

1P-5

高並列計算機 AP1000 上での 並列プログラミング

嶋田一郎

慶應義塾大学理工学研究科計算機科学専攻

1. はじめに

現在世の中の計算機への依存は言うまでもないが、計算機科学の分野の発展により、より複雑なタスクや、より大量のデータ処理が要求されるようになってきた。こうした要求に伴い、近年並列処理の研究が盛んになってきた。

並列処理における課題は山積みであり、まだまだ多くの問題点を含んでいる。今回、並列処理の特徴、問題点のうち、疎結合及び並列度と有用性という点に着目した。本研究は、疎結合並列計算機上での並列プログラミングに、固有な性質を見い出すことを目的としている。

2. 並列処理

2.1 並列計算機

並列処理の発展に伴い、多くの並列計算機が開発されている。それらは密結合同型並列計算機と疎結合同型並列計算機に大別できる。密結合同型並列計算機上での並列処理に関する研究に比べ、疎結合同型並列計算機上での並列処理に関する研究はまだまだ研究途上であると言える。

表 1: 密結合同型と疎結合同型並列計算機の特徴

	利点	欠点
密	高速通信可能	台数制限
疎	台数の無制限	通信のボトルネック

その大きな理由の1つに、通信のボトルネックという問題がある(表1)。メッセージ通信に関する問題は、

並列計算機に限ったことではなく、現在でも解決したとは言えないが、徐々に解決の方向には向かっている。

一方、疎結合同型並列計算機でのプログラミングにおいては、排他制御を行なう必要がない。このため、密結合同型並列計算機では考えられない数のプロセッシングエレメントを搭載することができるという利点を持っている。これに通信のボトルネックという大きな問題がなくなれば、非常に大きな並列効果が期待できる。このため、疎結合同型並列計算機を用いた並列処理の研究が盛んになりつつある。

こうしたハードウェアの発展に対し、疎結合同型並列計算機上で有効なプログラム開発を行なうための研究は現在それほど多く見られないのである。

2.2 並列度と有用性

並列処理の目標の1つに「実行速度の向上」があげられる。この目標が達成されなければ、多くの場合並列処理をする意味はない。理想的には図1における通常速度向上が見られればよい[1][2]。しかし並列プログラムを設計・実装するにあたって、理想通りの速度向上を達成することは非常に困難である。

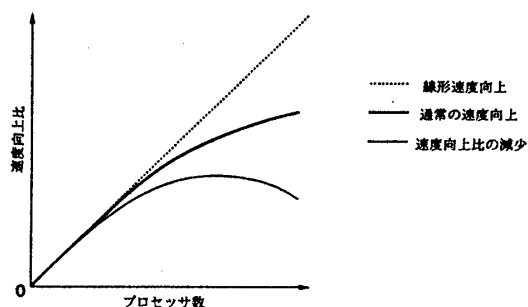


図 1: マルチプロセッサにおける速度向上比

A study of parallel programming on AP1000

Ichiro SHIMADA

Department of Computer Science, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa Pref., 223, Japan

そこで、並列プログラムにおいて並列処理の有用

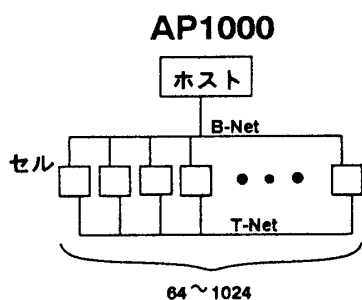


図 2: 高並列計算機 AP1000

性¹を達成するためには、並列度を適当に定めることが、非常に重要な要素となる。

今回は、特に並列度の決定が比較的難しいと思われる、データに並列性のない探索問題を取りあげ、並列に実装した。

3. 高並列計算機 AP1000

本研究では、富士通研究所が開発した高並列計算機 AP1000 を使用した (図 2)。AP1000 はメッセージ通信を基本とした分散メモリ型の並列計算機であり、その通信機構は、小レイテンシ・高スループットなどを実現している。

現在 AP1000 を用いた研究としては、UtiLisp を AP1000 へ移植した研究 [3] などの例がある。

4. 実験及び結果

疎結合型並列計算機 AP1000 上で探索問題 (パズル) を実装し、実験を行なった。そのうち、特徴的な性質がみられた 4 種類についてグラフに示した (図 3)。

図 3 においてそれぞれ 4 本の折れ線はパズルの種類を表している。type8 については、速度向上比がずっと 1 のまま向上していない。type8 のみ他の 3 つに比べ問題のサイズが小さいため、すぐに探索を終了してしまい、台数効果が全くでていないのである。他の 3 種類については、問題のサイズは同じであるが、それぞれ生成するプロセスの数が異なるために差が現れているのである。

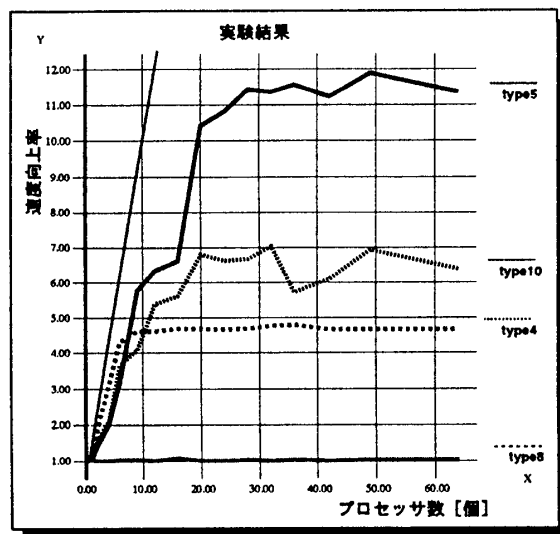


図 3: 並列処理による速度向上率

5. 結論

上の実験により以下の結論を得た。

- 並列プログラムにおいて並列度の選択は最も重要な事柄の 1 つである。
- 実行中、各プロセッサにかかる負荷はできるかぎり分散させる必要がある。
- 疎結合型並列計算機上でプログラムを設計する際、デッドロックの回避を常に念頭におく必要がある。

6. 今後の展望

今後の展望として、次の 2 点をあげる。

- 超並列にした場合の速度向上の達成
- 疎結合並列計算機上での一般的な並列アルゴリズムの提案

参考文献

- [1] M. ベン-アリ 著・渡辺 栄一 訳、並行プログラミングの原理、啓学出版、1986。
- [2] 富田真治・末吉敏則 共著、並列処理マシン、オーム社、1989。
- [3] 和田 英一 著、AP1000 用に拡張した UtiLisp/C とプログラミング、電子情報通信学会論文誌 (Vol.J78-D-I, No.2), 1995

¹ ここでいう有用性とは、いわゆる台数効果である。