

FPGA を用いたグラフに対する正規パス問合せの高速化

三浦 賢人[†] 天笠 俊之^{††} 北川 博之^{††}

[†]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 はじめに

正規パス問合せ (RPQ)[1] を用いることで, Metapath[2] 等の処理に必要な, グラフ中での特定パスの出現箇所を得ることが可能である.

ここで課題としてあげられるのは RPQ 処理の高速化である. 近年のデータ分析での対象データの大規模化の傾向から, RPQ の対象となるグラフも大規模化しており, RPQ の処理の高速化が求められる.

FPGA は任意の論理回路をプログラミングによって自由に実装できるデバイスであり, 任意の処理に特化した演算パイプラインをデバイス上に構成できるため, パイプライン化が効果的な処理では CPU や GPU に比べて高い性能を得ることが可能である.

本研究では FPGA を用いた RPQ 処理の高速化手法を提案する. 具体的には, FPGA 上に RPQ 処理を並列に処理する演算パイプラインを実装し高速化を目指す.

2 前提知識

2.1 正規パス問合せ (RPQ)

正規パス問合せ (RPQ) は, ラベル付きグラフ中の任意の 2 ノード間に指定したパスが存在するかを調べる問合せである. 処理対象となるラベル付きグラフを $G = (V, E, \lambda)$ とする. V はグラフ内の頂点の集合, E はグラフ内のエッジの集合 ($E \subseteq V \times V$) である. さらに, λ はグラフ内のラベルの集合を \mathcal{L} とした場合, エッジ集合 E からラベル集合 \mathcal{L} に対する写像関数 $\lambda: E \rightarrow \mathcal{L}$ である. ここで, $\ell \in \mathcal{L}$ とすると, 以下のような RPQ 評価のセマンティクス R が定められる.

$$R ::= \epsilon | \ell | \ell^- | R \circ R | R \cup R | R^{i,j} \quad (1)$$

ここで, ϵ は空列, ℓ はラベルを順方向に, ℓ^- は逆方向にたどる表現であり, \circ は結合, \cup は和, $R^{i,j}$ は R の k 回 ($i \leq k \leq j$) の繰り返し $\underbrace{R \circ \dots \circ R}_{k \text{ times}}$ を表す. 例として, 図 1 のグラフを G_{ex} とし, $R = a \circ b \circ b$ を適用す

ると

$$R(G_{ex}) = \{(2, 5)\} \quad (2)$$

となる. この RPQ 処理では図中の赤色でハイライトされたパスが検出され, 最終的にその始点・終点ノードの集合である $\{(2, 5)\}$ が得られる.

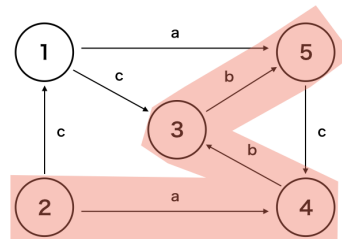


図 1 RPQ の例 ($R = a \circ b \circ b$ を適応)

2.2 FPGA

FPGA(Field Programmable Gate Array) は様々な回路をプログラムによって実装可能なハードウェアチップであり, 実装された回路は application-specific integrated circuit(ASIC) のように動作させることが可能である. 従来, FPGA は Verilog や VHDL のようなハードウェア記述言語によってプログラムされることが一般的であり, 大規模な回路を作成する場合の開発コストの高さが問題視されていた. しかし, 近年では C/C++ や OpenCL 等のより高レベルな言語によって FPGA 上の回路をプログラムできる高位合成 (HLS) といった技術が確立されてきている.

本研究では, HLS を用いて FPGA 上への回路の実装を行なった.

3 関連研究

RPQ の高速化に関する研究の中で代表的なものとして Fletcher らの手法 [3] が挙げられる. この手法ではパスの長さ k を事前に定め, グラフ中に登場する長さ k までのパスの登場を事前に列挙し, インデックスとしてストレージに保存しておく. これによって, 長さ k 以下のパスを指定した場合 ($|R| \leq k$) は, インデックス内を検索するだけで結果を得ることができ, パスの長さが k よりも大きな場合 ($|R| > k$) でもインデックスを用いることで, 効率的に結果を導出することができる. この方法は実行速度の面では効果的だが, ストレージサイズを圧迫してしまうといった問題点も存在する.

Accelerating Regular Path Queries on Graph using FPGA

Kento Miura[†]

Toshiyuki Amagasa^{††} and Hiroyuki Kitagawa^{††}

[†]College of Information Science, University of Tsukuba

^{††}Center for Computational Sciences, University of Tsukuba

4 提案手法

本研究では、FPGA 上で RPQ を並列に処理するための手法を提案する。RPQ 処理を複数のステージに分割し、FPGA 上でパイプライン処理を行う。

4.1 処理の概要

図 2 に提案手法の処理の概要を示す。図の例はユーザから RPQ として $R = a \circ b \circ b$ を与えられた場合を表している、全体の処理は Host 側 (FPGA と通信を行う PC) で行われる処理と FPGA 側で行われる処理に分割される。まず、Host ではユーザから与えられた RPQ に出現するラベルを持つエッジのデータ ($R = a \circ b \circ b$ の場合、ラベル a を持つエッジ群とラベル b を持つエッジ群) を FPGA に転送する。FPGA は受け取ったエッジを RPQ の順番の通りに結合する。最終的に得られた結果は FPGA から Host を通ってユーザへ送信される。

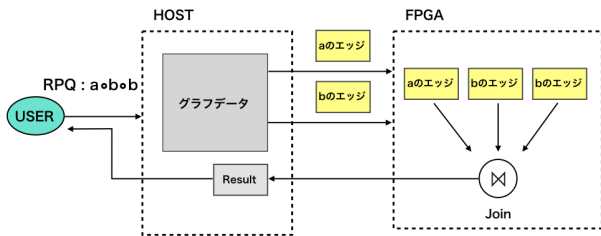


図 2 提案手法の概要 ($R = a \circ b \circ b$ を適応)

4.2 FPGA 上の演算パイプライン

FPGA 上でエッジの結合処理を行うために図 3 に示す 2 種類のモジュールを考案する。

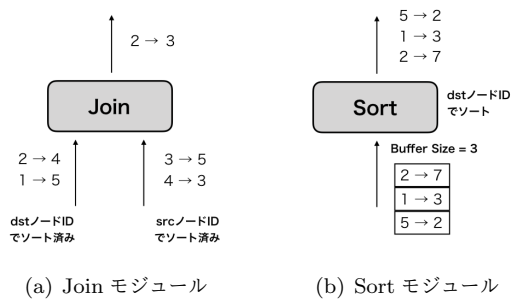


図 3 FPGA 上に実装するモジュール

それぞれのモジュールは複数のエッジデータを受け取る。ここでエッジデータは src ノード ID \rightarrow dst ノード ID と表記する。例えばノード 2 からノード 4 に張られるエッジは $2 \rightarrow 4$ と表される。Join モジュールは dst/src ノード ID でそれぞれ昇順にソートされたエッジデータを受け取りそれらをマージ結合する。図 3(a) の例では $2 \rightarrow 4$ と $4 \rightarrow 3$ のエッジが結合され結果として $2 \rightarrow 3$ が出力される。Sort モジュールは受け取ったエッジデータをバッファに保存し、バッファが満杯になるとエッジデータを dst ノード ID で昇順にソートし、

ソート後のデータを出力する。

次に先ほどの 2 種類のモジュールを用いて FPGA 上で実際にどのように RPQ 処理を行うのかを示す。例として $RPQ : R = a \circ b \circ b \circ c$ が与えられた場合、2 種類のモジュールを FPGA 上で図 4 のような木構造に連結する。

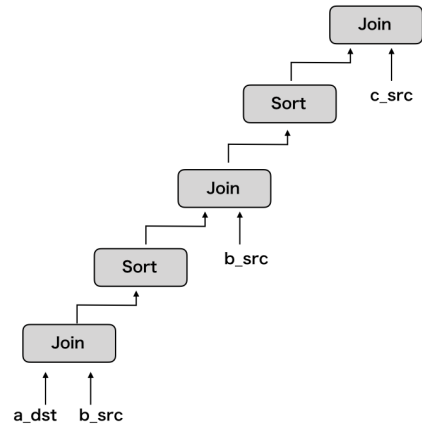


図 4 FPGA 上の演算パイプライン

ここで、 a_dst はラベル a の振られた全エッジを dst ノード ID で昇順にソートしたデータ、 b_src, c_src はそれぞれラベル b, ラベル c の振られた全エッジを src ノード ID で昇順にソートしたデータを表す。このようにモジュールを連結しエッジデータを入力することで、一番上段の Join モジュールの出力は $RPQ : R = a \circ b \circ b \circ c$ の結果となる。

5 まとめ

本稿では FPGA を用いた RPQ 処理の高速手法を提案した。FPGA 上に Join と Sort の二種類のモジュールを実装し、それらを連結させ演算パイプラインを構成した。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] Isabel F. Cruz, Alberto O. Mendelzon, and Peter T. Wood. A graphical query language supporting recursion. *SIGMOD Rec.*, Vol. 16, No. 3, pp. 323–330, December 1987.
- [2] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB' 11*, 2011.
- [3] George H. L. Fletcher, Jeroen Peters, and Alexandra Poulouvasilis. Efficient regular path query evaluation using path indexes. In *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France*, pp. 636–639, March 2016.