

令和元年6月14日現在

機関番号：15401

研究種目：基盤研究(C) (一般)

研究期間：2016～2018

課題番号：16K00171

研究課題名(和文)GPU向けビットレベル並列化による超ハイスループット計算

研究課題名(英文)Ultra high-throughput computation using bit-level parallelization for the GPU

研究代表者

伊藤 靖朗 (Ito, Yasuaki)

広島大学・工学研究科・准教授

研究者番号：40397964

交付決定額(研究期間全体)：(直接経費) 3,400,000円

研究成果の概要(和文)：本研究では、GPU向けビットレベル並列化による超ハイスループット計算手法を開発した。本手法では、大量の入力に対して、ビットレベルで並列計算をおこなうBPBC手法を開発した。BPBC手法とはバルク計算を高速化させるための手法であり、bitwise演算を用いることで高速化を行う。bitwise演算を用いてバルク計算を行うアルゴリズムに対応する組み合わせ回路をシミュレートする。BPBC手法の応用の一例として、CKYパーシングとSmith-Watermanアルゴリズムを並列に実行するGPU実装を行い、逐次処理CPU実装に対してそれぞれ400倍、646倍の高速化を実現した。

研究成果の学術的意義や社会的意義

本研究テーマであるGPU向けビットレベル並列化による超ハイスループット計算手法は、通常、ソフトウェアを専用回路化することで計算の高速化を図るアイデアとは真逆でハードウェア計算をソフトウェアとして計算するものである。本研究テーマで得られた結果より、本手法はGPUのアーキテクチャに対して十分有効で、既存の並列計算手法にない新しいタイプの計算高速化手法である。この手法は既存の専用回路を用いた計算高速化手法に対しても有効である可能性が高く、様々なアプリケーションに対して本手法を用いることで計算高速化が実現可能であると考えられる。

研究成果の概要(英文)：In this research, we proposed a ultra-high-throughput computation method using bit-level parallelization for the GPU. In this method, we developed BPBC technique that employs parallel computation in bit-level for a large number of inputs. BPBC technique is used to accelerate the bulk-computation using bitwise operations. The technique simulates the algorithm of the bulk-computation as combinatorial circuit computation using bitwise operation. We applied BPBC technique to CKY parsing and Smith-Waterman algorithm and implemented them on the GPU. As a result, we attains a speed-up factor of 400 and 646 over the sequential CPU implementation, respectively.

研究分野：計算機科学

キーワード：GPU 並列計算 専用回路 ビット演算

様式 C - 19、F - 19 - 1、Z - 19、CK - 19 (共通)

1. 研究開始当初の背景

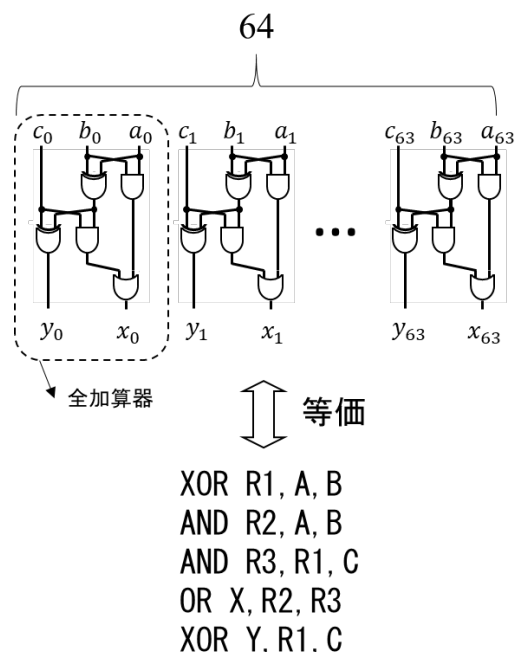
本研究のターゲットデバイスである GPU(Graphics Processing Unit)は、グラフィックス以外の汎用計算に利用する技術 GPGPU (General-purpose computing on GPUs)の研究が盛んに行われている。GPU は本来グラフィックス処理のための補助演算用の LSI であるが、これをグラフィックス以外の汎用計算に利用する GPGPU が注目されており、さまざまな研究開発が行われている。現在の GPU は多数の演算コアと単一の命令スケジューラから成るマルチコアプロセッサが共有メモリで複数接続する構造になっており、SIMT(Single Instruction and Multiple Threads)方式と呼ばれる複数のスレッドが同一の命令を実行する機構になっている。これにより、汎用的なプロセッサと比較して、ハードウェアの構造を簡略化し性能を向上させることができるが、スケジューラが1つしかないため同時に実行するスレッドの命令が異なる場合、その実行が逐次化される。これは汎用的な処理を実行するために必要不可欠な機能であるが、単一の演算コアの性能が高くない GPU では、処理の逐次化は大幅な性能低下をもたらす。つまり、GPU の演算資源を使い切るには少なくとも同時に実行されるすべての命令を揃える必要がある。このような GPU の特徴を考慮し、GPU の性能を最大限に利用する手法として本研究テーマであるビットレベル並列化による計算手法を提案した。

2. 研究の目的

本研究の主な目的は、(1) GPU 向けビットレベル並列化による超ハイスループット計算手法の提案と、(2) 超ハイスループット計算ツールの開発を行うことである。(1) では、論理演算命令を駆使して回路シミュレーションを行い、GPU で数十万オーダーの並列計算を実現し、特定の計算のスループットを上げる手法を提案する。そして、(2) ではその研究成果を元に、回路の最適化手法を取り入れることによって、入力された問題の仕様に対して、上記並列計算のさらなる高速化を図る。

3. 研究の方法

現在ある汎用 CPU や GPU では、基本的に 32 ビットや 64 ビット単位で演算などの処理を実行する(簡単のため、以降 64 ビットで説明する)。これは加算や乗算などの演算命令の基本単位となっており、値の範囲が 10 ビットに収まるとわかっていても 64 ビット単位として扱う必要がある。一方、論理演算命令ではビット毎の演算、つまり、64 回のビット演算を 1 命令で計算する。そこで、論理演算命令を 1 回の論理演算で 64 並列の論理演算可能な並列演算命令と考えることができる。例として、全加算器を考えると、64 個の全加算器は計算を論理演算命令 5 回でシミュレートすることが可能である(右図)。全加算器は、右図のように 5 個の論理ゲートから構成され、3 ビットの値 a 、 b 、 c の入力の合計を計算し、キャリー x と合計 y の 2 ビットの値を出力する。論理演算命令を用いて全加算器の出力を求める場合は、論理ゲート数と同じ回数、この場合、5 回の論理演算命令で計算できる。実際には、ビット毎の演算をしているので、この場合、64 個の全加算器の計算と等価である。このアイデア



より、これは論理演算命令を用いて回路シミュレーションを行うことで、ビット単位で並列実行が可能であることを示している。この回路シミュレーションに基づくビットレベル並列演算を大量のプロセッシングコアを持つ GPU で同時に実行することで、既存のスループット計算を超える超高スループット演算を実現する。

4. 研究成果

GPU 向けビットレベル並列化による超ハイスループット計算手法の一例として、(1) CKY パージングと(2)Smith-Waterman アルゴリズムを並列に実行する GPU 実装を行った。以下にそれらの手法と結果を示す。

CKY パージングは構文解析アルゴリズムの一つで、与えられた文法が入力された文字列を導出可能かどうか判定する。CKY パージングでは、CKY テーブルと呼ばれる二次元テーブルの各要素を動的計画法を用いて計算することで判定を行う処理のことで、この計算をビットレベルで並列化して同時に大量のインスタンスに対して GPU を用いて同時に計算した。このとき、回路設計技術の一つである論理合成による回路サイズの最適化手法を用いることで、巨大な文法を計算する回路サイズを削減し、それに対応する bitwise 演算命令数を削減することに成功した。これはプログラムの命令数に直結することなので、実行時間の削減に寄与している。この文脈自由文法のパージングを行う CKY 処理に対し、ビット演算命令を用いて回路シミュレーションを並列に行うことで、GPU の計算リソースを効果的に使用する実装をおこなった。その結果、32768 の規則を持つ文脈自由文法の CKY パージング処理を長さ 32 文字の 2 百万個の文字列に対し、提案 GPU(NVIDIA GeForce GTX TITAN X)実装は、逐次処理を行う CPU(Intel Core i7-4790) 処理に対して最大 400 倍高速化を実現した。

Smith-Waterman アルゴリズムは動的計画法を用いて、ローカルアライメントを求める計算手法で、バイオインフォマティクス分野において、DNA 配列中に類似した部分列の探索に用いられる。Smith-Waterman アルゴリズムの GPU に実装では、回路シミュレーション計算をビット演算命令で実行するために、入力データをビットレベルで転置し、計算結果を逆転置する必要がある。そのため、命令数を最適化したビットレベル転置手法を提案し、転置および逆転置の計算時間を削減することに成功した。さらに回路最適化手法を Smith-Waterman アルゴリズム計算の回路シミュレーションにも応用することで、ビット演算の命令数及び使用メモリサイズの削減を実現した。以上のアイデアを用いて複数のインスタンスに対して並列に計算する手法を GPU で実装をおこなった。その結果、32768 のシーケンス列に対し、長さ 128 のパターンのローカルアライメント計算を、提案 GPU(NVIDIA Tesla V100)実装は逐次処理を行う CPU(Intel Xeon E7-8870 v4)処理に対して最大 646 倍の高速化を実現した。

以上より、本研究テーマであるビットレベル並列化による超高スループット計算手法は、GPU アーキテクチャの特徴を効果的に利用することを示し、さらに既存の回路設計手法をソフトウェアアプローチである本手法に適応することで、さらなる計算高速化を実現可能であることを示した。

5. 主な発表論文等

〔雑誌論文〕(計 4 件)

the fast Smith-Waterman algorithm using BPBC technique, Bulletin of Networking, Computing, Systems, and Software, Vol.8, No. 1, pp. 50-53, January 2019, 査読無.

<http://www.bncss.org/index.php/bncss/article/view/105>

Takahiro Nishimura, Jacir L. Bordim, Yasuaki Ito, and Koji Nakano, Accelerating the Smith-Waterman Algorithm Using Bitwise Parallel Bulk Computation Technique on GPU, Proc. of International Parallel and Distributed Processing Symposium Workshops, pp. 932-941, May 2017, 査読有.

DOI: 10.1109/IPDPSW.2017.75

Toru Fujita, Koji Nakano, Yasuaki Ito, and Daisuke Takafuji, An Efficient GPU Implementation of CKY Parsing using the Bitwise Parallel Bulk Computation Technique, IEICE Transactions on Information and Systems, Vol. E100-D, No. 12, pp. 2857-2865, December 2017, 査読有.

DOI: 10.1587/transinf.2017PAP0018

Toru Fujita, Koji Nakano, Yasuaki Ito, Bitwise Parallel Bulk Computation on the GPU, with Application to the CKY Parsing for Context-Free Grammars, Proc. of International Parallel and Distributed Processing Symposium Workshops, pp. 589-598, May 2016, 査読有.

DOI: 10.1109/IPDPSW.2016.64

〔学会発表〕(計 1 件)

Takahiro Nishimura, Jacir L. Bordim, Yasuaki Ito, Koji Nakano, A GPU Implementation of the Smith-Waterman Algorithm using Bitwise Parallel Bulk Computation Technique, IEICE Technical Report, Vol. 117, No. 28, IEICE-COMP2017-4, pp. 23-30, May 2017, 査読無.

6 . 研究組織

(1)研究分担者

研究分担者氏名：中野 浩嗣

ローマ字氏名：Koji Nakano

所属研究機関名：広島大学

部局名：工学研究科

職名：教授

研究者番号(8桁): 30281075

科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。