

## 並列プログラムの処理粒度調整による高速化

1J-1

松本 剛 大澤 範高 弓場 敏嗣

電気通信大学大学院情報システム学研究科

## 1 はじめに

並列プログラムにおいて実行時間を速くするために重要なことの一つは、そのプログラムに内在する並列性をどのように抽出するかである。一般に並列処理を行なうと並列処理オーバーヘッドと呼ばれるものが存在する。並列オーバーヘッドは処理粒度を細かくすると大きくなり実行時間に影響を与える。一方、粒度を粗くすると並列性が少なくなってしまう。したがってその間に並列性があり、実行時間が速くなるような粒度がある。本研究では、処理粒度を変えることが可能で、静的に解析可能なプログラムについて処理粒度調整を行なった。また、静的に解析可能な並列プログラムが粒度調整によって高速化が可能か不可能かを並列計算機上で実際に粒度調整をしながら調べるのは効率が悪い。そこで並列計算機のモデルを考えてその上で、粒度調整を行なった時の実行時間を予測する式を作り、実行時間が最も速くなるような最適な粒度を求めることを考える。並列計算機のモデルとしてPRAMモデル、LogPモデル[1]と本研究で提案したオーバーヘッドの種類を拡張したモデルの3種類を使い、それぞれのモデルから推測される最適な粒度と実機との違いについて考察する。

## 2 並列計算機のモデル

PRAMモデルやLogPモデルは、アルゴリズムの設計と解析を行なう並列計算機モデルである。PRAMモデルは、アルゴリズムの大まかな評価には役に立つ。しかし、プロセッサ間の通信時間が0であるといった現実の並列計算機と当てはまらない仮定がある。LogPモデルは、並列計算機の重要なボトルネックを次のパラメータで表す：通信遅延( $L$ )、プロセッサがメッセージの送信または受信を行なっている時間( $o$ )、通信のバンド幅( $g$ )、プロセッサ数( $P$ )。これらを使うことで、プロセッサ間通信に生じるオーバーヘッドをモデル化している。並

列計算機における並列処理オーバーヘッドには、プロセッサ間通信以外に、スレッド生成時間や同期の実行時間がある。並列プログラムの処理粒度調整を行なうと、それに伴って生成されるスレッドの数が変わる。また、スレッド間で同期をとる処理があると同期の実行時間も変わってくる。これらのオーバーヘッドも考慮すべきかという仮定の下で、プロセッサ間の通信時間をLogPモデルで表現し、さらにスレッド生成時間や同期命令時間のオーバーヘッドを考慮した並列計算機モデルをLogP拡張モデルとして提案する。

## 3 並列計算機モデルの評価

PRAMモデル、LogPモデル、本研究のモデルの3種類のモデルの精度を評価する。評価のために、それぞれの並列計算機モデルでの並列プログラムの実行時間の予測する式から求めた値と、実際に実機を使って実行した実行時間の値との違いについて調べる。並列プログラムは、粒度調整が可能で、かつ、静的に予測可能であることが必要であることから、1から $N$ までの和を求めるプログラムを使った。また、実機として、電子技術総合研究所で開発された高並列計算機EM-4[2]を用いた。同機では、並列処理の粒度を細粒度から粗粒度まで自由に変えたプログラムを実行できる。

1から $N$ までの和を求めるループを分割して、それぞれを各プロセッサに均等に割り当てる。1から $N$ までの和を逐次的に処理する時間は、静的に命令数から計算して、 $7(N+1)$ クロックとなった。なお、1命令の実行時間を基本的には1クロックとし、メモリにアクセスを行なう命令には、アクセス1回につき1クロック加算した。この処理をプロセッサ $P$ 台で並列処理を行なう時の理想的な時間は $7(N/P+1)$ クロックである。PRAMモデルでの1台で計算する逐次処理から $P$ 台で計算する並列処理へ粒度を調整したときの実行時間を予測する式でもある。次に並列オーバーヘッドの部分の式を求める。その式は、同じ並列計算機でもどの様にスレッドを生成していくかというスケジューリング戦略によって異なる。本稿では、2分木上にスレッドを生成するスケジューリ

<sup>†</sup>Optimal granularity determination of parallel programs, Tsuyoshi MATSUMOTO, Noritaka OSAWA and Toshitsugu YUBA. The University of Electro-Communications. {matumoto, osawa, yuba}@yuba.is.uec.ac.jp

ング戦略 [3] とした。LogP モデルでは、P 台で実行するように粒度調整を行なったときの実行時間の予測式は次のようになった。

$$T_l = 29 \times \log_2(P+1) + 7 \times N/P - 22$$

パラメータの値をそれぞれ  $L$  が 11 クロック、 $o$  が 1 クロック、 $g$  が 2 クロックとした。また、LogP 拡張モデルでは、次のような式になった。

$$T_{le} = 71 \times \log_2(P+1) + 7 \times N/P - 64$$

スレッド生成時間を 21 クロック、同期命令時間を 0 とした。同期命令時間が 0 であるのは、他のパラメータと重なっていることによる。

実機での  $N=100$  のときのプロセッサ数を変えたときの実行時間と、モデルから推測される実行時間を図 1 に示す。

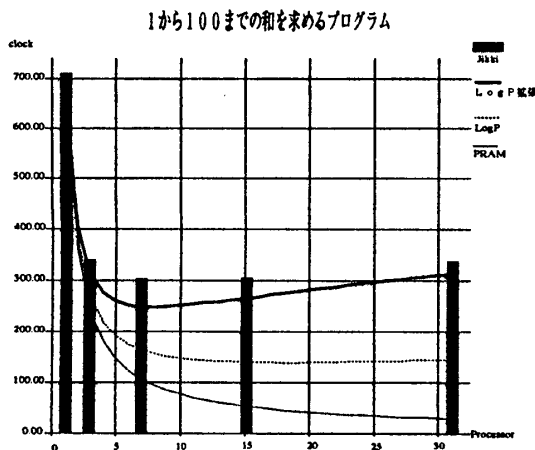


図 1: それぞれのモデルと EM-4 での結果 ( $N=100$ )

図 1 をみてわかるように、本研究で提案した LogP 拡張モデルが EM-4 の動作に最も近い。LogP 拡張モデルと実機との違いとして、 $L$  の値を一定にしていることや、パケットの衝突を考えていないことがあげられる。また、プロセッサ数を増やした時、つまり、粒度を小さくした時に EM-4 との差が他の 2 つのモデルに比べて小さい。したがって、実機を使わないで最も速くなる粒度を求めるための現実的な並列計算機のモデルとして有効である。

#### 4 粒度調整による高速化

図 1 の実機での結果からわかるように粒度調整によって実行時間の高速化ができています。 $N=100$  のときは、プ

ロセッサを  $P$  台使う時の処理粒度から、プロセッサを 7 台使うときの処理粒度に調整すると高速化が可能であるとわかる。また、並列計算機モデルから求めた式を使って最適な粒度を求めることができる。LogP 拡張モデルでの最適粒度は、 $\frac{\partial T_{le}}{\partial P} = 0$  として  $P$  の極小値を求めればよい。 $N=100$  のとき  $P=7.71$  となり  $p$  は整数であるから実機での結果と一致する。

#### 5 おわりに

本研究での成果は次のとおりである。

1. 並列プログラムの粒度調整によって高速化が可能であることがわかった。
2. 静的な解析が可能な並列プログラムについて、最適粒度を求める式を並列計算機のモデル上で作った。
3. モデルと実機で粒度調整を行なったときの違いを調べた。その結果、並列オーバーヘッドをプロセッサ間の通信時間だけでなく、スレッドの生成時間や同期命令実行時間を考慮する LogP 拡張モデルは、実機での最適な粒度を求めることができた。

今後の課題としては、スケジューリング戦略を本研究では固定していたが、可変にして粒度調整を行なうときの高速化について調べることや、実行時間が予測できないプログラムに対する最適な粒度を決定する方法の研究などがあげられる。

#### 謝辞

本研究を遂行するにあたりデータ駆動型並列計算機 EM-4 の実行環境を御提供頂いた、通産省電子技術総合研究所情報アーキテクチャ部計算機方式研究室の諸氏に感謝いたします。

#### 参考文献

- [1] Culler, D., Karp, R., Patterson, D., Sahay, A., Erik, K., Schauer,., Santos, E., Subramonian, R. and Von Eicken, T., "LogP: Towards a Realistic Model of Parallel computation", Proc. the 4th ACM SIGPLAN Symposium on Principles & Practice of Parallel Programming, pp.1-12 (1993)
- [2] Sakai, S., Yamaguchi, Y., Hiraki, K., Kodama, Y. and Yuba, T., "An Architecture of a Dataflow Single Chip Processor", Proc. ISCA 89, pp.46-53(1989)
- [3] Kodama, Y., Sakai, S. and Yamaguchi, Y., "Load Balancing by Function Distribution on the EM-4 Prototype", Proceedings Supercomputing '91, pp.522-531(1991)