

並列推論マシンPIM

——クラスタ内処理のメモリアクセス特性——

中川貴之、松本明、佐藤正俊、清水肇、後藤厚宏
(財)新世代コンピュータ技術開発機構)

1.はじめに

中期の並列推論マシンPIM(Parallel Inference Machine)のハードウェアは、図1に示すように共有メモリを介した密結合マルチプロセッサと、マルチPSIで導入している疎結合マルチプロセッサの二階層の構成をとる¹⁾。クラスタと呼ぶ密結合マルチプロセッサではPIMの対象言語であるKL1(Kernel Language/1)と、その処理系が有する参照局所性を活用することが課題となった。本発表では、クラスタ内処理におけるシミュレーション(CLSIM)結果の静的解析をもとに、メモリアクセス特性のモデル化を試みる。また、ライトバックタイプ並列キャッシュにおいてヒープ的使用を行う場合にバストラフィックを削減する一手法の効果を評価する。

2.共有データアクセスモデル

図1のような共有バス結合に基づく並列プロセッサにおいては、転送時間が一律に短いために使い勝手が良い反面、並列効果の上限がバストラフィックがネックとなって規定される短所がある。そこで、バストラフィックを削減する、さまざまな並列キャッシュプロトコルが提案されている²⁾。これらの並列キャッシュプロトコルの根本理念は“プロセッサ間に共有されないデータ”の状態を動的に管理する、バスマスターに体现される。すなわち真に必要なキャッシュ間/キャッシュ-メモリ間相互作用のみに共有バスを使うために制御を複雑化するが、どのような制御が有効であるかは、言語/処理系/応用プログラムに依存する。

プログラムを実行した場合のプロセッサからのアクセスパターンを静的に記述する以下のパラメタが考えられる。

(1)静的アクセス特性

read : read-frequency
write : write-frequency
acc : access-frequency(i.e.read + write)
word : used-words
wword : written-words
pea(i) : count-of-pe-which-accessed-to-a-word(i)

これに対し、キャッシュの動的なふるまいはヒット率(hit)、キャッシュの状態に占める変更率(modify)、キャッシュのミスヒット時における他キャッシュ上での存在確率(common)により表されることが考えられる。すなわち、バストラフィックの発生要因は以下のように定式化できる。

(2)バストラフィック発生要因

write-back = $(1 - \text{hit}) \times \text{acc} \times \text{modify}$
swap-in = $(1 - \text{hit}) \times \text{acc} \times (1 - \text{common})$
cache-xfer = $(1 - \text{hit}) \times \text{acc} \times \text{common}$
invalidation = $\text{hit} \times \text{write} \times \text{common}$

共有バスの使用サイクル数は、コマンドの転送サイクルを2サイクル、共有メモリのアクセスタイムを3サイクル、キャッシュのブロック長をBLKとして、以下のように定式化できる。

(3)バストラフィック

bus-cycles-with-write-back =
swap-in $\times$ modify $\times$ (BLK + max(BLK, 3) + 2) +
cache-xfer $\times$ modify $\times$ (BLK + BLK + 2)
bus-cycles-without-write-back =
swap-in $\times$ (1 - modify) $\times$ (BLK + 3 + 2) +
cache-xfer $\times$ (1 - modify) $\times$ (BLK + 2)
invalidation-bus-cycles = invalidation $\times$ 1

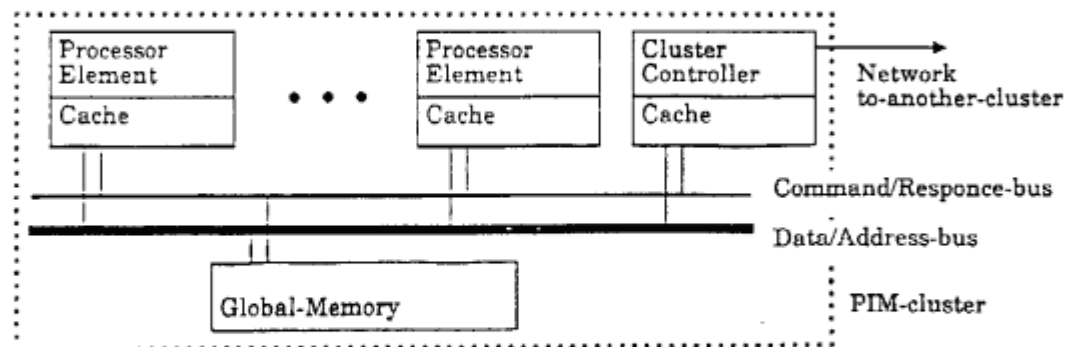


図1 共有バスモデル

Parallel Inference Machine : PIM- On Memory Access Characteristics of the TCMP Cluster -
Takayuki NAKAGAWA, Akira MATSUMOTO, Masatoshi SATO, Hajime SHIMIZU, Atsuhiko GOTO
Institute for New Generation Computer Technology (ICOT)

静的アクセス特性とキャッシュの動的なふるまいは、キャッシュサイズが無限大でインバリデーションの効果がヒット率に関して無視しうるとき、以下のように関連付けられる。

(4)パラメタモデル

$$\begin{aligned} \text{hit} &= (\text{acc} - \Sigma \text{pea} / \text{BLK}) / \text{acc} \\ \text{common} &= (\Sigma \text{pea} - \text{word}) / \Sigma \text{pea} \\ \text{modify} &= \text{wword} / \Sigma \text{pea} \end{aligned}$$

KL1の処理には、データエリアを単調に消費するヒープ、機械語命令であるコード、制御情報としてのゴール/メタコール/サスペンドレコードの計5種類のデータ領域を使用している。これらの領域のアクセス特性は互いに異なるため、領域間の相互作用を無視すると、総体としてのパラメタは領域毎にモデルを適用した期待値になる。

なお、キャッシュサイズが有限である等の効果はスワップイン回数の増加として位置付けられ、 $2\text{word} \times 1024\text{column} \times 2\text{set}$ キャッシュの動的シミュレーション結果との比較では1.67倍に増加する。

(5)期待値

$$\begin{aligned} \text{common-at-write} &= \Sigma (\text{write} \times \text{common}) / \Sigma \text{write} \\ \text{common-at-access} &= \Sigma (\text{acc} \times \text{common}) / \Sigma \text{acc} \end{aligned}$$

表1に示すメモリアクセスデータは我々が作成したクラスタの実験処理系においてBUP(Bottom Up Parser)のプログラムを8台のプロセッサからなるクラスタで実行した場合の各プロセッサからのアクセスパターンを静的に解析したものである。このプログラムでは36Kリダクションを570KステップのKL1-b命令(KL1のマシン語)で処理するに際し、1.3Mアクセスを要している。

PIMではプロセッサ間の負荷配分をゴールフォークとして実現するが、これによるデータの共有率 common-at-access は0.54であり、高いといえる。従ってキャッシュ間にわたる転送等の相互作用は本質的に多く、共有バスネックは深刻な問題となっている。また、 common-at-write は0.12であり、インバリデーションの削減効果は大きい。

3.共有バスネックの一軽減手法

逐次型マシンPSIではスタック伸長時にアドレス昇順に書き込みから使用する領域に対するライトバックキャッシュのフェッチ回数軽減のために、このような領域への書き込みにミスヒットしてもフェッチを伴わないwrite-stack命令を用意している。我々はこの拡張として、ミスヒット時においてもフェッチおよびインバリデーションを伴わないdirect-write命令を用意する。

すなわちdirect-write命令においては、キャッシュのブロック境界の先頭をアドレス指定した場合、フェッチおよびインバリデーションを伴わず、先頭以外をアドレス指定した場合、通常のwrite命令の処理を行ってブロックへのアクセス間に発生しうるスワップアウトに対応する。

direct-write命令によるバストラフィック削減効果は、direct-write命令の発行回数をd-writeとして、スワップイン回数をd-write/BLKだけ減じ、更にスワップインを伴わない以下のバストラフィックを算入することにより評価できる。このモデルでは約30%の減となり、動的シミュレーション結果と一致する。

$$\begin{aligned} \text{write-back-cycles} &= \\ & \text{d-write} / \text{BLK} \times \text{modify} \times (\text{BLK} + 1) \end{aligned}$$

4.結論

本発表では並列プロセッサ環境における共有データへのアクセスモデルを提示し、さらにストアインキャッシュを備えた共有バスモデルのマシンにおけるバスネック軽減の一手法としてヒープ的使用を行うデータに対するdirect-write命令の適用を提案し、その評価を試みた。今後は更に並列度をあげる検討を進める予定である。

<参考文献>

- (1)後藤、松本、佐藤、"並列推論マシンPIM", 情報処理学会第33回全国大会予稿集3B-5~7
- (2)P.Bitars, A.M.Despain, "Multiprocessor cache synchronization", the 13th annual international symposium on computer architecture

表1.Bottom-Up-Parserの静的アクセス特性

	heap	code	goal	metacall	suspend
word(wword)	62,675(同)	1,983(0)	3,515(同)	10(9)	682(同)
read	94,747	672,495	150,903	62,557	7,071
write	35,349	0	150,903	26,840	6,717
d-write	62,675	0	0	0	0
Σpea	73,003	12,753	3,515	80	3,070