

5ZB-7

自動並列化コンパイラ MIRAI における 並列実行速度予測および並列度向上に関する一考察

善 隆裕[†] 峰尾 昌明[†] 上原 哲太郎[‡] 齋藤 彰一^{††} 國枝 義敏^{‡‡}

[†]和歌山大学大学院システム工学研究科 [‡]京都大学大学院工学研究科

^{††}和歌山大学システム工学部 ^{‡‡}立命館大学情報理工学部

概要

自動並列化コンパイラ MIRAI は、主にループを対象とした自動並列化コンパイラである。また、ソフトウェア分散共有メモリシステム Fagus 上で並列実行させることを目的としている。ループを並列実行可能にする最適化手法の中には排他的な手法や実行台数によって効果が変化する手法が存在する。そこで各種最適化手法を施した場合の予想実行速度を計算し、最も速度が速くなる最適化を施すことが理想と考える。よって、各種最適化を効果的に適用するために、並列化対象のループに対して最適化手法を施す前後の実行速度予測機能を MIRAI へ実装することを提案する。

1. はじめに

我々は自動並列化コンパイラ MIRAI^[1]の開発を行っている。MIRAI は、プログラム中の主にループ部分の並列化を自動的に行う自動並列化コンパイラである。また、実行時環境として、分散共有メモリを有する PC クラスタ環境を想定している。本稿は MIRAI において最適化を行うための基盤となる、PC クラスタ環境における実行時間予測について述べる。

ループ実行時間予測を行うことによって、最適化部で施す各種ループ再構築手法が実際の PC クラスタ環境で有効か否かを、事前に判断することが可能となる。これにより、従来は確実に高速化可能なパターンのループだけを再構築することで並列化を行っていたのに対し、これまで並列化をあきらめていた依存のあるループを並列化したり、同じくあきらめていたループ再構築手法を適用したりすることが可能となる。さらに、本実行時間予測結果は排他的なループ再構築手法の適用選択の指標となる。これは、排他的な複数の手法から適用手法を選択する際に、その有効度を事前に評価可能となるためである。

本稿では、MIRAI が想定する PC クラスタ環境における分散共有メモリシステムである Fagus について、その個々の操作の詳細な処理時間について計測した結果について述べる。さらに、これらの計測結果から自動並列化のための実行時間予測方式について概要を述べる。

2. 基本的な処理の時間測定

MIRAI の実行時システムである Fagus が並列処理のために提供する主要な関数を表 1 に示す。これらについて、実行時間に影響すると思われる転送メモリページ数(1 ページ 4KB)と PC クラスタの台数を変化させて測定を行った。評価は、プロセッサ Xeon 1.7GHz, メモリ 1GB の PC

表 1: Fagus が提供する並列処理のための主な関数群

fagus_acquire(WLOCK)	書き込み所有権を獲得する
fagus_acquire(RLOCK)	読み込み所有権を獲得する
fagus_release()	各所有権を解放する
fagus_sync()	他 PC クラスタと同期を取る
fagus_updatecache()	変数を最新の状態に更新する
fagus_barrier()	他 PC クラスタと同期を取り、変数を最新の状態に更新する

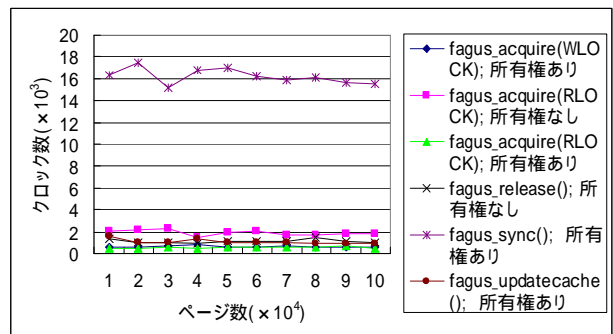


図 1: ページ転送が発生しない Fagus 関数群の実行クロック数

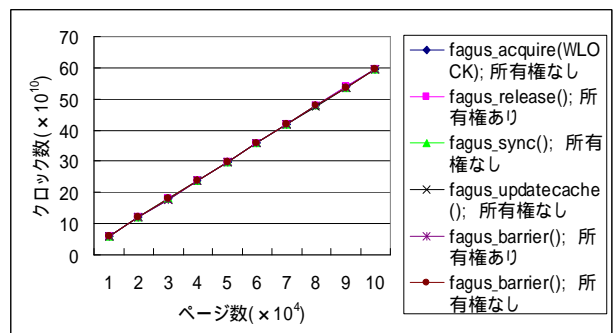


図 2: ページ転送が発生する Fagus 関数群の実行クロック数

A Consideration of Forecasting Approach on Parallel Execution Time and Improving Paralleling Features of an Automatic Paralleling Compiler MIRAI

[†]Takahiro Zen, Masaaki Mineo: Graduate School of Systems Engineering, Wakayama University

[‡]Tetsutarou Uehara: Graduate School of Engineering, Kyoto University

^{††}Shoichi Saito: Faculty of Systems Engineering, Wakayama University

^{‡‡}Yoshitoshi Kunieda: College of Information Science and Engineering, Ritsumeikan University

表 2：台数による変化がほぼ見られなかった関数群

Fagus が提供する関数	平均クロック数
fagus_acquire(WLOCK)所有権なし	55730457
fagus_acquire(WLOCK)所有権あり	692
fagus_acquire(RLOCK)所有権なし	35610901
fagus_acquire(RLOCK)所有権あり	624
fagus_release()所有権なし	39384462
fagus_release()所有権あり	50319148
fagus_sync()所有権あり	22060
fagus_updatecache()所有権あり	818
fagus_updatecache()所有権なし	50729551
fagus_barrier()所有権あり	51081776
fagus_barrier()所有権なし	53386252

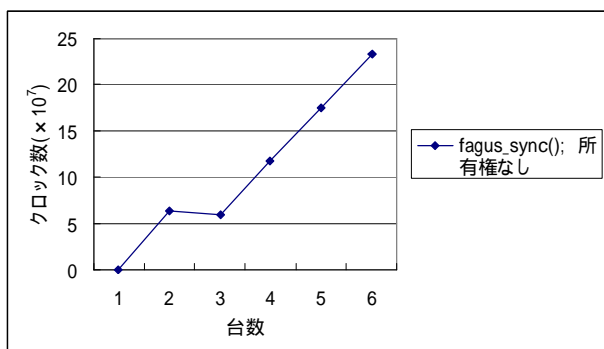


図 3: 台数によって時間が変化する関数

をギガビットイーサネットで接続した PC クラスタで行った。

図 1 と図 2 は 2 台のクラスタで並列実行させて転送メモリページ数を変化させて測定した結果である。図 3 と表 2 は転送メモリページ数を 100 ページで固定し、PC の台数を 1 台から 6 台で 1 台ずつ変化させ測定した結果である。

測定の結果からまず図 1 と表 2 を見てみると、転送メモリページ数と台数にまったく依存しない処理があることがわかる。これらのうち、所有権ありの状態における fagus_acquire() による書き込み所有権要求をした場合、所有権のチェックを行うだけで命令が終了し、実際のページ転送を伴わない。また、fagus_release() は所有権を解放するだけの処理のため、ページ転送は行われない。このような処理が台数や転送メモリページ数に依存しない命令であるとわかる。これらと逆に、図 2 で転送メモリページ数に比例して実行時間が増加する処理が見られる。例えば書き込みや読み込み所有権が無い場合の fagus_acquire() による書き込みや読み込み要求である。これらは Fagus が自ノードに所有権が無いことを感知して、ページを最新の状態に更新するために実際に転送しているために、送受信に掛かる時間がほぼそのまま実行時間となって現れていると考えられる。

次に台数による変化を見る。表 2 は台数による変化が見られなかった処理の一覧である。逆に、台数によって時間が増加する命令は、図 3 に示した所有権の無い

fagus_sync() だけである。この命令は他の PC クラスタと同期を取るための待ち時間が発生するためであると考えられる。同じことが fagus_barrier() 命令にも言えるのだが、こちらはデータ転送が起こりそちらの方に時間が掛かっているのが台数による差が隠れているのではないかと考えられる。これらのことから MIRAI に実装する上で実行時間予測に必要なものは実際に転送されるメモリページ数、転送が必要かどうか、実行時台数となると考える。

3. 実行時間予測の方式

実際に MIRAI に必要とされる実行時間予測部の大まかな処理の流れを説明する。MIRAI で生成された内部表現を用いループの流れに沿いながら実行時間を積算して予測実行時間を決定する。このとき PC 台数、転送メモリページ数を考慮に入れる。これはループの回数が実際に実行する際の台数で変化するためである。

また転送メモリページ数は Fagus の関数が使用される場合に重要となってくる。これは先の測定結果にもあるように転送メモリページ数に比例して時間が掛かる関数が存在するからである。しかし、この実行時間予測の時点では現在の MIRAI の仕様では転送メモリページ数を知ることができないために正確に算出することが困難となっている。ただ、配列のサイズは、宣言からわかるが、並列実行を前提としているので配列のサイズの半分を上限として採用することを考えている。

4. おわりに

現在の MIRAI にこの実行時間予測部を追加することによってよりきめ細かい最適化を行なうことができるようになる。このことによって並列実行可能となるソースコードやループが増加する。ひいてはプログラム全体の高速化が行えると思われる。しかし、実行時間予測部だけで完全に予測を行うことができない。そこで、必要だと思われるが完全ではない情報については適当と思われる値を用いて一度実行時間予測を行い、その結果を最適化部で用いる。最適化を行うことによって新たに判明した事実があれば、それを再度実行時間予測部にフィードバックさせ使用して実行時間予測を行う。このように最適化部と実行時間予測部を繰り返すことによってより正確に近い予測を行なうことができると考える。

参考文献

- [1]. 信原裕文, 峰尾昌明, 上原哲太郎, 齋藤彰一, 國枝義敏: “PC クラスタを対象とするループレベル並列化機能を有する MIRAI コンパイラにおけるループ再構築部の実装”, 情報処理学会研究報告, 2004-EVA-8, pp. 7-12 (2004).
- [2]. 横手聡, 林章仁, 齋藤彰一, 上原哲太郎, 國枝義敏: 高速通信ライブラリ Wind を用いたソフトウェア分散共有メモリシステム Fagus の性能評価, 情報処理学会研究会報告 2001-OS-88, Vol. 2001, No. 78, pp. 35-42 (2001).