

並列推論マシンPIM

—並列キャッシュとロック機構—

松本 明、中川 貴之、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

ICOTで開発中のPIMは、クラスタ概念を用いた階層構造で、クラスタ内の各要素プロセッサ(PE)は、並列キャッシュを介して共有メモリに接続している[1]。

PIMのKL1(核言語第一版)処理系とハードウェアの設計は、新たに開発したクラスタ内処理シミュレータと並列キャッシュ・シミュレータを用いて、処理系のメモリ参照特性を両者の設計にフィードバックしながら進めている。

PIM用並列キャッシュとロック機構の設計ポイントは、①KL1処理系に依存した最適化を施した構成にすること、②並列キャッシュのヒット率を高くするよりも、むしろ共有バスの使用率の低減に重点を置くこと、③共有メモリのロック操作は、並列キャッシュの状態を利用した最適化をすることである。本稿では、以上のポイントを念頭において設計を進めているPIMの並列キャッシュ機構と共有メモリのロック機構について報告する。

2. KL1処理のメモリ参照特性

クラスタ内処理シミュレータで簡単な構文解析プログラム(BUP)を実行した時のメモリ参照特性を表1に示す。

BUPプログラムのメモリ参照の内、約50%強は命令コードの読み出しである。しかし、命令コードを除く他のメモリ参照は、従来型言語に比べて書き込みの比率が高い。一般に書き込みをすると、他の並列キャッシュ上のコピーを無効化する等して並列キャッシュ間の一貫性確保を行うので、共有バスの使用頻度が増加する。この共有バスの使用頻度の増加をいかに抑えるかが、並列キャッシュ機構の設計における大きな課題である。

複数のPE間で共有されるヒープ領域へのメモリ参照頻度は、全体の十数%程度であるが、共有ヒープ領域中の未定義変数への具体化を排他制御するために、ロック操作が必須である。この共有変数へのロック操作に係わる共有バスの使用頻度の増加をできるだけ少なくするためには、並列キャッシュの状態を利用した高速なハードウェア・ロック機構を用意することが効果的である。

表1. KL1処理のメモリ参照特性

領域名	読み出し	書き込み	ロック
命令コード	53 %	0 %	不要
グローバル・レコード	12 %	12 %	不要
ヒープ	7.5 %	7.5 %	要
メタコール・レコード	5 %	2 %	要
サスペンション・レコード	0.5 %	0.5 %	不要
合計	78 %	22 %	

Parallel Inference Machine: PIM

- On Parallel Cache and Lock Mechanism -

Akira MATSUMOTO, Takayuki NAKAGAWA, Atsuhiko GOTO

Institute for New Generation Computer Technology (ICOT)

3. 並列キャッシュ機構

3.1. 並列キャッシュ機構設計のポイント

並列キャッシュ機構では、キャッシュのヒット率よりも、むしろ共有バス使用頻度の低減が大きな設計ポイントとなる。PIMでは、Bitar & Despainが提案[2]したライトバック方式モデルを基に、KL1の並列実行特性に合わせて最適化した6状態並列キャッシュ機構を設計中である。

並列キャッシュ・コントローラは、自PEの書き込み/読み出し操作と並列に、共有バスを通して他PEから来るバスコマンドにも応答するハードウェア機構を持つ。

3.2. 6状態並列キャッシュ

PIMでは、次の4個のフィールドを用いて6状態並列キャッシュを実現する。①Valid/Invalidは、キャッシュ・ブロックが有効/無効であることを示す。②Origin/Copyは、キャッシュ・ブロックが原本/コピーであることを示す。③Exclusive/Sharedは、他の並列キャッシュにコピーが存在しない/するかを示す。コピーが存在しない状態での書き込み時には、他の並列キャッシュの無効化を抑制できる。④Clean/Modifiedは、キャッシュ・ブロックが変更されていない/いるかを示す。変更されていない状態でのスワップアウトは抑制できる。次に6状態の説明を、また状態遷移を図1に示すが、このように、多くの状態を設けたのは、並列キャッシュ間での一貫性を保ちつつ、共有バスの使用頻度を減らすためである。

①EC: Valid, Origin, Exclusive, Clean

コピーが無い原本で、変更の無い状態。

②EM: Valid, Origin, Exclusive, Modified

コピーが無い原本で、変更の有る状態。

③SC: Valid, Origin, Shared, Clean

コピーが有る原本で、変更の無い状態。

④SM: Valid, Origin, Shared, Modified

コピーが有る原本で、変更の有る状態。

⑤C: Valid, Copy

上記SC, SMのコピーである状態。

⑥I: Invalid

無効(未使用を含む)な状態。

3.3. メモリ操作用のCPUコマンド

KL1処理系を考慮して用意した、メモリ操作用の主なCPUコマンドを以下に示す。

①R: Read (データの読み出しコマンド)

②W: Write (データの書き込みコマンド)

③DW: Direct Write

書き込みミスヒット時に、ブロックを共有メモリ、または、他のプロセッサから転送せずに直接自PEのキャッシュ・ブロックに書き込む。参照しないことが予め分かっているデータのスワップインを抑制して共有バスの使用頻度を下げる最適化を行ったコマンド。

K L 1 処理系では、各 P E が共有メモリ上に新しい環境領域を確保する時に利用している。

④ R B : Read Buffer

共有メモリを介して P E 間通信を行う時に、受信側の P E がブロック内データを読み出し終わった時点で送信側 P E のブロックを無効化する。さらに、受信側 P E が最後のブロック内データを読み出し終わった時点で、受信側のブロックも無効化する。できる限り早い時点で無効化をすることにより、後で参照されることのないブロックがスワップアウトされることを抑止して共有バスの使用頻度を下げる最適化を行ったコマンド。

K L 1 処理系では、P E 間のゴール・レコードの通信等に利用できる。

3.4. メモリ操作のバスコマンドと応答

メモリ操作の主なバスコマンドと応答を次に示す。

① F : Fetch (ブロックのフェッチ要求バスコマンド)

② F I : Fetch and Invalidate

ブロックのフェッチ要求と同時に、他の全ての P E のブロックを無効化するバスコマンド。

③ I : Invalidate (ブロックの無効化要求バスコマンド)

④ H : Hit (ヒットを示す応答)

4. ロック機構

4.1. ロック機構設計のポイント

集中管理型のロック・メモリを用いる方式は、ロック操作の度に共有バスへのアクセスが起こるので良くない。P I M では、ロック操作を分散管理するために、各 P E に並列キャッシュのコントローラとは別に、ロックディレクトリとロック関係のバスコマンドを監視するハードウェアからなるロック機構を設ける。

同時ロック可能な数は、変数同士の単一化を効率的に行うために 2 個は必要である。但し、デッドロックを回避するため、同時ロックは必ず番地の昇順に行うことにする。

ロック操作における共有バスの使用頻度を低く抑えるために次の方策を考えている。

① 並列キャッシュの状態を利用した最適化

他の P E にコピーが無い、つまり、自 P E の並列キャッシュが Exclusive 状態の番地をロックする場合には、ロックしたことを他の P E に知らせる必要がない。従って、L K バスコマンドの送出を抑止して共有バスの使用頻度を下げる事が可能である。

② ビジーウェイト方式による最適化

自 P E が既にロックしている番地を参照しようとする他 P E のバスコマンドを検出した時には、L H 応答を返

して、参照しようとした P E のビジーフラグをセットする。以後、参照に失敗した P E はビジーウェイト状態に入るため共有バスを使用しない。一方、自 P E は、L 状態からビジーウェイト中の他 P E が存在することを示す L W 状態に変更する。自 P E がアンロックする時に、L W 状態であれば、他 P E のビジーフラグをリセットするために U L バスコマンドを送出する必要があるが、L 状態ならば U L バスコマンドの送出を抑止して共有バスの使用頻度を下げる事が可能である。

4.2. ロックディレクトリの状態

ロックディレクトリは、同時にロック可能な数分のエントリ数が必要で、次の 3 状態を表わす。

① L : Lock without Waiter

自 P E がロックしている状態。

② L W : Lock with Waiter

自 P E がロックしているために、ビジーウェイト状態にある他の P E が存在する状態。

③ E : Not Locked (Empty)

ロックしていない状態 (未使用エントリ)。

4.3. ロック操作の CPU コマンド

ロック操作の CPU コマンドは、次の 3 種類用意した。

① L R : Lock and Read (ロックして読み出すコマンド)

② U : Unlock (アンロックするコマンド)

③ U W : Write and Unlock

書き込み後にアンロックするコマンド。

4.4. ロック操作のバスコマンドと応答

ロック操作のバスコマンドと応答は、次の 3 種類ある。

① L K : Lock (ロック要求バスコマンド)

② U L : Unlock (アンロックを伝えるバスコマンド)

③ L H : Lock Hit

他の P E が参照しようとした番地を、自 P E がロック中であることを示す応答。

5. 評価

5.1. Exclusive の効果

Exclusive 状態を設けることにより、並列キャッシュの I バスコマンドとロック操作の L K バスコマンドを抑止できる。この効果を前出の B U P プログラムで評価した結果、バスサイクルを約 50% 短縮できた。

5.2. Direct Write の効果

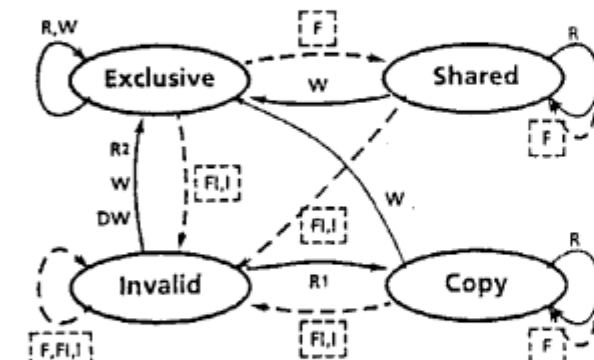
Direct Write によって、スワップインを抑止した効果と同 B U P プログラムで評価した結果、D W を用いない場合に対してバスサイクルを約 30% 短縮できた。

6. おわりに

P I M では、クラスタ内処理系と並列キャッシュの 2 つのシミュレータを用いて、K L 1 処理系のメモリ参照特性を考慮した並列キャッシュ機構とロック機構の設計を行っている。その結果、クラスタ内のハードウェア構成の中心となる並列キャッシュ機構とロック機構を上手に設計すれば、K L 1 の並列実行モデルを効率良く実現できる見込みを得た。今後は、さらに詳細なハードウェアの設計とその評価を進めると共に、クラスタ間のハードウェア構造についても検討を進めて行く予定である。

(参考文献)

- (1) 後藤, 松本, 佐藤 他, "並列推論マシン P I M", 情報処理学会第 33 回全国大会予稿集 3B-5~3B-7
- (2) P. Bitar, A. M. Despain, "Multiprocessor cache synchronization", ISCA'86



(注 1) R: 原本キャッシュからの読み出し, R2: 共有メモリからの読み出し
(注 2) Exclusive と Shared には, Clean と Modified があるため, 各 2 状態となる。

→ CPU コマンド (R: Read, W: Write, DW: Direct Write)
--- バスコマンド (I: Invalidate, F: Fetch block, FI: Fetch & Invalidate)

図 1 6 状態並列キャッシュの状態遷移図