

並列推論マシン PIM/i の  
アドレストレースによる  
メモリスステムの評価

7M-5

大原輝彦 武田浩一  
沖電気工業株式会社

1 はじめに

筆者らは、第5世代コンピュータプロジェクトの一環として並列推論マシン PIM/i の研究開発[1]を行ってきた。PIM/i は KL1 で記述されたプログラムを効率よく実行することを目的として開発されたマシンである。

PIM/i のクラスは8台のプロセッサと共有メモリを単一の共有バスで結合した構成となっている。共有バス結合の問題点は、共有メモリのアクセス競合による性能低下である。そこで、メモリスステムの構築に際してアクセス競合を軽減するため、並列キャッシュと局所メモリを実装するという方法を採用している。

本稿では、共有バスとプロセッサのメモリアクセストレース情報をもとに、並列キャッシュを含むメモリスステムの振舞いを調べ、その評価を行ったので結果を報告する。

2 PIM/i のメモリスステムの概要

2.1 リソース緒元

PIM/i のメモリスステムは表1に示すリソースから構成されている。それらの諸元を以下に示す。なお、PIM/i の1語は40ビット(タグ8ビット、データ32ビット)である。

表1 PIM/i メモリスステムの構成

リソース	緒元
データキャッシュメモリ	32K 語
コードキャッシュメモリ	32K 語
局所データメモリ	16K 語
局所コードメモリ	32K 語
共有メモリ	64M 語
共有バス(データ)	40 bit
共有バス(アドレス)	38 bit

2.2 キャッシュメモリ

共有バス結合のマルチプロセッサシステムにキャッシュメモリを実装することは一般的である。そこで PIM/i では、表1に示したようにデータとコードキャッシュを分離して実装し、データキャッシュにはライトバック方式の並列キャッシュ[2]を採用している。コヒーレント制御は書き込み更新により行っている。状態制御は次の4つのフラグを組み合わせた6状態により行っている。

- 1: S / E 共有/非共有
- 2: 0 / U キャッシュ間転送の供給元/非供給元
- 3: C / D 書き戻しの必要なし(clean)/あり(dirty)
- 4: I キャッシュエントリが有効かどうかを示す

2.3 共有バスのメモリ供給能力

1台のプロセッサが命令を1ステップ実行する時間に共有バスがどのくらいのメモリ語をキャッシュメモリに供給できるか(以下、メモリ供給能力という)をキャッシュ

動作毎について表2に示した。キャッシュはメモリ4語を1ブロックとして管理しそれについて前述の状態制御を行っている。

表2

キャッシュ動作	状態	動作後の状態	
		非共有	共有
リードミス	clean	0.25	0.57
	dirty	0.13	0.20
ライトヒット	非共有	1.00	-----
	共有	-----	0.25
ライトミス	clean	0.25	0.57
	dirty	0.13	0.20

この表から、「dirty 状態のキャッシュブロックにリードあるいはライトミスしたとき」が最もメモリ供給能力が低いことがわかる。この処理は言いかえると、書き変えたメモリを共有メモリに書き戻し、新たなメモリをキャッシュに転送する場合である。

2.4 KL1 のメモリアクセス特性

ここでは KL1 のメモリアクセス特性について簡単にふれる。KL1 は並列プログラミングの概念を自然に記述できるすぐれた言語であるが、これを実装する上ではいくつかの問題がある。KL1 は単一代入言語であり、変数に対する破壊的な代入を許さない。また prolog のようにバックトラック時にメモリ領域を解放するといった機構を持たない。したがって、メモリ消費が速いという問題があるといえる。これに関する実装の技法は[3]にその詳細が述べられている。

KL1 の実装では MRB 方式という低コストな即時方式の GC と一括方式の GC を併用している。一括 GC は実装されているメモリを消費し尽くすと起動される。

ここで重要な点はメモリの消費速度である。メモリの消費速度は2.3節の共有バスのメモリ供給能力と関係が深い。つまり、新たにメモリを消費する速度は共有バスのメモリ供給能力の制限を受ける。PIM/i ではプロセッサでメモリを新たに消費する命令の出現頻度が 0.0166(0.133/8)すなわちプロセッサ8台につき約60ステップに1ステップ以上あると共有バスのメモリ供給能力が不足となる。

3 測定

測定は、実機とシミュレータ[4]を併用して行った。

3.1 シミュレータによる測定

評価の基準として、シミュレータを用いて理想共有メモリを実現し、データを採取した。理想共有メモリとはアクセス競合が全く起きないと仮定したメモリである。表3にキャッシュ付きシミュレータと理想メモリの場合の比較を実行時間の比で示した。

表3 理想共有メモリとの性能比

ベンチマーク	1pe	2pe	4pe	8pe
pentomino	1.01	1.01	1.07	1.31
qk	1.02	1.05	1.04	1.13
uq	1.07	1.08	1.10	1.25

Evaluation of System on Parallel Inference Machine  
PIM/i Using Address Tracing  
Teruhiko OOHARA, Koichi TAKEDA  
Oki Electric Industry Co., Ltd.

ヒープの使用量とヒープ領域で発生した書き込み更新の割合を表4に示した。なお1ページは256語である。

表4

ベンチマーク	割合	ページ数
pentomino	72.7%	99
qk	42.3%	95
uq	37.1%	462

3.2 実機による測定

実機ではバストレース情報をもとにプロセッサ稼働率を測定した。それを表5に示す。表5には、さらにキャッシュの共有ブロックの割合も示す。ここで稼働率とはプロセッサがメモリ供給を待っている時間を差し引いた時間が全実行時間のどれくらいを占めているかを率で表わしたものである。

表5

ベンチマーク	稼働率	共有ブロックの割合
pentomino	70.5%	61.7%
qk	92.2%	32.8%
uq	81.0%	29.1%

3.3 アドレストレース

本評価に用いたベンチマークプログラム uq のメモリ割当て(ヒープ領域)の様子をシミュレータによるアドレストレース情報をもとに視覚化したものを図1に示す。uq は表5に示されるようにベンチマークプログラム中最も大きなヒープ領域を使用する。この図は横軸にシミュレーション時間、縦軸にヒープアドレスをとり、全てのメモリアクセス(リード、ライト等)についてプロットしたものである。この図でヒープが割当てられる様子がよくわかる。

4 評価と考察

PIM/i のメモリシステムに関して、アドレストレース情報をもとにその振舞いを調べた。

表3では理想共有メモリに対する PIM/i メモリシステムのオーバーヘッドが示されている。メモリアクセス情報とキャッシュの共有ブロックの割合から pentomino では書き込み更新が、uq・qk では共有バスのメモリ供給能力がオーバーヘッドの主要な原因であると考えられる。pentomino では MRB によるヒープの再利用の結果共有ブロックが増加したことにより書き込み更新が多くなったと考えられる。

uq のようにヒープを単調に使用するベンチマークでは、時間と使用量の関係を1次式で近似できる。それを図2に示す。さらに図2には、理想共有メモリを使用した場合と、共有バスのメモリ供給能力の限界値も示してある。この例では、まだ共有バスのメモリ供給能力に余裕がある。

以上のように PIM/i のメモリシステムでは(1)書き込み更新(2)共有バスのメモリ供給能力、の2種類のオーバーヘッドが存在することがわかった。オーバーヘッドの最悪値は表3よりそれぞれ1.31と1.25であった。これらのオーバーヘッドは同時に発生するとは考えにくい。なぜならヒープを多く使用すると共有ブロックがキャッシュから追い出されるからである。

共有バスのメモリ供給能力を向上させることは両方に対して有効な手段であり、前者に対しては、さらにキャッシュプロトコルの最適化による改善が有効であると考えられる。

謝辞

日頃、助言をいただく新世代コンピュータ技術開発機構第1研究室、および沖電気の PIM 担当諸氏に感謝する。

参考文献

- [1] 大原他：「並列推論マシン PIM/i の概要」、情報処理学会第40回全国大会予稿集
- [2] 大原他：「並列推論マシン PIM/i の並列キャッシュ機構の評価」、情報処理学会第43回全国大会予稿集
- [3] 今井：「VPIM 処理方式解説書」、ICOT Technical Memorandum:TM-1044
- [4] 吉田他：「並列推論マシン PIM/i の開発環境—シミュレーター—」、情報処理学会第41回全国大会予稿集

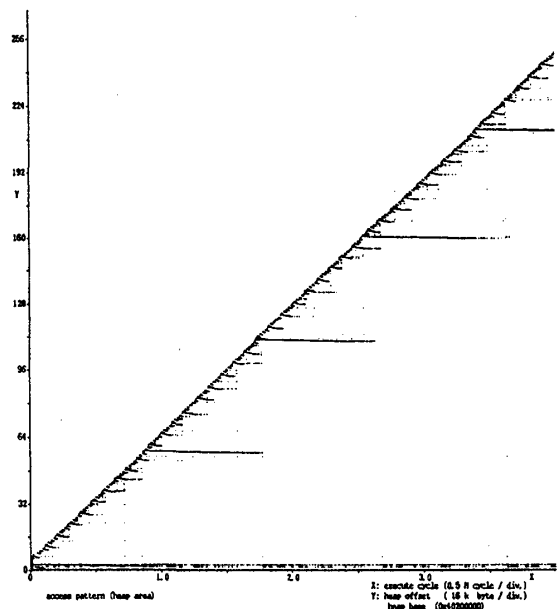


図1

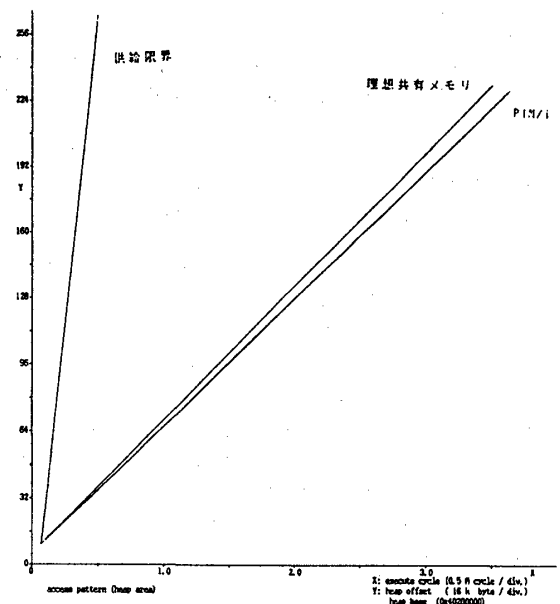


図2