

制御フローグラフの編集によるプログラム並列化支援ツール

高橋昌也[†]太田剛[‡]静岡大学大学院情報学研究科[†]静岡大学情報学部[‡]

1 研究の背景及び目的

プログラム並列化とは逐次プログラムを並列プログラムに変換することであり、自動によるものと手作業によるものに分けられる。自動並列化は作業の手間からプログラムを解放してくれるが、プログラムの解析の難しさや入力データの特徴などから望ましい性能改善が得られないこともあり、手作業による細やかな並列化に頼らざるを得ないのが現状である。しかし、手作業による並列化にも自動化と同様、困難な点が多々ある。手作業の並列化支援ツールに関しては多くの研究がなされており、プログラムはそれらを用いてプログラムの並列性などを調べることが出来る。しかし、並列化のためのプログラム改訂については基本的にプログラマ任せであり、並列プログラミング初心者にとってはその改訂によってバグが生じてしまうことも多々ある。

本稿では、制御フローグラフの編集による並列化支援ツールを提案する。プログラムを視覚的に表現、編集可能にすることで、直感的に作業することができ、プログラム改訂にツールが介入することでバグの発生を防ぐことが出来る。

2 研究のアプローチ

手作業による並列化には、並列化の方針検討、コーディングによる適用、テストといった作業が必要になるが、並列性解析の複雑さや、粒度・ロードバランスの検討の難しさ、性能予測の難しさによる試行錯誤の必要性、コーディング作業の煩わしさ、バグが発生しやすいなど様々な問題を抱えている。

本研究では、プログラムの解析による方針検討の支援、図を用いたプログラム構造の理解支援、グラフの直接編集による並列化適用作業という 3 点のアプローチによってプログラムの並列化作業を支援する。解析による方針検討の支援によって、プログラムはホットスポットがどこにあるか、その箇所のコードに並列性があるかなどを容易に把握することができる。また、図を用いたプログラムの構造の理解支援により、プログラミング初心者や対象のプログラムのアルゴリズムを知らない者でも理解しやすくなる。図の直接編集による並列化適用作業とは、まず先に挙げた理解支援のための図に対して、プログラムが並列実行させたい箇所を編集し、次いで、ツールが図を解析、並列プログラムを自動生成するという方法である。図の直接編集による並列化適用作業を用いることで、プログラム構造を視覚的に捉え

たまま作業を行うことができ、コーディング作業の簡略化も図ることが出来る。これらの支援によって、プログラマはどのように並列化するのかだけを意識すればよく、さらに様々な並列化の試行錯誤が容易にできるようになる。なお、共有メモリ型並列計算機を対象とし、プログラミング言語は C 言語、並列 API には OpenMP[1]を用いる。

2.1 図を用いた並列化作業支援

関数のコールグラフやデータ依存関係などを図で表現するツールは数多くある。本研究では制御フローとデータ依存関係を図で表現する。

プログラム改訂については、プログラマ任せのものがほとんどである。同一のコードを複数のスレッドで実行するデータ並列については for 文の直前にプラグマを挿入するだけで容易に適用することができるが、タスク並列については依存関係の無いコードを探し出して集めるという作業の手間は軽減できない。ただし、このコードの収集作業は、該当する処理のコードの移動に留まり、プログラムの意味を変更し得る処理を新たに書き加えるわけではない。本研究ではその点に着目し、ある単位の処理をノードとして図で表現し、図を編集することで制御の流れを変更するという手法を考えた。この手法を用いることでプログラムから制御の流れやデータ依存関係といった並列化に必要な情報だけを抽出し、プログラマはそれらに注目して視覚的に並列化を行うことができる。

2.2 マクロフローグラフ

ある程度の大きさの処理をノードとし、制御の流れ、データ依存関係を表現することができる図として、自動並列化コンパイラの粗粒度タスク間の並列性抽出手法である最早実行可能条件解析で用いられるマクロフローグラフ[2](以降 MFG)を用いる。MFG とは、マクロタスク(以降 MT)と呼ばれる単位をノードとして構成し、MT 間の制御の流れ、データ依存関係をエッジで繋いだ図である。MT は以下の 3 種類に分類される。

- 基本ブロック(BPA)
分岐・合流のない命令列
- 繰り返しブロック(RB)
while、do-while、for ループ
- サブルーチンブロック(SB)
サブルーチンコール

A tool for supporting program parallelization by editing its control flow graph

[†]MASAYA TAKAHASHI, Graduate School of Informatics, Shizuoka University

[‡]TSUYOSHI OHTA, Faculty of Informatics, Shizuoka University

3 プログラム並列化支援ツール

本研究の提案手法の検証に当たり、それを実現するツールを開発した。その実装には、COINS[3]を用いている。

3.1 COINS

COINS とは、新しいコンパイラ方式を容易に実験、評価できるような共通インフラストラクチャである。その基本は、2 種類の間中表現 HIR, LIR からなり、ソース言語から HIR への変換部、HIR での各種最適化部、HIR から LIR への変換部などの独立したモジュールから構成されている。本研究のツール実装に当たっては、C 言語から HIR への変換モジュール、HIR に対するデータ並列性解析モジュール・MFG 構成モジュール、HIR から C 言語への変換モジュールを利用している。

3.2 ツールの概要

ユーザは、まず並列化したいソースファイルをツールに入力する。ツールはソースコードを解析し、関数ごとに MFG を構成、ユーザに提示する。ユーザは MFG の制御の流れを編集し、並列化を適用する。最後にツールが MFG を走査し、その制御の流れを実現する並列プログラムを出力する。

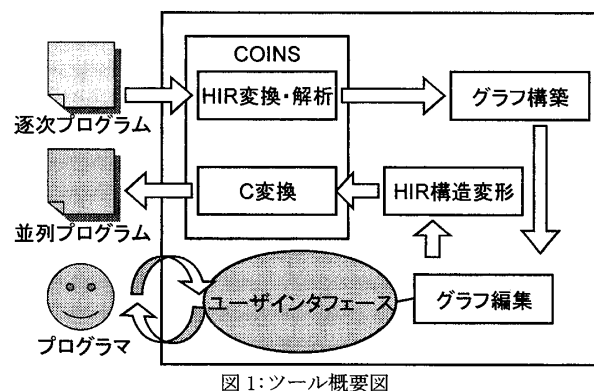


図 1: ツール概要図

ユーザによる MFG の編集に当たり、次のような機能を提供する。

- 実行順序変更機能

ユーザは、ノードをどのような順番で実行するのか指定することで、制御の流れを任意に変更することができる。但し、ノード n の制御の流れを変更する場合には、 n の移動元と移動先は互いに厳密な支配関係になければならない。

- ノード統合機能

ユーザは任意の数のノードをまとめて一つのノードとして構成することができる。但し、それらのノードは制御の流れで結ばれており、入り口・出口をそれぞれ一つだけ持たなければならない。この機能により、ユーザなりにプログラム構造を把握しやすいように図を構成することができ、また編集作業の操作性を向上させることができる。

- 並列化機能

ユーザは並列実行させたいノードを操作し、並列ノードとして構成することができる。並列化には次の操作を適用することができる。

- データ並列化

並列性を有する RB に対して適用する

- タスク並列化

任意のノードを並列タスクとして構成して適用する。並列タスクの追加や除外が可能。

- 依存関係検査機能

実行順序変更機能、並列化機能による制御の流れの変更の際に、各ノード間のデータ依存関係に違反がないか検査する。違反があった場合にはユーザに警告を促す。この機能により、バグの発生防止を働きかけることができる。

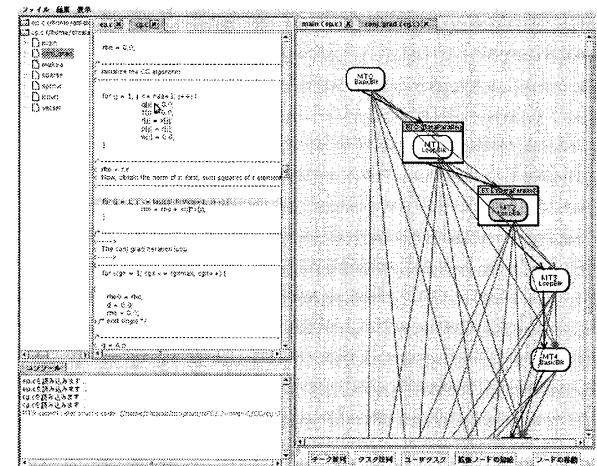


図 2: 並列化支援ツールのスクリーンショット

4 おわりに

本稿では、制御フローグラフの編集によりプログラム並列化を行うというアプローチによる並列化支援ツールを提案し、実装した。

今後、並列プログラミング初心者による本ツールの有効性の評価や、引き出すことのできる並列化効率の制限についての評価を行っていく。

今後の課題としては、ツールのさらなる改善が必要である。現状では、ノードの最小単位が MT であり、思うような並列化を行えない場合も多々あるため、ノードの分割機能の必要性が考えられる。また、同期制御などを用いた複雑な並列化を適用できるようにすることが望ましい。GUI のさらなる改良も必要となるだろう。

参考文献

- [1] OpenMP.org,
<http://www.openmp.org/wp/>, (2010/01/13)
- [2] 笠原博徳, “並列処理技術”, コロナ社, 1991
- [3] COINS コンパイラ・インフラストラクチャ,
<http://www.coins-project.org/>, (2010/01/13)