

CUDA C++ を用いた高速なアルゴリズムライブラリの作成

学籍番号 202010171 金鹿 優輝

2021 年 7 月 27 日

1 概要

GPGPU (General-Purpose Computing on Graphics Processing Units) により, さまざまな計算を高速化することが可能である. たとえば私が参加したスパコンを用いたコンテスト SupercomputingContest 2018, 2019 では, GPU プログラミングにより量子計算や多体問題のシミュレーションの高速化を実現した. 今回私は, CUDA C++ を用いてワーシャルフロイド法 (Floyd-Warshall Algorithm) のライブラリを実装し, C++ から GPU を用いたアルゴリズムを利用できるインターフェースを作成した. すなわち, C++ プログラムのレイヤーを超え, GPU を用いたプログラムを作成する代理人としての役割を果たすものを作成した. そして, 実際に頂点数 1024 のグラフにおいて, CPU を用いたプログラムに対して 40 倍近くの高速化を実現した.

2 説明

2.1 アイデア

ワーシャルフロイド法は, グラフ上の全点对最短経路 (All-Pairs Shortest Paths) の長さを $O(N^3)$ で求めるアルゴリズムであり, 端的に言えばグラフ上での動的計画法 (Dynamic Programming) の反復である. そしてまた, この動的計画法は非常に単純であり, この方法を少し工夫することにより, 計算量は $O(N^3 \log N)$ へ悪化するが, トロピカル半環 (Tropical Semiring) 上での行列累乗に帰着できることが知られている. トロピカル半環上では通常の行列積のような高速化 (通常の行列積は $O(N^{2.3728639})$) で計算できることが知られている [1]) はできず, 全点对最短経路を $O(N^{3-\epsilon})$, ($\epsilon > 0$) で求める方法は未だ発見されていない [2] が, GPU を利用して, マルチスレッドで計算することで定数倍の改善が可能である.

2.2 デザイン

C++ のクラスを利用することで, GPU プログラムの実装を気にすることなく通常の C++ アルゴリズムライブラリと同様に利用できるようにした. プログラマ側は nvcc をインストールし, ライブラリをインクルードするだけで利用が可能である.

2.3 実装の工夫

NVIDIA が作成した、行列積の計算をシェアードメモリを使用して高速化する非常に素朴なアルゴリズムのソースコード [3] を参考に、シェアードメモリを使用して行列累乗する方針を採用した。また、テスト用のプログラムを何度か実行したところ、スレッドブロックの大きさの平方根 $BS = 12$ のときに GPU プログラムがもっとも高速になることがわかったため、そのように実装した。

2.4 開発したもの

このプログラムは、GitHub[4] で公開している。仕様は以下のようになっている。

apsp.cu (ライブラリのソースファイル)

- apsp::FloydWarshall ワーシャルフロイド法を実装したクラス
- apsp::FloydWarshall::SetDist グラフの距離行列を設定する関数
- apsp::FloydWarshall::GetDist 計算結果を取得する関数
- apsp::FloydWarshall::Calc ワーシャルフロイド法で CPU 上で計算する関数
- apsp::FloydWarshall::DeviceCalc ワーシャルフロイド法で GPU 上で計算する関数

apsp_test.cu (ライブラリのテスト用ファイル)

1. 定義された N の値に基づき、頂点数 N の完全グラフを作成し、その距離を 1 から 1000 までのランダムに設定する。
2. GPU 上と CPU 上でワーシャルフロイド法で全点对最短経路長を計算し、その実行時間を計測し、計算結果が正しいかどうかを検証する。

3 考察

$N = 1024$ として apsp_test.cu を実行し 10 回の平均をとったところ、GPU の平均実行時間は 0.02185914(s), CPU の平均実行時間は 0.8042372(s) となった。なお、使用した CPU は Ryzen7 3700X, GPU は GeForce RTX 3070 である。このことから、 $0.8042372 / 0.02185914 = 36.79180\dots$, すなわち 40 倍近くの高速度が実現されていることがわかる。しかしながら、Web 上の情報によれば、「再帰的なワーシャルフロイド法 [5]」を用いてさらに 10 倍程度の高速度が可能 [6] ということである。このアルゴリズムを実装することで、さらに高速なライブラリを作成できると考えられる。

4 まとめ

CUDA C++ を用いてワーシャルフロイド法 (Floyd-Warshall Algorithm) のライブラリを実装し、C++ から GPU を用いたアルゴリズムを利用できるインターフェースを作成した。そして頂点数 1024 のグラフにおいて、CPU を用いたプログラムに対して 40 倍近くの高速度を実現したが、まだ改善の余地があることがわかった。今私には、再帰的なワーシャルフロイド法のアルゴリズムを理解し、実装することが求められている。

参考文献

- [1] <https://dl.acm.org/doi/10.1145/2608628.2608664>
- [2] http://lealgorithm.blogspot.com/2017/08/blog-post_22.html
- [3] https://www.gsic.titech.ac.jp/supercon/main/attwiki/index.php?plugin=attach&refer=Supercomputing%20Contest%202012%2FGPU%E3%83%97%E3%83%AD%E3%82%B0%E3%83%A9%E3%83%9F%E3%83%B3%E3%82%B0%E8%B3%87%E6%96%99&openfile=matmul_shared.cu
- [4] <https://github.com/shuzaei/cuda-lib>
- [5] <https://escholarship.org/content/qt9v89p5wv/qt9v89p5wv.pdf?t=lnr0ng>
- [6] <https://proc-cpuinfo.fixstars.com/2015/11/all-pair-shortest-path-html/>