

TM-0951

並列キャッシュにおける更新型プロトコル
と無効果型プロトコルの動的切り換え方式

佐藤 正樹

August, 1990

© 1990, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~5
Telex ICOT J32964

Institute for New Generation Computer Technology

並列キャッシュにおける更新型プロトコルと 無効化型プロトコルの動的切り換え方式

佐藤正樹*

msato@okilab.oki.co.jp

沖電気工業（株） 総合システム研究所

1 はじめに

共有バス・共有メモリ型マルチプロセッサにおいて、各プロセッサにキャッシュメモリを持たせ、処理の高速化とバストラフィックの軽減化を図る方法がよく知られている。そこでは、複数のキャッシュ間でのデータの一貫性を保証する機構が必要となり、そのための種々のプロトコルが提案されている[1]。これらのプロトコルを、共有データに対し書き込み処理が行われた場合の同一コピーを持つ他のキャッシュ上のデータの扱い方について分類すると、大きく3つに分けることができる。すなわち、他のキャッシュ上にある同一コピーのデータを更新する更新型と、他のキャッシュ上にある同一コピーのデータを無効化する無効化型と、更新型と無効化型を混在させる混在型である。本稿では、データのアクセスの局所性を利用してプロトコルの切り換えを動的に行う新しい混在型プロトコルを提案する。

2 MRU方式

更新型と無効化型にはそれぞれ次のような問題点がある。

更新型の問題点は、共有データに対する書き込みが行われ、そのデータを含むブロックを所持するキャッシュで、データの更新が行われた後、そのデータに対するアクセスがいつい生じない場合には、更新のためのバストランザクションが余計に発生したものになり、システム効率の低下を引き起こすことである。すなわち、複数のプロセッサ間で密に共有しないデータに対しては、更新型は適さない方式であると言える。

一方、無効化型の問題点は、共有データに対する書き込みが行われ、そのデータを含むブロックを所持するキャッシュで、ブロックの無効化が実施された後、そのブロック中のデータに対しアクセスが生じた場合には、再びこのブロックを共有メモリあるいは他キャッシュからロードするためのバストランザクションが発生し、システム効率の低下を引き起こすことである。すなわち、複数のプロセッサ

間で共有される度合いが非常に密接であるようなデータに対しては無効化型は適さない方式であるといえる。

これらの問題点を解決する方法として、更新型と無効化型を混在させ、プロトコルの切り換えを実行時に動的に行おうとするのが混在型である。混在型の場合には、プロトコルの選択を如何に行うかが鍵となる。今までに提案されている方式には、以下のようなものがある。

1. プログラムの性質により、プロトコルを選択する。
各キャッシュにモードレジスタを設け、プログラムでモードレジスタを書き換えることによりプロトコルを切り換える。[2]
2. データの特性により、プロトコルを選択する。ページ管理機構を利用し、ページごとにプロトコルのタイプを表す属性を付加し、ページ割付をデータの特性に応じて行う。[3]

本稿で提案する方式は、データアクセスの時間的・空間的局所性を利用するもので、プロセッサから最近にアクセスされたデータは、今後もプロセッサからアクセスされる確率が高いことが予想されるので、更新型プロトコルを選択し、プロセッサから最近にアクセスされていないデータは、今後はプロセッサからアクセスされる確率が低いことが予想されるので、無効化型プロトコルを選択するという方式である。最近アクセスされたかどうかは相対的な問題であり、比較対象をどう選ぶかによっては、膨大な処理が必要になってしまう。そこで、マッピング方式をセットアソシアティブ構成にし、比較対象をセット内で閉じることにより、簡単に上記方式が実現できる。すなわち、各ウェイトごとのプロセッサからのアクセスによるヒット判定結果を、各セットごとに記憶することにより、各セットでどのブロックが最も最近にアクセスされたかが特定できることになる。本稿で提案する方式を、キャッシュブロックの置換アルゴリズムであるLRU方式に対し、MRU (Most Recently Used) 方式と呼ぶ。

3 実現方法

ここでは、プロトコルの切り換えの実現方法を述べるにとどめ、個々のプロトコルの詳細な説明は省略するが、他の並列キャッシュと同様、ブロックが共有状態か否かを示

*Method for dynamic protocol switching in coherent cache
Masaki SATO
Systems Laboratory, Oki Electric Industry Co., Ltd.

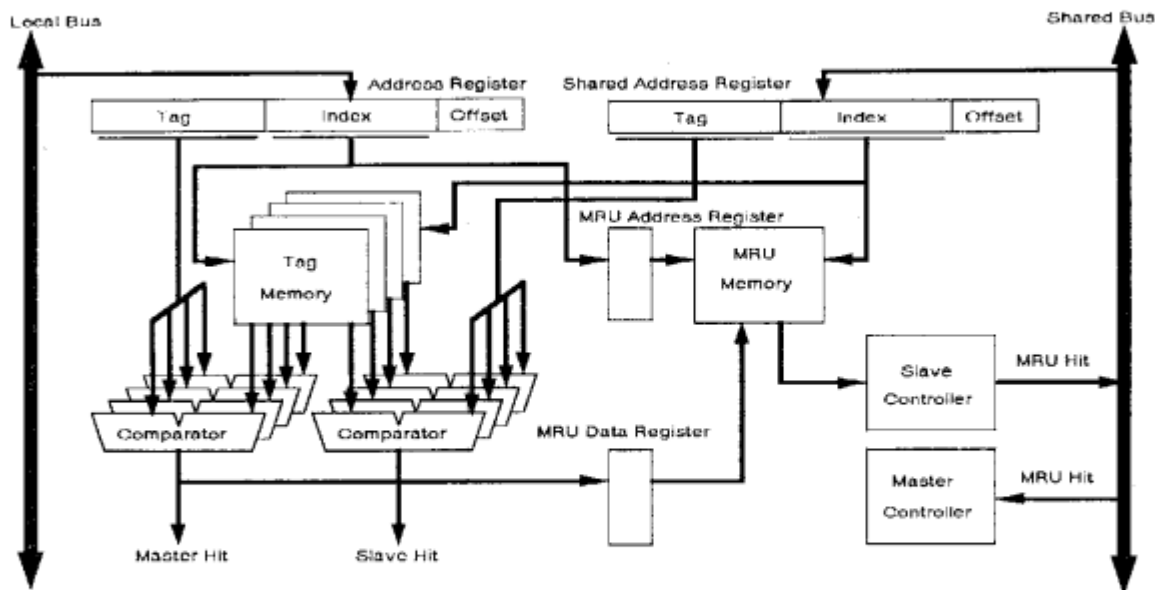


図 1: ハードウェア構成図

すフラグを持ち、ライトアクセス時のバストラフィックを削減する制御を行っているものとする。

図 1 に本方式のハードウェア構成図を示す。例として、4 ウェイセットアソシアティブの構成を示している。MRU メモリには、キャッシュに格納されている各ブロックごとの最近のヒット判定結果が記憶されている。すなわち、セット内で、最近にプロセッサからアクセスされたブロックが属するウェイに相当するビット (MRU ビットと呼ぶ) がセットされている。

共有データに書き込みを行うときのキャッシュ制御を次に説明する。

共有データにライトヒットしたキャッシュは、バス権獲得後、共有バス上に書き込みデータとアドレス及びライトコマンドを送出する。

同じアドレスのデータを持つキャッシュは、該当するブロックの MRU ビットがたっていたなら、このブロックは今後もプロセッサからアクセスされる確率が高いとみなし、更新型プロトコルを選択、バス上に流れているデータをキャッシュに書き込み、MRU ヒット線をアクティブにする。該当するブロックの MRU ビットがたっていなかったなら、このブロックは今後もプロセッサからアクセスされる確率が低いとみなし、無効化型プロトコルを選択、該当するブロックを無効化し、MRU ヒット線はアクティブにしない。

書き込み要求側のキャッシュは、MRU ヒット線の信号値により、ブロックのステータスを共有状態にするか私有状態にするかを決定する。すなわち、MRU ヒット線がアクティブの場合は、同一コピーを持つキャッシュの中で、データを更新したキャッシュがあることを示しているので、キャッシュブロックのステータスを共有状態に設定

する。MRU ヒット線がアクティブでない場合は、同一コピーを持つキャッシュはすべて該当するブロックを無効化したことになるので、キャッシュブロックのステータスを私有状態に設定する。

4 おわりに

以上、並列キャッシュにおいて、更新型プロトコルと無効化型プロトコルの切り換えを、データアクセスの局所性を利用することにより、動的に制御し、かつキャッシュブロックごとにプロトコルの選択が可能である方式について述べた。この方式によれば、プログラムの性質やデータの特徴を前もって解析する必要がなくシステムの効率をあげることが可能となる。今後はこの方式をシミュレーション等で評価する予定である。

なお本研究は、ICOT からの委託により行っている PIM/i の研究開発の一環として行われた。

参考文献

- [1] J. Archibald and J. L. Bare, "Cache Coherence Protocols : Evaluation Using a Multiprocessor Simulation Model" ACM Trans. on Computer System, Vol.4, No. 4, pp. 273 - 298, 1986.
- [2] 森脇、清水：高速並列処理ワークステーション (TOP-1) - 無効化型と更新型のスヌープキャッシュプロトコルの共存を許すための機構 -、情処 38 全大論文集、5 T-1、1989.
- [3] 福田、松本：共有バス共有メモリ型マルチプロセッサ支援機構の提案、情処 38 全大論文集、7T-5、1989.