

## 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 —

## 5T-1

森脇 淳、清水 茂則

日本アイ・ビー・エム株式会社 東京基礎研究所

## 1. はじめに

共有メモリ・共有バス型のマルチプロセッサシステムでは、バスおよびメモリへのアクセスの競合を低減するために、各プロセッサにプライベート・キャッシュ（以下簡単にキャッシュと呼ぶ）を付加するという構成がよく用いられる。この構成では、同一メモリ・ロケーションに対する複数のコピーがシステム中に存在することが許されるので、これら複数のコピーが一致すること（コンシステンシ）を保証する方法が問題となる。その中でも特に、複数のコピーのひとつが更新されたときに残りのコピーをいかに扱うかという点に注目すると、既存の方法には無効化と更新のふたつの手法があり、このふたつの手法の優劣は一概に決めることはできない。

一方、コンシステンシの問題を各キャッシュごとに分散的に制御することにより解決する、スヌープ・キャッシュと呼ばれる方法が近年広まってきている。この方法では集中型の制御方法に比べて各キャッシュごとの制御がローカルに行えるかわり、複雑な制御を必要とする。

本稿では、プログラムの実行時に各キャッシュごとに無効化か更新かのいずれかの手法を選択するという、一方のみのときに比べてより柔軟な手法を提唱する。そしてその手法のスヌープ・キャッシュにおける実現のしかたについて論じ、最後に我々の試作したTOP-1における事例について述べる。

## 2. 無効化手法と更新手法の比較

複数のキャッシュの間で同一メモリ・ロケーションのコピーが存在するとき、対応するデータを共有データとよぶ。キャッシュにおいて共有データに対するライト・ヒットが発生したときに、コンシステンシを保つために、他のキャッシュに存在するコピーを全て無効にする手法を「無効化手法」、全て同一の内容に更新する手法を「更新手法」と呼ぶ。

以後ライトスルーでなくコピーバックのキャッシュの場合のみを考える。この場合、キャッシュを有するマルチプロセッサシステムの効率を決める重要な要因にキャッシュのヒット率とキャッシュ間のデータの共有率がある。ヒット率が高いほどキャッシュ・ミスによるバス・アク

セスが少なく、共有率が低いほど共有データに対するライト・ヒットによるバス・アクセスが少なく、ともにバスの競合が減少するため効率がよい。この観点からふたつの手法を比較してみると、あきらかに更新手法のほうが無効化手法に比べてヒット率を高くするが、共有率も高くする。したがって、どちらの手法のほうが良い効率が得られるかを一概に決めることはできない。むしろプログラムの性質を考慮した上でふたつの手法のトレード・オフを論じなければならない。

同様に、これらふたつの手法をとるスヌープ・キャッシュを各々無効化型および更新型のスヌープ・キャッシュとよぶことにする。初期のスヌープ・キャッシュは実現の容易さから無効化型が多かったが、最近ではより高いヒット率が得られる更新型を用いることが多い。しかし、さきに述べたように更新型は共有率が高くなるので、必ずしも無効化型よりもよい効率が得られるとはいえない。

## 3. 無効化手法と更新手法の共存を許す方法

プログラムの実行中にそのプログラムの性質に応じて無効化手法と更新手法を切り換えることができるならば、より良い効率を得ることができる。そのための自然な方法として、キャッシュを制御する機構を各々の手法ごとに別々に実現して実行時にどちらか一方を用いるように切り換えると言う方法が考えられる。しかし、この方法には次のふたつの問題がある。

- (1) 実行中の各時点で無効化手法と更新手法のどちらが有効かはシステム全体でひとつに決めるべきものではなく、むしろ各キャッシュごと（各プロセッサごと）に決めるべきものである。
- (2) ふたつの手法の間で切り換えている途中に発生するであろう共有データにたいするライトに関してもコンシステンシが保証されねばならない。特にスヌープ・キャッシュにおいては、分散型の制御を行うがゆえに、(2)が問題となる。

以上の問題を解決するために次のような方法をとる。各キャッシュにひとつの属性を設け、この属性は「無効化モード」もしくは「更新モード」のいずれかをとることにする。あるキャッシュにおいて共有データに対するライト・ヒットが発生したときに、他のキャッシュに存

High-Performance Multiprocessor Workstation (TOP-1)

-- A Mechanism for Coexistence of Invalidate- and Update-type Snoop Cache Protocol --,

Atsushi Moriwaki, Shigenori Shimizu

IBM Research, Tokyo Research Laboratory

在するコピーはそのキャッシュが無効化モードを持つとき無効化され、更新モードを持つとき同一の内容に更新されるとする。明らかにこの方法は任意の無効化モードと更新モードの組合せに対してにコンシステンシーを保証するので、キャッシュを制御する機構が実際に実現できれば(1)の問題は解決する。すなわち、無効化と更新の両手法が同時に共存することになる。さらに、この実現が次の条件を満足すれば(2)の問題は解決する。

(3) 各キャッシュの属性を切り換える操作と共有データにたいするライトとは排他的に制御される。

スヌープ・キャッシュのような分散型の制御を行う場合においても(3)さえ満たされているならばキャッシュ間の同期をとることなく属性の切り換えができ、すなわち実行時のダイナミックな属性の切り換えが可能となる。

#### 4. スヌープ・キャッシュの場合の実現方法

周知のごとく、スヌープ・キャッシュは、各プライベート・キャッシュに付加されるローカルな制御装置が共有バス上の信号を用いて互いに通信しながら動作し、システム全体としてコンシステンシーを保証する動作を行うことにより実現される。このときに用いられる通信の規約(スヌープ・プロトコル)は、普通、無効化型のキャッシュと更新型のキャッシュとは異なったものとなる。無効化手法と更新手法との共存するキャッシュの制御を実現するためには、このふたつの手法に共通のプロトコルを定義し、各制御装置が自らの属性をよっていづれの手法に基づいて動作するかを選択するようにしなければならない。特に、一般にキャッシュを用いたシステムにおいては制御装置の動作を決めるための情報をキャッシュの各ラインごとにタグ情報として用意するので、共通のプロトコルは制御装置がこのタグ情報を矛盾なく管理できるように定めなくてはならない。

制御装置は、プロセッサからの要求に基づいて動作し他の制御装置に要求信号を伝える「マスター」ステートマシンと、他の制御装置からの要求信号に基づき動作(無効化もしくは更新)し返答信号を返す「スレーブ」ステートマシンのふたつにより構成される。ふたつの手法が共通のプロトコルのもとに実現するためには、マスターの動作とバス上の信号の意味は同じでなければならず、スレーブの動作のみが無効化か更新かの手法に依存する。つまり、無効化モードおよび更新モードの属性が動作におよぼす影響はスレーブ側に限らなければならない。

次に、以上に述べたプロトコルと制御装置を実現した事例を挙げる。

#### 5. TOP-1における実現

TOP-1においては、無効化モードと更新モードとの区別を各制御装置中のモード・レジスタの値を用いて選択する。要求信号を受けた制御装置のスレーブは該当する共有データのコピーがキャッシュ中に存在するならば、自らのモード・レジスタの値によってそのコピーを無効化するか同一の値に更新するかのいずれかを行なう。

一方TOP-1においては通常のスヌープ・キャッシュと同様に、キャッシュのラインにそのデータが共有されているか否かをしめすタグ情報を付随させることによりライト・ヒットに起因するバス・アクセスを最小限に抑える制御を行なっている。そのため、あるキャッシュの共有データにライト・ヒットが発生した場合、制御装置のマスターは返答信号の値によりこのキャッシュ中のコピーを共有データか否かタグで指定しなおす。ただし返答信号としては、スレーブは同一のコピーを持ちかつ更新モードのときにのみ真値を返しそれ以外のときには偽値を返す。そしてバス上でこれらの論理和をとったものがマスターに返され、真値ならば共有データとすべきことをあらわす。以上のようなプロトコルを用いるため、モード・レジスタの書き換えを行なっている間のコンシステンシーが問題となるが、スレーブが動作中はモード・レジスタへの書き込みを待たすことによりこの問題を解決している。

#### 6. むすび

以上、無効化手法と更新手法を共存させる方法について、その最も簡単に思える実現をどうして述べた。しかし、この機能を実際に用いてシステムの効率を上げるためには、以下の件に関する研究が不可欠である。

- ・各々の手法が有効な状況はいかなる場合か
- ・手法を静的もしくは動的に選択するにはいかなるアルゴリズムを用いるべきか

また手法をキャッシュごとに選択することが妥当か否かも不明であり、仮に妥当であっても、手法の区別がキャッシュごとに選択できるだけで十分とはいえない。手法を選択する単位については、例えば以下のような議論の余地がある。

- ・手法をキャッシュのラインごとに選べないか
- ・要求信号をだす制御装置が共有バスの信号を使って他のキャッシュの手法を選択できないか

本稿ではひとつの実現を述べたに過ぎないが、プロトコルを実行時の状況に応じて動的に換えるという試みは、共有バス・共有メモリ方式のマルチプロセッサシステムの性能を上げるための方法として、今後の研究を通してその是非を問うべきものである。

#### 【参考文献】

- [1] 大庭, 清水: 高速並列処理ワークステーション(TOP-1) —— スヌープ・キャッシュ ——, 第37回情報処理学会全国大会論文集(1988).
- [2] P.Sweazey, A.J.Smith: A Class of Compatible Cache Consistency Protocols and their Support by the IEEE Futurebus, in the Proceedings of the 13th international Symposium on Computer Architecture, IEEE, 1986.
- [3] A.R.Karlin, M.S.Manasse, L.Rudolph, and D.D.Sleator: Competitive Snoopy Caching, in the Proceedings of the 27th Annual Symposium on the Foundations of Computer Science, IEEE, 1986.