

oneAPI を使った FPGA 上でのグラフ探索アルゴリズムの実装

溝谷 祐大[†] 天笠 俊之[‡]

筑波大学 情報学群 情報科学類[†]

筑波大学 計算科学研究センター[‡]

1 始めに

幅優先探索とはグラフ探索アルゴリズムの一種であり、デジタル回路のテスト・検証、道路ネットワークの解析など、幅広い応用分野で利用されている。

近年では、処理対象のグラフが大規模になってきているので FPGA (Field Programmable Gate Array) を利用した高速化手法が提案されている。FPGA とは、任意の回路をプログラミングによって繰り返し実装可能なハードウェアチップである。一方で、幅優先探索は不規則なメモリアクセスが多くなるのでメモリ帯域を有効に利用できないといった問題がある。

そこで本研究では、Intel oneAPI を使った FPGA 上でのグラフ探索アルゴリズムの実装を提案する。

2 関連研究

本研究に関連する 2 つの先行研究について述べる。Umuroglu らの研究 [1] では、幅優先探索の開始の数ステップと最後の数ステップを CPU 上で処理し、それ以外のステップでは FPGA 上で処理するといった手法を提案した。しかし、処理対象のグラフが大規模になると CPU 上での処理が性能のボトルネックとなることや、

メモリコストが大きくなってしまいうことが問題として挙げられる。

Zhang らの研究 [2] では、連続してアクセスされることが多いグラフと強い結合を持つ頂点を見つけ、そのグラフを HMC (Hybrid Memory Cube) に保持する事によりメモリコストを削減する手法を提案した。これによりメモリコストを削減したが、探索中にエッジ係数が小さくなると高速化が働かないという問題がある。

3 前提知識

3.1 幅優先探索

幅優先探索とは、グラフ探索アルゴリズムの一種である。根ノードから隣接した全てのノードを探索し、そこからさらに隣接しているノードに対して同様のことを繰り返して探索対象ノードを見つけることができる。また、根ノードから各ノードへの最短経路も求めることができる。

図 1 はノード 0 を根ノードとした際の幅優先探索の流れを表している。まず、フロンティアテーブルにノード 0 と隣接しているノード 1, 3, 5 が格納される。このときに dist 配列の 1, 3, 5 番目にそれぞれ 1 が格納される。次に、ノード 1, ノード 3, ノード 5 に隣接するノードがフロンティアテーブルに格納され、距離計算を行う。この動作をすべてのノードがフロンティアテーブルに格納されるまで繰り返すことで探索対象ノードとそれまでの最短距離を得ることができる。しかし、不規則なメモリアクセスが発生してしまう。

An implementation of graph search algorithm on FPGA using oneAPI

Yuta Mizotani[†]

Toshiyuki Amagasa[‡]

[†]College of Information Science, University of Tsukuba

[‡]Center for Computational Sciences, University of Tsukuba

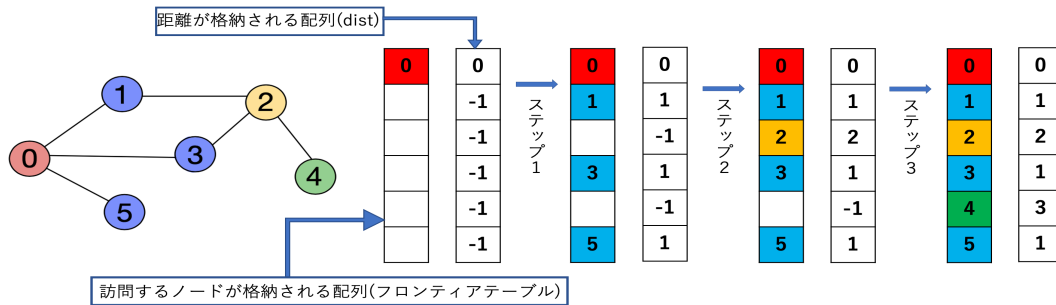


図1 幅優先探索の流れ

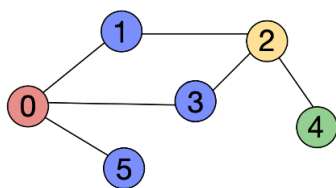
3.2 oneAPI

oneAPIとは、intel社が提供しているクロスアーキテクチャプログラミングモデルである。C++ベースの言語(DPC++)を用いて複数のアーキテクチャ向けソフトウェアを容易に開発できる。また、最適なハードウェアを自由に選択することができる。これによって各アクセラレータに向けた、個別のプログラミングモデルやツールが不要となる。本研究ではこのDPC++を利用して実装を行った。

4 oneAPIで実装する幅優先探索

4.1 データ構造

本研究では、グラフのデータ構造はCSC(Compressed Sparse Column)を利用した。CSCとは2つの配列(ptrs, elements)を利用してグラフを表現する方法である(図2)。



ptrs = {0, 3, 5, 8, 10, 11, 12}
elements = {1, 3, 5, 0, 2, 1, 3, 4, 0, 2, 2, 0}

図2 Compressed Sparse Column

また、フロンティアテーブルは長さがノードの値のbool型の配列を利用した。これは探索条件のノードに隣接している場合に1をそうでない場合に0と表現することができるからである。

4.2 幅優先探索の実装

幅優先探索の実装では通常の幅優先探索と距離生成を切り離して実装した。距離生成するには現在のフロンティアテーブルと次のステップのフロンティアテーブルを各ステップが終了したあとに調べればよいからである。

この切り離しによって通常の幅優先探索、距離生成の並列化が安易になる。幅優先探索の実装ではあるノードの隣接ノードを発見する動作を並列化した。この動作を並列に処理することで幅優先探索が高速に処理することができる。

5 まとめ

本稿ではoneAPIを使ったFPGA上でのグラフ探索アルゴリズムの実装を提案した。今後は評価実験を通して本手法の有用性を評価する予定である。

参考文献

[1] Yaman Umuroglu, Donn Morrison, and Magnus Jahre. Hybrid breadth-first search on a single-chip fpga-cpu heterogeneous platform. In *2015 25th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–8. IEEE, 2015.

[2] Jialiang Zhang, Soroosh Khoram, and Jing Li. Boosting the performance of fpga-based graph processor using hybrid memory cube: A case for breadth first search. In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 207–216, 2017.