

CarSh で複数の flow-model を接続したアプリケーションが実行できるようになったが、1つの GPU のみを逐次的に使い回す方法で実行していた。それを複数の GPU でパイプライン実行させる自動並列化方式 PEA-ST を開発した[雑誌論文、学会発表]。PEA-ST アルゴリズムは図 3 に示すような複

数の flow-model をつないだアプリケーションから並列性を自動的に抽出し、マクロ並列化を実現する。マクロに並列化された複数の flow-model は複数の GPU に並列に割り当てられ、同時に実行される。図 3 では、この並列化のステップを示している。図 3 (a) は単純に接続された flow-model を並列化している様子をしめす。図 3 (a)-(3) に示すように 3 つの GPU で並列実行でき。さらに、これらの GPU の間での通信を自動生成する。図 3 (b) は結果がフィードバックされる flow-model 間の依存関係がある場合を示している。この場合、どの flow-model が並列実行できるかを目視だけで判断することは難しいが、図 3 (b)-(3) にあるとおり、3 つの GPU で実行できる自動並列化が行われる。

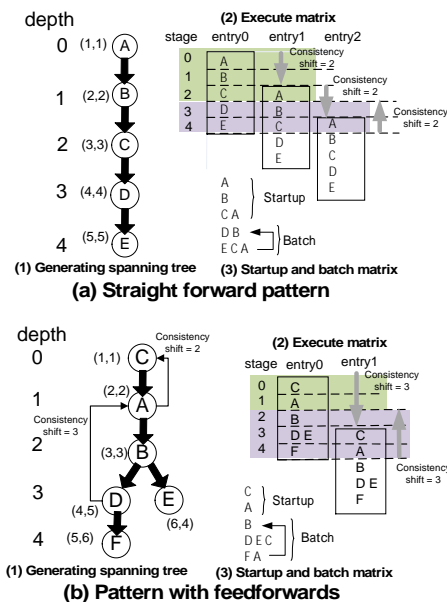


図 3 : PEA-ST で自動的に並列化する例

以上から、本研究の目的にあるマクロ並列化を自動的にを行い、通信までを自動生成するコンパイラの開発ができた。この技術によって、これまでの GPU へのプログラミングは格段に見通しの良いものになると共に、自動的に GPU の潜在能力を引き出すため、細かなチューニングは必要のない新しいプログラミング環境を開発できた。

以上の成果は国内外とも、マクロ並列化というアプローチで GPU のプログラミング環境を構築することは未だ試みられておらず、先駆的な研究である。

(3) 物性物理学への応用

以上の成果を使って、計算物性物理学のシミュレーションに応用した。超伝導物質の温度変化に伴う電子の振る舞いを、Kernel Polynomial Method と呼ばれる固有値解法を使ったものと、モンテカルロ法による方法を使って、GPU クラスタで並列化し、シミュレーションを行った。その性能は、GPU クラス

タにおいて、CPU での並列化に比べ、GPU での並列化が 12 倍にも及ぶことがわかり、十分に計算性能の高密度化ができたことを確認できた[雑誌論文、学会発表]。

このように、スーパーコンピュータなどから比べ小規模な GPU クラスタであっても、十分に性能を引き出すことができるようになったため、スーパーコンピュータの性能を底上げするだけでなく、研究室単位でも高い計算性能を得られる基盤技術となることが今後、予想される。

以上のような成果が得られたが、本研究では以下のような将来への課題も残された。

(1) 近年、組込み型のメニーコアプラットフォームが市場で見られるようになった。これらのプラットフォームでも本研究成果が有効であることを示す必要がある。

(2) 上記の PEA-ST によって並列化の組合せを求めると、膨大な組合せが発生する可能性もある。このような場合のために並列化の条件を、プログラム実行先のプラットフォームの仕様に合わせ決定する機構が必要になる。

(3) 本研究での成果を簡易に使えるよう、GUI といったプログラミングのためのツールを拡充する必要がある。

<引用文献>

David B. Kirk and Wen-mei W. Hwu. Programming Massively Parallel Processors: A Hands-On Approach (1st ed.). Morgan Kaufmann Publishers Inc., 2010.

Shinichi Yamagiwa, leonel Sousa, Caravela: A Novel Stream-Based Distributed Computing Environment, Computer, vol.40, no.5, 2007, pp.70-77.

Shinichi Yamagiwa, Leonel Sousa, Diogo Antao, Data Buffering optimization methods toward a uniform programming interface for GPU-based applications, Proceeding of ACM Intl' Conference on Computing Frontiers, 2007, pp. 205 - 212.

5 . 主な発表論文等

[雑誌論文](計 9 件)

Shinichi Yamagiwa, Guyue Wang and Koichi Wada, Development of an Algorithm for Extracting Parallelism and Pipeline Structure from Stream-based Processing flow with Spanning Tree, International Journal of Networking and Computing, Vol 5,

No 1, 2015, pp.159-179.

<http://www.ijnc.org/index.php/ijnc/article/view/101>

Shixun Zhang, Shinichi Yamagiwa, Seiji Yunoki, A Study of Parallelizing O(N) Green-Function-Based Monte Carlo Method for Many Fermions Coupled with Classical Degrees of Freedom, Journal of Physics: Conference Series (JPCS), 2013, pp. 1-15.

<http://arxiv.org/abs/1303.6016>

Shixun Zhang, Shinichi Yamagiwa, Masahiko Okumura and Seiji Yunoki, Kernel Polynomial Method on GPU, International Journal of Parallel Programming, Volume 41 Number 1, 2012, pp.59-88.

DOI: 10.1007/s10766-012-0204-y

[学会発表](計 6 件)

Guyue Wang, Parallelism Extraction Algorithm from Stream-based Processing Flow applying Spanning Tree, IPDPS/APDCM14, 2014年5月19日, Phoenix (USA).

Shinichi Yamagiwa, Exploiting Execution Order and Parallelism from Processing Flow Applying Pipeline-based Programming Method on Manycore Accelerators, Sixth International Workshop on Parallel Programming Models and Systems Software for High-End Computing (P2S2), 2013年10月1日, Lyon (France).

Shinichi Yamagiwa, Scenario-based Execution Method for Massively Parallel Accelerators, The 11th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA-13), 2013年7月16日, Melbourne, (Australia).

Shinichi Yamagiwa, CarSh: A Commandline Execution Support for Stream-based Acceleration Environment, International Conference on Computational Science (ICCS 2013), 2013年7月5日, Barcelona (Spain).

Shinichi Yamagiwa, Operation Synchronization Technique on Pipeline-based Hardware Synthesis Applying Stream-based Computing Framework, 15th Workshop on Advances in Parallel and Distributed Computational Models/IPDPS2013, 2013年5月20日, Boston (USA).

Shinichi Yamagiwa, GPU-based Parallelization of Kernel Polynomial

Method for Solving LDOS, 3rd Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems / Supercomputing 2012, 2012年11月11日, Salt Lake City (USA).

〔図書〕(計 1件)

Shinichi Yamagiwa, Gabriel Falcao, Koichi Wada and Leonel Sousa, Horizons in Computer Science Research. Volume 11, NOVA Science Publishers, 2015, 80.

〔産業財産権〕

出願状況(計 3件)

名称：プログラム・及び情報処理装置
発明者：山際伸一
権利者：筑波大学
種類：特許
番号：特願 2013 - 144661
出願年月日：平成 25 年 7 月 10 日
国内外の別：国内
名称：アクセラレータ処理実行装置、及びアクセラレータ処理実行プログラム
発明者：山際伸一
権利者：筑波大学
種類：特許
番号：特願 2013 - 109741
出願年月日：平成 25 年 5 月 28 日
国内外の別：国内
名称：ハードウェア設計装置、及びハードウェア設計用プログラム
発明者：山際伸一
権利者：筑波大学
種類：特許
番号：特願 2013 - 105024
出願年月日：平成 25 年 5 月 17 日
国内外の別：国内

6. 研究組織

(1)研究代表者

山際 伸一 (YAMAGIWA Shinichi)
筑波大学・システム情報系・准教授
研究者番号：10574725

(2)研究分担者

和田 耕一 (WADA Koichi)
筑波大学・システム情報系・教授
研究者番号：30175145

中野 浩嗣 (NAKANO Koji)
広島大学・工学研究科院・教授
研究者番号：30281075

柚木 清司 (YUNOKI Seiji)
独立行政法人理化学研究所・柚木計算物性
物理化学研究室・准主任研究員
研究者番号：70532141