

論文審査の要旨

| | | | |
|---|----------------|-------|-------|
| 博士の専攻分野の名称 | 博 士 (工 学) | 氏名 | 高藤 大介 |
| 学位授与の要件 | 学位規則第4条第1・2項該当 | | |
| 論 文 題 目 A Study on Efficient GPU Implementations to Compute the Diameter of a Graph (グラフの直径計算のための効率的な GPU 実装に関する研究) | | | |
| 論文審査担当者 | | | |
| 主 査 | 教 授 | 中野 浩嗣 | 印 |
| 審査委員 | 教 授 | 藤田 聡 | 印 |
| 審査委員 | 准教授 | 伊藤 靖朗 | 印 |
| 〔論文審査の要旨〕 | | | |
| <p>GPU (Graphics Processing Unit) は内部に多数の演算コアを搭載したグラフィクス処理用のハードウェアであり、並列演算能力に優れている。グラフの全点対間の最短距離を求めるアルゴリズムとしてフロイド・ワーシャルアルゴリズムがよく知られているが、これを並列計算によって高速化するために、ブロック化フロイド・ワーシャルアルゴリズムが提案されている。本論文ではこのアルゴリズムを GPU 上で効率的に実行するための実装手法を提案している。</p> <p>第1章では、本研究の背景と目的を示し、本論文の概要を述べている。</p> <p>第2章では、GPU アーキテクチャ及び統合開発環境 CUDA の用語、諸定義などを説明している。</p> <p>第3章ではブロック化フロイド・ワーシャルアルゴリズムの GPU 実装を高速化し、その性能を評価している。</p> <p>複雑なネットワークの構造解析などのために点と辺で構成されたグラフにモデル化されており、グラフの全点対間の最短距離はよく使用されている。たとえば、全点対間の最短距離の中の最大値はグラフの直径と呼ばれ、ネットワークの通信遅延解析に応用がある。グラフの全点対間の最短距離を求めるアルゴリズムとしてフロイド・ワーシャルアルゴリズムがよく知られているが、計算量が非常に大きい。これを並列計算によって高速化するために、ブロック化フロイド・ワーシャルアルゴリズムが提案され、多くの並列環境で実装されてきた。既存の GPU 実装は複数回の同期処理が原因となり計算時間が大きくなっていたため、本論文では同期処理を減少させた効率的な GPU 実装を提案している。この性能を NVIDIA Tesla V100 GPU 上で評価した結果、既存実装に比べて最大 1.31 倍の高速化を実現した。さらなる高速化のために、1つの命令を同時に複数のデータに適用する並列化手法である SIMD 演算を実装し、提案実装に適用している。その結果、既存実装に比べて最大 1.28 倍の高速化を実現した。1つのグラフに対する全点対間の最短距離を高</p> | | | |

速に計算することに着目したが、異なる複数のグラフが与えられたときにも効率的に同時に計算する GPU 実装を提案している。このように複数の独立した入力に対し、順にまたは並列に計算することを特にバルク計算という。

第 4 章では、まず逐次アルゴリズムのバルク計算を効率的に行う CUDA C プログラムを自動生成するツールを開発している。このツールを使うと逐次アルゴリズムの C 言語プログラムからそのバルク計算を行う CUDA C プログラムに自動で変換でき、生成された CUDA C プログラムを CUDA で動作可能な GPU で実行できる。並列計算の初学者にとって効率的な CUDA C 言語プログラムを作成することは困難であり、このツールは初学者にとっても有効なものである。

アルゴリズムが oblivious であるとは、実行中にアクセスされるアドレスが入力に依存しないものをいう。oblivious なアルゴリズムであるバイトニックソート、フロイド・ワーシャルアルゴリズム、モンゴメリ乗算アルゴリズムに対するバルク計算を NVIDIA GeForce GTX Titan GPU 上に実装し、それらの性能を評価した。その結果逐次実装に比べて、バイトニックソートでは最大 199 倍、フロイド・ワーシャルアルゴリズムでは最大 54 倍、モンゴメリ乗算アルゴリズムでは最大 78 倍の高速化を実現した。

最後に、第 5 章で結論について述べている。

以上、審査の結果、本論文の著者は博士（工学）の学位を授与される十分な資格があるものと認められる。

備考：審査の要旨は、1,500 字以内とする。