

令和 4 年 6 月 8 日現在

機関番号：12601

研究種目：若手研究

研究期間：2018～2021

課題番号：18K18032

研究課題名(和文) 高度なループ自動並列化技術の開発とベクトル化との統合

研究課題名(英文) Advanced loop parallelization and integrated vectorization

研究代表者

佐藤 重幸 (Sato, Shigeyuki)

東京大学・大学院情報理工学系研究科・助教

研究者番号：90779464

交付決定額(研究期間全体)：(直接経費) 3,200,000円

研究成果の概要(和文)：本研究プロジェクトでは、複雑なリダクションループの自動並列化を研究した。開発した技術により、逐次的で直観的な計算仕様を、さまざまな並列計算機に適した分割統治型の実装へとシステムチックに変換できるようになる。具体的には、コンパイラの基礎技術の一環として、1) 演算子抽出に基づく分岐除去手法、2) 動的な振る舞いに基づく並列化手法、及び 3) データシャッフルに基づくSIMD化手法を開発した。これらの手法を評価するために、4) 削減ループのベンチマーク集も開発した。さらに、複雑なリダクションの応用に関するケーススタディとして、5) 並列字句解析と 6) 並列正規表現マッチングの実装手法を開発した。

研究成果の学術的意義や社会的意義

今日において、既にさまざまな並列計算機が至る所にあると同時に、新しい並列計算機が次々と開発されている。その並列計算機のハードウェア性能を引き出すには、並列プログラムが必要になる。しかし、並列プログラムを上手く作ることは人間にはとても難しく、しばしば、間違った計算をしたり、性能が悪くなったりする。本研究で扱った自動並列化技術は、人間にとって直観的で間違えにくい計算仕様を、並列計算機に都合の良い計算形式に変換するものである。これにより、多様な並列計算機のハードウェア性能を、簡単かつ安全に引き出すことができるようになる。

研究成果の概要(英文)：In this project, we studied automatic parallelization of complicated reduction loops. Our developed techniques enable us to systematically transform serial intuitive specifications into divide-and-conquer implementations amenable to various parallel computers. Specifically, we developed 1) a branch elimination method by operator extraction, 2) a parallelization method based on dynamic behaviors, and 3) a vectorization method based on data shuffling as part of the fundamental compiler technology. To evaluate these techniques, we also developed 4) a benchmark suite of reduction loops. Furthermore, we developed implementation techniques for 5) parallel lexing and 6) parallel regular expression matching as case studies on applications of complicated reductions.

研究分野：プログラミング言語

キーワード：プログラム合成 自動並列化 コンパイラ

1. 研究開始当初の背景

コモディティ・組み込み・サーバなど普く用途において、並列計算機が使われている。この並列計算機のハードウェア性能を引き出すには、人間にとって分かり易くて間違えにくい逐次プログラムではなく、並列計算機に都合の良い並列プログラムでなければならない。しかし、一般に、並列プログラムを作ることは、同等な逐次プログラムを作ることに比べて大変難しい。人間にとって間違えやすい上に、苦勞して作成しても、逐次プログラムより性能が悪くなることも珍しくない。したがって、人間が並列プログラムを作ることは、生産性が低い。

この問題を解決するべく、コンパイラ研究の文脈では、ループ自動並列化が長く研究されてきた。ループ自動並列化とは、逐次的に記述されたループを、並列計算機に都合の良い並列ループへと変換する技術である。これにより、原理的には、明示的に並列プログラムを記述することなしに、人間が並列計算機の性能を享受できるようになる。しかし、実際には、自動並列化が成功するようなループの形式はとても限られている。特に、複雑なループ繰り越し依存を生み出すリダクションがあると、最先端の並列化コンパイラでも、ループの自動並列化が失敗する。

そのような複雑なリダクションループであっても、そのループが表現する計算を代数的に定式化できれば、並列化可能である。この代数的定式化に基づく自動並列化は、理論的な取り組みはあるものの、コンパイラ技術として十分研究されている訳ではない。SIMD 命令列への変換 (SIMD 化) のような、ハードウェアの特徴に合わせた低水準実装の構成法も、十分研究されているわけではない。

2. 研究の目的

代数的定式化に基づいて、複雑なリダクションループを系統的に並列化する実装技術を開発し、コンパイラ技術の形で自動化する。そして、SIMD 化を統合して、効率の良い低水準実装を系統的に構成できるようにする。これによって、並列計算の生産性を高める。

3. 研究の方法

4 年間の研究期間を概ね次のように進めた。

1 年目 入力言語を Python の埋め込み言語として設計し、プログラム検証器を統合した静的解析とコード変換器の基盤のプロトタイプを実装した。これにより、自動並列化技術を試作・検証する環境を整えた。並行して、手作業で SIMD 化したリダクションループの性能評価を実施した。

2 年目 複雑なリダクションループの並列化において障害となる条件分岐を、網羅的に簡約して除去する技術を開発し、1 年目で作成した基盤の上にプロトタイプ実装した。

3 年目 複雑なリダクションループの動的な振舞いから、その計算の代数的性質を発見する技法を開発した。また、多様なリダクションループを効率的な SIMD 命令列に系統的に変換する手法を開発した。これらの技術を評価する基盤として、リダクションループのベンチマーク集を開発した。並行して、複雑なリダクションの応用に関するケーススタディとして、SIMD 対応した並列字句解析の実装手法を開発した。

4 年目 開発したベンチマーク集を用いて、本研究プロジェクトで開発した並列化手法や SIMD 化手法を実験的に評価した。並行して、複雑なリダクションの応用に関するケーススタディとして、並列正規表現マッチの実装手法を開発した。そして、研究成果全体を発展・応用させる方向を模索した。

4. 研究成果

研究成果は大きく分けて、A) 条件分岐の除去手法、B) 動的な振舞いに基づく並列化手法、C) データシャッフルに基づく総称的な SIMD 化手法、D) リダクションループのベンチマーク集、E)

SIMD 対応した並列字句解析の実装手法, F) 並列正規表現マッチの実装手法, この6つである.

成果 A は, 条件分岐付き計算を max/argmax 演算に簡約することで分岐を除去する手法である. 先行研究では網羅的に除去されなかった max 演算を網羅的に発見でき, 既存研究では扱われていなかった argmax 演算の抽出も可能にする手法である. これにより, 複雑なリダクションループを max/argmax 演算による並列リダクションへと変換できるようになる.

成果 B は, リダクションループを, 具体的な入力値を使って実行し, その出力値を観測することで, リダクションループが持つ代数的性質を推定する手法である. 代数的性質が特定できれば, 既存のリダクションループの並列化に帰着できる. この手法は, その性質上, 推定の正しさを理論的に保証できないものの, 様々なリダクションループを簡便に並列化できる. 実際, リダクション並列化の既存研究のベンチマークのほぼ全てを並列化できることが確認できている.

成果 C は, データシャッフルを用いて, リダクションを総称的に効率良く SIMD 化する手法である. その骨子は, データシャッフルのコードを追加で挿入することで, 与えられたループの構造を極力変更せずに, ループ内のスカラ演算をベクトル演算に拡張することである. これにより, 多様な要素型の配列に対する複雑なリダクションループを, 効率的な SIMD 命令列へと変換できるようになる.

成果 D は, 多様なリダクションループを取り揃えたベンチマーク集である. 成果 B の評価では, 既存論文で使われた例を集めた際, 極めて類似した例や別名の重複があり, リダクション並列化の評価基盤が明快ではなかった. 更に, 個々の例の実用性も不明確であった. そこで, 既存の例を統一的に分類・整理してマイクロベンチマークとし, 更に実用コードに含まれる複雑なリダクションループを抽出して実用的なベンチマークとし, その両者を併せて, ベンチマーク集を作成した. これは, 新しい並列化手法を開発するときの評価基盤になるため学術的な意義が大きい.

成果 E は, テキストデータ処理に役立つ並列字句解析の実装手法である. 字句解析におけるトークンの最長一致の計算を, オートマトンにエンコードすることで, SIMD 命令を使って効率良く実装できるようにする. この手法を組み込んだ字句解析器生成系を開発し, 大規模 JSON データ処理に応用した.

成果 F は, 正規表現マッチングにおいて, 部分入力に対するマッチングを表現するオートマトンを使って, 効率的に並列マッチングを行う実装技法である. その骨子は, マッチングの最中に, 必要に応じて部分的にそのオートマトンを構成することで, オートマトン構成に係るオーバーヘッドを極小化することにある. この技法は, 成果 E を含め, 有限オートマトンに基づく文字列処理に広く応用できる技法である.

尚, 字句解析も正規表現マッチングも, オートマトンの状態遷移を計算する並列リダクションとして定式化されるので, 成果 E と F は, 状態遷移の代数的性質に基づく, リダクションループの並列化の応用技法と見做すことができる.

5. 主な発表論文等

〔雑誌論文〕 計2件（うち査読付論文 2件 / うち国際共著 0件 / うちオープンアクセス 1件）

1. 著者名 Li, L., Sato, S., Liu, Q., and Taura, K.	4. 巻 -
2. 論文標題 Plex: Scaling Parallel Lexing with Backtrack-Free Prescanning	5. 発行年 2021年
3. 雑誌名 Proceedings of the 35th IEEE International Parallel and Distributed Processing Symposium	6. 最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Moriyama, A. and Sato, S.	4. 巻 -
2. 論文標題 Reverse Engineering for Reduction Parallelization via Semiring Polynomials	5. 発行年 2021年
3. 雑誌名 Proceedings of the 42nd ACM SIGPLAN Conference on Programming Language Design and Implementation	6. 最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） 10.1145/3453483.3454079	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 0件）

1. 発表者名 佐藤重幸, 森畑明昌
2. 発表標題 一般化max演算子の自動抽出
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

1. 発表者名 佐藤重幸, 森畑明昌
2. 発表標題 一般化max演算子の抽出による分岐除去
3. 学会等名 情報処理学会第126回プログラミング研究発表会
4. 発表年 2019年

1. 発表者名 高品 剛大, 佐藤 重幸, 田浦 健次朗
2. 発表標題 Simultaneous Finite Automaton の部分構成による並列正規表現マッチ
3. 学会等名 第24回プログラミングおよびプログラミング言語ワークショップ (PPL 2022)
4. 発表年 2022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
---------------------------	-----------------------	----

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
---------	---------