

令和 5 年 6 月 9 日現在

機関番号：32682
研究種目：基盤研究(C) (一般)
研究期間：2019～2022
課題番号：19K11901
研究課題名(和文)大規模グラフ並列処理プログラムのプログラム合成・変換手法に基づく開発手法の確立

研究課題名(英文) Developing a methodology for large-scale graph parallel processing based on program synthesis and transformation

研究代表者
岩崎 英哉 (Iwasaki, Hideya)

明治大学・理工学部・専任教授

研究者番号：90203372
交付決定額(研究期間全体)：(直接経費) 3,300,000円

研究成果の概要(和文)：本研究は、大規模グラフ並列分散処理を行うプログラムを効率的に開発し、かつ、プログラム毎に適したモデル上での実行が可能なプログラムを開発する手法を確立することを目的として研究を行った。その結果、次のような成果を得ることができた。(1)頂点主体計算と部分グラフ主体計算との適応的な共存機構として、計算の途中で頂点主体計算から部分グラフ主体計算に移行するシステムを実現した。(2)頂点主体計算よりも抽象度が高く、頂点の部分集合を扱うことのできる領域特化言語を設計し、そのコンパイラを構築した。(3)頂点主体と部分グラフ主体、同期的実行と非同期的実行が可能な並列分散実行基盤の基本部分を実現した。

研究成果の学術的意義や社会的意義

本研究は、大規模グラフ並列分散処理を、主として頂点主体と部分グラフ主体という代表的なふたつのモデル、さらに実行方式として同期的実行と非同期的実行に注目している。巨大なグラフデータに対する処理への需要が高まって久しいが、どのモデルや処理方式が適しているかは、問題そのものの性質、グラフの形状等に依存しており、実際に処理を行ってみたいとわからないという側面がある。本研究による頂点主体計算と部分グラフ主体計算の適応的な共存機構、公平な実行基盤提供するシステム等は、大規模グラフ並列処理プログラムの効率的な開発に寄与するものである。

研究成果の概要(英文)：The objective of this research was to develop a methodology both for efficiently developing programs for large-scale graph parallel/distributed processing and for executing programs on their suitable processing models. The following results were obtained. (1) As an adaptive co-existence mechanism between vertex-centric computation and subgraph-centric computation, we designed and developed a system that enables a program to be run initially in the vertex-centric manner and then in the subgraph-centric manner in the course of its computation. (2) We designed a domain-specific language (DSL) that can handle subsets of vertices at a higher level of abstraction than vertex-centric computation, and built a compiler for the DSL. (3) We implemented the basic part of a parallel and distributed processing system that enables both vertex- and subgraph-centric computation, and synchronous/asynchronous execution.

研究分野：プログラミング言語，並列処理

キーワード：大規模グラフ並列処理 頂点主体計算 部分グラフ主体計算 プログラム運算

1. 研究開始当初の背景

今日、大規模グラフデータが身近な存在となっており、そのようなグラフデータから新しい知識を獲得しようという需要が高まっている。大規模グラフデータはその大きさ故に、並列分散処理を行うという自然な要求が生まれる。我々は、大規模グラフデータに対するスケーラブルな並列分散処理のためのモデルとして、Pregel に注目した。その最大の特徴は、バルク同期並列に基づく頂点主体の計算モデルを採用し、並列分散プログラミングの難しさに対処している点である。このモデルでは、グラフ内の頂点は計算ノードに分散され、全頂点がスーパーステップ(以下 SS と略記する)と呼ばれる一単位の処理を同期的に繰返し実行する。このような同期的頂点主体計算に基づくシステムとして代表的なものに、Giraph、Pregel+などが存在する。Pregel モデルは、単純で理解しやすいという長所がある一方で、実際にプログラムを記述しようとする、記述面と性能面の二つの側面からの問題点に直面する。

我々は、主に記述面の問題を解決するために、グラフ処理のための高階関数群を用いて、同期的な頂点主体計算を宣言的に記述することが可能な関数型領域特化言語 Fregel (Functional Pregel) を提案しそのコンパイラのプロトタイプを開発した。その一方で、この研究を柁に発展させるためには、頂点主体計算からの更なる発展の必要性、プログラム変換・合成による最適化の必要性を認識するに至った。

2. 研究の目的

Fregel は、宣言的なプログラム記述を採用することにより、頂点主体計算の記述のしにくさの問題を解決したが、同期的頂点主体計算モデルが万能というわけではない。実装したいアルゴリズム、処理対象のグラフの形状、さらには並列分散グラフ処理基盤に依存して、性能が得られるアプローチは異なると考えられる。このことを考慮すると、同期的頂点主体だけにとどまらない柔軟な手法が必要である。さらに、同期的頂点主体からの発展に対応して、プログラム変換・合成に基づく新たな最適化手法も求められる。

本研究は、これらの点を追究し、大規模グラフ並列分散処理を効率的に行うための手法を提案し、それを実現することを目的とする。そこで、具体的には次のような目標を掲げて研究を進めることとした。

(1) 同期的な頂点主体計算からの発展：

上で述べたように、実際に性能が得られるモデルは様々な要因に依存して自明ではない。そこで同期的頂点主体計算を発展させ、どのような問題に対してどのようなモデルが有効かを考察する。頂点主体計算以外のモデルとして有力なのは、部分グラフ主体計算である。部分グラフ主体とは、計算ノードに分散させる単位を、頂点ではなくひとかたまりの部分グラフとするというモデルである。さらに実行の方式として、同期的実行/非同期的実行のバリエーションも検討する。

(2) プログラム演算の利用：

同じアルゴリズムであっても、有効なモデルは様々な要因によって変化し得る。しかし、モデルごとにプログラムを新たに一から記述し直すのは、プログラム開発効率やコストの面、バグ混入の可能性の面からも現実的ではない。そこで、プログラム演算の立場からこの問題に取り組む。ここでプログラム演算とは、プログラムの意味を変えないことが数学的に保証されている変換規則を用いてプログラムの形を変換し、何らかの意味でプログラムを「改善」する手法である。これは、Fregel のような宣言的に記述されたプログラムの扱いに適している。

3. 研究の方法

本研究を進めるにあたり、現状最もポピュラーな同期的頂点主体計算の特性を把握する必要がある。そこで、同期的頂点主体計算と同期的部分グラフ主体計算の両方に対応する既存の大規模グラフ並列分散処理基盤である Blogel (Pregel+の後継システム)を用いて、評価を行った。その結果をふまえた上で、頂点主体モデルからの発展という形で他のモデルを捉え、各種モデルの共存などの新たな機構を模索した。

また、既存研究である Fregel システムを本研究の成果と有機的に結びつけることを可能とするため、一部にプロトタイプ的な実装を含んでいた Fregel コンパイラの完成度を向上させる作業も並行して行った。

4. 研究成果

(1) 実用的な Fregel コンパイラの実装と評価

Fregel コンパイラは、メッセージ送受信の削減に注目した最適化を、(a)重複する値の送信の削減、(b)頂点における計算の単位元値の送信の削減、(c)頂点の非活性化に基づく終了判定のための値の集約の削減、等に注目して行っているが、試験的な実装にとどまっていた。これらの最適化を実際に行うための手法としては、最適化が可能な条件をコンパイラ中に知識としてあらかじめ保持させておく手法、および、最適化の可能性を SMT ソルバによって判定する方法がある。これらを Fregel コンパイラの中でプログラム演算に基づいて実用的なレベルで組み込み、コンパイラの起動オプションによりどちらの手法による最適化を用いるかを選択できるようにした。さらに、Fregel コンパイラの出力先システムを Giraph および Pregel+の両方に設定した上で、複数のベンチマークプログラムによる評価実験を行い、実装した Fregel コンパイラの性能を客観的に示した。これらの成果を含めた Fregel システム全般について議論した論文は、プログラミング言語分野での一流のジャーナルである Journal of Functional Programming に採択、掲載された。

(2) 頂点主体計算と部分グラフ主体計算の比較実験

同期的な計算において、頂点主体計算と部分グラフ主体計算のいずれが有利になるかは、状況に依存する。このことを確認するため、両計算モデルに対応する既存の大規模グラフ並列分散処理基盤である Blogel を用いて評価を行った。Blogel の上で、性質の異なる大規模グラフ（道路網グラフ、ソーシャルネットワークグラフ）に対して、性質の異なるアプリケーション（単一始点最短経路問題、ページランク）を実行する実験を行った。その結果、次のような傾向が観察された。

- ・道路網のように、疎で直径の大きなグラフは、部分グラフ主体計算の方が有利である。
- ・ソーシャルネットワークのように、密で small-world 的なグラフは、部分グラフ主体の効果は少なく、頂点主体よりもかえって遅くなることもある。

総じていえば、頂点主体、部分グラフ主体のどの処理形式が有利かは、グラフの形状や性質、さらに、アプリケーションの性質に大きく依存することが、改めて確かめられた。ここで得られた結果、知見は、以降の研究の基礎となっている。

(3) 頂点部分集合を直接扱うことが可能な機構

頂点主体計算が大規模グラフ処理に有効であることは誰もが認めるところであるが、頂点集合を陽に扱い処理単位とするアルゴリズムなどは、このモデルでは簡単には記述できない。それは、頂点主体計算は各頂点を「局所的な」処理対象と捉えるのに対し、頂点集合を処理単位とする計算はこの集合を「大域的な」処理対象として捉えるため、両者間に視点のギャップが存在するためである。このギャップを埋めるため、頂点集合を直接扱いプログラムを「大域的」な視点から記述することを可能とする手続き型の領域特化言語 G2PL を設計し、そのコンパイラを実装した。この言語は、頂点集合変数、集合操作演算、内包表記を用いた集約などの構文を備えている。G2PL コンパイラは、プログラムを頂点主体計算のプログラムへと変換する。

G2PL において頂点集合を表現する際の数学的な基礎は、集合の指示関数(indicator function)である。指示関数は、引数となる頂点が対象とする頂点集合に含まれるか否かを返す述語である。各頂点に、プログラム中で利用する頂点集合の数だけ指示関数値を保持させることにより、頂点集合に基づく計算を頂点主体計算に変換することが可能となる。

コンパイラのプロトタイプ実装では、変換先の言語を Fregel とした。Fregel は、グラフ処理のための高階関数群を提供しているため、G2PL プログラムを直接 Pregel プログラムに変換するよりも、Fregel プログラムに変換する方がはるかに小さい労力で実装できるためである。このことは、Fregel におけるグラフ処理の表現力の高さを示している。最密部分グラフ抽出(DS)プログラム、および、最適準クリーク抽出(OQC)プログラムを用いた評価実験では、Fregel プログラムをさらに Giraph に翻訳した場合、G2PL プログラムは計算ノード数に関して良いスケールビリティを示すものの、手書きのコードと比較して 3~10 倍程度遅いという問題点も残った。これまでの研究成果は、著名な国際ジャーナルである The Journal of Supercomputing に論文が採択、掲載された。

続いて、上の問題点に対処すると同時に G2PL の記述力を増すため、以下のような拡張を行った。(a)辺集合や辺の重みを利用するアルゴリズムを記述可能とした。このようなアルゴリズムの例としては、最適マッチング問題(MM)がある。(b)変換先の言語を Blogel としてコンパイラを再実装し、性能向上のための最適化機構を実現した。この拡張においては、頂点集合と同様にし

て、辺集合変数と演算、内包表記による辺集合の定義と集約などを提供する。再実装におけるポイントは、頂点に関する集約演算のコンパイルと、頂点の状態管理である。集約演算は、グラフ全体が対象の場合には Blogel の集約子 (Aggregator) を利用し、隣接頂点が対象の場合にはメッセージ送受信を利用する。さらに辺集合の集約演算は、頂点の集約演算に置き替える。頂点の状態については、集約演算ごとに SS を分割しなければならないことを考慮し、現在の状態を適切に管理する。さらに、不要なメッセージ送受信、不要な状態分割の削減、依存関係のない文の入れかえによる SS 数の削減、通信データサイズの削減など最適化も行う。これにより DS と OQC の場合、Fregel を変換先としたプロトタイプと比較して実行時間の半分以上が削減でき、手書きコードに近い性能のコード生成が可能であることを確認した。また MM では、最適化により 3 倍程度の実行速度の向上が確認できた。

(4) 頂点主体計算と部分グラフ主体計算の適応的な共存機構の実現

上で述べたように、頂点主体計算、部分グラフ主体計算のどちらの方式が有効化は一概に定めることはできない。頂点主体計算は、隣接頂点の情報を利用するために、多くのメッセージ送受信を必要とするという問題点がある。部分グラフ主体計算は、頂点主体計算のこのような問題点を克服するねらいがあるが、必ずしも頂点主体計算よりも効率が良いとは限らず、さらに頂点主体計算よりもプログラムが書きにくいという欠点もある。そこで、頂点主体で計算を始めた後に、適当なタイミングで部分グラフ主体計算に引き継ぐという計算実行方式を提案した。プロトタイプ実装は、Blogel が提供する、頂点主体計算と部分グラフ主体計算を同一の SS 内で行うことができるハイブリッドモードを利用した。

評価実験は、単一始点最短路問題のプログラムにおいて、重みなしグラフと重みありグラフを用いて行った。ここで部分グラフ主体計算においては、各部分グラフでダイクストラ法を利用した。また実行切り替えのタイミングは、SS を一定回数実行し終わった時点とした。実験の結果、途中で計算手法を切りかえる実行が、頂点主体計算だけ、あるいは、部分グラフ主体計算だけによる実行よりも高速となる場合があることが示された。計算の継続可能性の検討、最適な切り替え基準の構築は今後の課題である。

(5) 公平な実行基盤として大規模グラフ並列分散処理システム

頂点主体計算と部分グラフ主体計算、あるいは、同期的実行と非同期的実行など、モデル・実行方式は、それらの組合せとしていろいろと考えられる。しかし従来は、並列分散処理基盤ごとに採用するモデルや方式は決まっておらず、いろいろな組合せが可能な共通の基盤を提供するシステムはほとんどなかった。最近のシステムでは、わずかに Blogel が同期的実行という前提のもとで、頂点主体と部分グラフ主体の双方をサポートしている程度であった。このような状況では、同じプログラムに対して様々な組合せの方式でプログラムを実行し、結果を公平に比較することはできない。

そこで、Blogel をベースとして、非同期的実行をサポートするような拡張作業を行った。この拡張を実現するため、SS を、他ノードとの同期を必要とするもの (Global SS) と同期を必要としないもの (Local SS) に分けて考え、自ノード上の他頂点に向けたメッセージ送信はその場で処理を行い、Local SS を進めて処理を行うようにした。このような拡張を加えたところ、バリア同期による不要な待ち時間を減らすことが可能となるが、その一方で、1 回の SS で処理可能な受信メッセージの数が減少する結果、同期的実行の場合よりも実行すべき SS 数が増加してしまい、かえってオーバーヘッドとなる場合もあることが判明した。この拡張は現状においては基礎部分を実現した段階であり、更なる洗練化を行う必要がある。

(6) その他

大規模グラフ並列分散処理基盤の評価に使用するための大規模ランダムグラフを、自動的に並列生成するための手法を研究し提案した。具体的には、実世界のグラフの特徴を持つランダムグラフとして知られる Watts-Strogatz モデルに対し、Hadoop MapReduce を用いた並列生成手法を提案し、その評価を行った。この研究成果は、日本ソフトウェア科学会「コンピュータソフトウェア」に採択され掲載されている。この手法で生成した大規模グラフは、上記のいくつかの研究において実際に用いている。

5. 主な発表論文等

〔雑誌論文〕 計4件（うち査読付論文 4件/うち国際共著 1件/うちオープンアクセス 4件）

1. 著者名 Hideya Iwasaki, Kento Emoto, Akimasa Morihata, Kiminori Matsuzaki, Zhenjiang Hu	4. 巻 32
2. 論文標題 Fregel: a functional domain-specific language for vertex-centric large-scale graph processing	5. 発行年 2022年
3. 雑誌名 Journal of Functional Programming	6. 最初と最後の頁 e4:1-69
掲載論文のDOI (デジタルオブジェクト識別子) 10.1017/S0956796821000277	査読の有無 有
オープンアクセス オープンアクセスとしている(また、その予定である)	国際共著 該当する
1. 著者名 神野 薫, 江本 健斗	4. 巻 37
2. 論文標題 Watts-Strogatz モデルに基づく大規模ランダムグラフの分散並列生成	5. 発行年 2020年
3. 雑誌名 コンピュータソフトウェア	6. 最初と最後の頁 34-45
掲載論文のDOI (デジタルオブジェクト識別子) 10.11309/jssst.37.2_34	査読の有無 有
オープンアクセス オープンアクセスとしている(また、その予定である)	国際共著 -
1. 著者名 Kento Emoto, Fumihisa Sadahira	4. 巻 76
2. 論文標題 A DSL for graph parallel programming with vertex subsets	5. 発行年 2020年
3. 雑誌名 The Journal of Supercomputing	6. 最初と最後の頁 4998-5015
掲載論文のDOI (デジタルオブジェクト識別子) 10.1007/s11227-019-02821-w	査読の有無 有
オープンアクセス オープンアクセスとしている(また、その予定である)	国際共著 -
1. 著者名 加藤直斗, 岩崎英哉	4. 巻 Vol. 36, No. 2
2. 論文標題 Fregel コンパイラにおける不要な値送受信の削減	5. 発行年 2019年
3. 雑誌名 コンピュータソフトウェア	6. 最初と最後の頁 28-46
掲載論文のDOI (デジタルオブジェクト識別子) 10.11309/jssst.36.2_28	査読の有無 有
オープンアクセス オープンアクセスとしている(また、その予定である)	国際共著 -

〔学会発表〕 計3件（うち招待講演 0件 / うち国際学会 1件）

1. 発表者名 福島 央章, 江本 健斗
2. 発表標題 大域的グラフ計算記述言語の最適化および辺集合に関する拡張
3. 学会等名 火の国情報シンポジウム2023
4. 発表年 2023年

1. 発表者名 Ryuichi Takakura, Kento Emoto
2. 発表標題 Switching execution between vertex-centric computation and block-centric computation for distributed large-graph processing
3. 学会等名 9th International Symposium on Applied Engineering and Sciences (SAES2021) (国際学会)
4. 発表年 2021年

1. 発表者名 神野 薫, 江本 健斗
2. 発表標題 Watts-Strogatz モデルに基づく大規模ランダムグラフの分散並列生成
3. 学会等名 日本ソフトウェア科学会第36回大会
4. 発表年 2019年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
研究 分担者	江本 健斗	九州工業大学・大学院情報工学研究院・准教授	
	(Emoto Kento)		
	(00587470)	(17104)	

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8 . 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------