

バス結合型マルチプロセッサ用 キャッシュ一致保証方式の性能評価

武和秀仁 小倉真 森岡道雄 黒澤憲一
(株)日立製作所 日立研究所

本報告では、サーバ機分野で一般的であるバス結合型マルチプロセッサのキャッシュ一致保証方式の性能評価について報告する。今回対象としたキャッシュ一致保証方式は、キャッシュ一致制御をプロセッサ間で同期して行う同期型キャッシュ一致保証方式と非同期型キャッシュ一致保証方式である。この2方式には、同期型は制御が容易でハード量も少なく済むが性能があまり出ない、一方非同期型は制御が複雑でハード量も多いが性能面で優れているという違いがある。しかしながら、両方式の動作を定量的に解析し、性能などを明かにした報告は今まで見当たらない。本報告では、マルチプロセッサシミュレータにより、両方式でのシステムの動作を模擬し、具体的な性能値などを明かにした。

Performance Evaluation of Cache Coherency Check for Multiprocessor Systems with Bus Architecture

Hidehito Takewa, Shin Kokura, Michio Morioka, Ken'ichi Kurosawa
Hitachi Research Laboratory, Hitachi, Ltd.

This paper is to report performance evaluation of the cache coherency check for multiprocessor systems with bus architecture. We compared the synchronous cache coherency check with the asynchronous cache coherency check. The synchronous cache coherency check has simple and small hardware, but is not of good performance. The asynchronous one needs complex and large hardware, and it is of better performance than the former. However, no paper reports quantitative analysis of the difference between the two. In this paper, by using a simulator, we show the difference of performance quantitatively.

1. はじめに

プロセッサ単体の性能は、スーパースカラ、VLIW(Very Long Instruction Word)などの新しいアーキテクチャの提案や、動作周波数の向上などのLSI技術の進歩により年々向上している。また、価格面では、PC、WS、サーバなどの市場の拡大により、プロセッサの低下価格化が進んでいる。この様なプロセッサの高性能化、低価格化を背景に、サーバ機の分野で、これらプロセッサを多数結合したマルチプロセッサシステムの製品化が相次いでいる。従来から様々なマルチプロセッサシステムが提案されているが、サーバ機では共有メモリ型で、かつ、バス結合型のマルチプロセッサが大部分である。

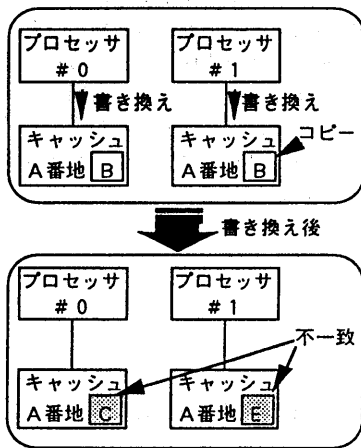


図1 キャッシュコヒーレンシ

マルチプロセッサシステムでは、図1のように、プロセッサがキャッシュを個々に持ち、それぞれが独立にキャッシュにアクセスするので、同一のメモリのコピーであっても個々のキャッシュ内では内容が異なる場合が生じる。システムを正常に動作させるためには、キャッシュ間での内容の相違を適当に制御し、キャッシュ間でデータの一貫性を保証することが必要になる。

このキャッシュ間でデータの一貫性を保証するキャッシュの一致保証方式には、共有キャッシュ方式、キャッシュ不可データ方式、スヌープバス方式、ブロードキャストライト方式、ディレクトリ方式がある。各方式の詳細、長所、欠点などは、関係する文献等を参考にして頂きたい。¹⁾

今回のシミュレーションで評価した方式はスヌープバス方式である。この方式では、各プロセッサ

が、マルチプロセッサバスに流れるメモリ書き込み要求を監視する。メモリの書き込み要求がバスを流れ、そのコピーをキャッシュが持っていたとすれば、そのコピーを無効にする。この方式には、監視処理を各プロセッサで同期して行う同期型キャッシュ一致保証方式と非同期で行う非同期型キャッシュ一致保証方式がある。二つの方式の詳細な説明は次章に譲るとして、簡単に両者の違いを述べると、同期型は制御が容易でハード量も少なく済むが性能があまり出ない、一方非同期型は制御が複雑でハード量も多いが性能面で前者よりも優れていると言える。しかしながらこの様に、両者の相違を定性的に述べることは容易であるが、両方式を採用したシステムの動作を定量的に解析し、非同期方式による性能向上率などを明かにした報告は今まで見当たらない。本報告では、マルチプロセッサシミュレータにより、両方式でのシステムの動作を模擬し、得られたデータから具体的な性能値などを明かにした。

以下、2章では2方式の詳細な説明、3章では、シミュレーションモデルの説明、4章では、シミュレーション条件の説明、5章では、シミュレーション結果の解析、評価、6章では解析、評価のまとめを行う。

2. キャッシュ一致保証方式

2.1 キャッシュ一致保証トランザクション

キャッシュの一貫性の問題は、キャッシュ内のコピーの書き換えに起因する。その観点からバスに流れるトランザクションは、キャッシュ一致保証制御の対象となるトランザクションと対象とならないトランザクションに分けることができる。例えば、前者はデータに関するアクセスであり、後者は命令に関するアクセスである。前者のキャッシュ一致保証制御の対象となるトランザクションを、以下キャッシュ一致保証トランザクションと呼ぶこととする。バススヌープ方式のシステムでは、プロセッサがバスを監視しており、キャッシュ一致保証トランザクションの発行をトリガに、プロセッサがキャッシュ一致保証制御を開始する。ここまでの処理は、同期、非同期とも同じである。

2.2 同期型キャッシュ一致保証

次に、同期型キャッシュ一致保証方式について

説明する。バスのスヌープをトリガに、キャッシュ一致保証制御を開始したプロセッサは、自キャッシュを調べ判定をし、一定の時間後にキャッシュの判定結果をバスに出力する。この一定時間をキャッシュ一致制御遅延時間と呼ぶ。キャッシュには、プロセッサからのリードライト (R/W)、メモリからのデータ登録、メモリへの登録データの吐き出し (リプレース) などのアクセスが起きる。キャッシュへの同時アクセスは禁止であるため、キャッシュ一致保証制御によるアクセスは禁止される。そのためプロセッサがキャッシュ一致制御遅延時間に判定結果を出せない場合が生じる。その際、プロセッサはキャッシュ一致保証延長信号を出して、他のプロセッサに通知する。同期方式では、キャッシュ一致保証遅延時間内および、キャッシュ一致保証延長信号が出ている時は、プロセッサからの次のキャッシュ一致保証トランザクションの発行は禁止される。このため、各プロセッサは、キャッシュ一致保証トランザクションを1つずつ処理をし、キャッシュの判定結果を返す。従って、同期型キャッシュ一致保証方式を実現に必要なハードは、図2に示すように、バスの監視ハードと判定ハードに加えて、監視したキャッシュ一致保証トランザクションを一致保証制御処理の間、記憶しておくバッファ (以下CCC Queueと呼ぶ) を1段用意すれば良い。

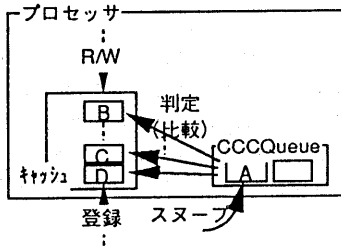


図2 ハード構成 (同期型)

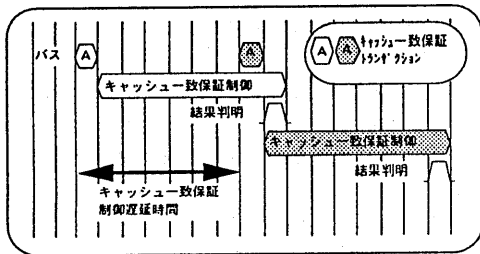


図3 動作例 (同期型)

同期型キャッシュ一致保証方式の一連の動作を図3に示す。最初のキャッシュ一致保証トランザ

クションが発行されたあと、判定結果が出るまでは、次のキャッシュ一致保証トランザクションが発行されていない。この時、キャッシュ一致保証制御はトランザクションを1つずつ処理する機能があれば良い。そのため、CCC Queueは一段で十分となる。

2.3 非同期型キャッシュ一致保証方式

次に、非同期型キャッシュ一致保証方式について説明する。前節で説明した同期型キャッシュ一致保証方式は、キャッシュ一致保証トランザクションの処理でバスを占有してしまう欠点があった。非同期型方式は、この欠点を改善するために考え出された方式である。この非同期型方式では、キャッシュ一致保証制御が動作している間でも、別のキャッシュ一致保証トランザクションの発行を許可することで、同期型の欠点を解決している。従って、非同期型キャッシュ一致保証方式を実現するには、同期型に必要なハードに加えて、図4に示すように数段 (図4では3段) のCCC Queueが必要になる。

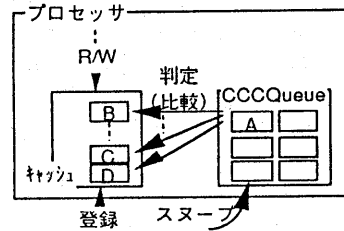


図4 ハード構成 (非同期型)

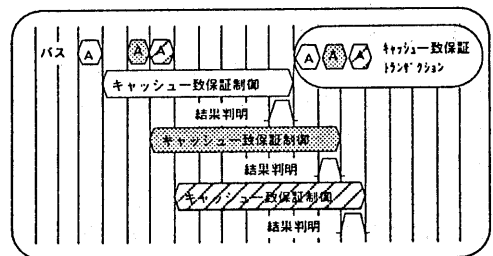


図5 動作例 (非同期型)

非同期型キャッシュ一致保証方式の一連の動作を図5に示す。キャッシュ一致保証トランザクションの判定を待たずに次のキャッシュ一致保証トランザクションが発行されている。この時、キャッシュ一致保証制御は、複数のトランザクションを処理する機能が必要となる。そのため、プロセッサ内に複数段のバッファを設けて、トランザクシ

ンを記憶する。記憶したトランザクションは逐次処理される。これにより、キャッシュ一致保証トランザクションによるバスの占有を解消している。

この様に、非同期型は、同期型の欠点を改良を目的に提案された方式で、性能は改善されるが、同期型と比べて、必要ハードが増加し、制御も複雑になる。

3. シミュレーションシステム

図6に今回使用したシミュレーションシステムの構成を示す。以下、図中の構成要素を簡単に説明する。

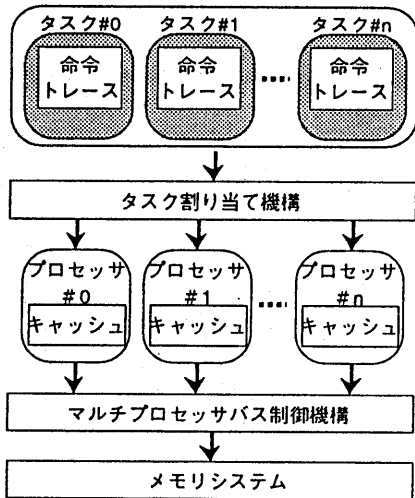


図6 シミュレーションシステム

3.1 命令トレース

単一プロセッサの実機で採取されたもので、命令トレースの採取は、分岐命令や、割り込みなどプログラムの流れが変わるポイントとメモリ参照が発生したポイントで、プロセッサにトラップをかけることで実現される。今回使用したトレースは、OLTP(Online Transaction Processing)用のベンチマークプログラムであるTPC-Bを実際の計算機で実行し、ユーザタスク、OS処理を含めて実行した命令列を採取したものである。

3.2 サーバプール

上記の1つの命令トレースを1つのサーバタスクと仮定した。タスクで発生するメモリアクセスに関しては、ユーザ空間へのアクセスは、オフセットを加算することでタスク毎に異なるアドレスと

なるようにした。一方カーネル空間へのアクセスは、全てのタスクで同じアドレスとなるようにした。

3.3 タスク割り当て機構

タスク割り当て機構は、サーバプール内のサーバタスクを各プロセッサに割り当てる。この割り当てられるタスクを選ぶタスクスケジュールの方法は指定することができる。プロセッサは、割り当てられたサーバタスクの命令トレースを最後まで実行すると、次のタスクが割り当てられる。

3.4 プロセッサ、マルチプロセッサ バス制御機構、メモリスシステム

プロセッサは命令トレースに従って、命令アクセス、データアクセスを実行する。キャッシュメモリにミスヒットした場合は、メモリスシステムに対してライン転送を要求する。マルチプロセッサバス制御機構、メモリスシステムは、システムクロックに従ってメモリスシステムをシミュレーションする。これにより、アクセス毎のレイテンシやプロセッサバスやメモリバスでの競合による待ち時間を評価している。

4. シミュレーション条件

マルチプロセッサシステムでキャッシュミスが生じると、プロセッサは、バスにトランザクションを発行する。発行されたトランザクションは、別のプロセッサにスヌープされ、キャッシュの内容と比較チェックされる。また、発行されたトランザクションはメモリコントローラに受信され、メモリアクセスを起こす。

このキャッシュミスに起因する一連の動作では、様々な箇所処理待ちが起きる。その主な競合(処理待ち)を、図7に示す。

CPU内競合では、キャッシュが他のトランザクションのスヌープによる判定を行っているためキャッシュアクセスが待たされる場合、また、自プロセッサがバスの使用待ちの状態であるためトランザクションを発行使用が待たされる場合などで、競合の原因が自プロセッサにある。

バス競合は、バスが他プロセッサに使われているためトランザクション発行のためのバスの使用が待たされる場合、また、メモリコントローラから応答の送信のためのバスの使用が待たされる場合

も含む。

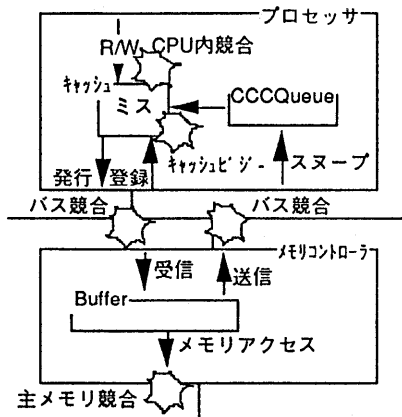


図7 競合各種

主メモリ競合は、メモリコントローラに受信されたが、バッファに滞留している場合、メモリバスの使用権を待っている場合などがある。

キャッシュビジーは、スヌープでのキャッシュの判定が、キャッシュへの登録、R/Wでのアクセスのため待たされる場合である。

これらの競合は、キャッシュサイズなどパラメータが複雑にからんでいるため、同期型、非同期型での違いで、どの競合による待ちサイクルが増えるか定量的には言えなが、同期型ではバス競合の待ちサイクルに含まれるキャッシュ一致制御遅延時間分の待ちが増加すると考えられる。

表1 シミュレーションシステム

分類	項目	内容
全体	プロセッサ、メモリコントローラ、主メモリの動作周波数の比	2 : 1 : 1
	マルチプロセッサ同期化オーバーヘッド	考慮せず。
キャッシュメモリ	キャッシュラインサイズ	32バイト
	キャッシュ方式	ライトインバリデート
	キャッシュ初期状態	ホットスタート
	キャッシュミスからプロセッサバス動作までのペナルティ	3プロセッササイクル
	キャッシュ一致保証遅延時間	5バスサイクル
	スヌープ動作によるプロセッサへの影響	なし。
メモリコントローラ	リード受信バッファ	プロセッサ台数分
	ライト受信バッファ	2段
	メモリコントローラ通過遅延	2バスサイクル

今回のシミュレーションシステムでは、これらの競合が生じるポイントで、待ちサイクルをカウントしてシステム性能を測定する。待ちサイクルは、競合が起きなかった時のランザクションの発行、スヌープ、メモリアクセスなどに費やしたサイクルに、競合により生じた待ちサイクルを合計したものとなる。

非同期型は、同期型のバスを占有するという欠点を改善したものである。そのため、プロセッサバスの性能がシステムの性能を左右するモデルでシミュレーションを実施することが非同期型への変更による効果を顕著にすると考えられる。そういった見地からシミュレーション条件を設定した。設定した条件を表1に示す。

5. シミュレーション結果の解析および評価

同期型と非同期型では、その方式の違いは、プロセッサバスでの競合による待ちサイクル数（ペナルティ）に表われる。プロセッサバスでのペナルティには、プロセッサのバスの使用権待ち、キャッシュ一致保証制御の終了待ち、キャッシュ一致保証制御の判定到着待ち、メモリコントローラのバスの使用権待ちがある。同期型から非同期型に変更すると、キャッシュ一致保証制御の終了待ちが無くなり、バスの競合が改善される。

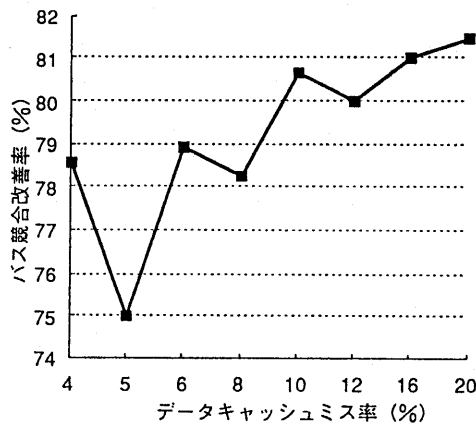


図8 非同期型方式による競合の改善率

図8は、非同期型キャッシュ一致保証方式によるマルチプロセッサバスでの競合の改善率を測定したものである。データキャッシュミス率が大きいとキャッシュ一致保証ランザクションの発行

が多くなるため、その分改善率が良くなる。測定結果によれば、非同期型に変えることで、バスの競合が約80%から約70%改善されることが判明した。このときのシステム全体の処理性能を図9に示す。方式を変更することで、システム全体の性能は8%から15%にアップした。

参考文献

1) B.Wilkinson,高橋義造 監訳：計算機設計技法：'94：トッパン

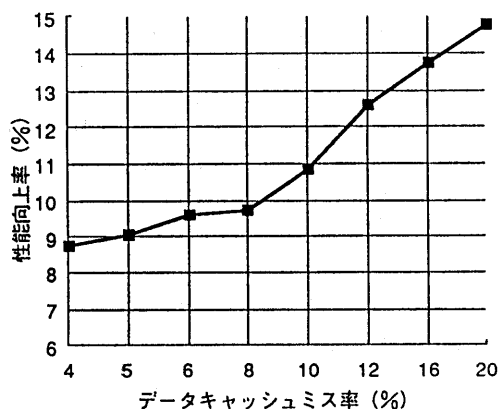


図9 非同期型方式による性能改善率

6. まとめ

バス結合型マルチプロセッサシステムで用いられるバススヌープによるキャッシュ一致保証方式で、プロセッサ間で同期を取って一致保証制御を行う、同期型キャッシュ一致保証方式と、同期を取らない非同期キャッシュ一致保証方式について、性能面で比較および評価を行った。評価にあたっては、システム全体の性能が判るように、バス競合、主メモリ競合などのペナルティが測定できるシミュレータを使用した。

シミュレーションの結果、データキャッシュミス率が大きくなると、キャッシュ一致保証方式の相違がはっきり表われ、非同期型キャッシュ一致保証方式では、マルチプロセッサバスでの競合によるペナルティが70%から80%改善される事が判明した。また、この時、システム全体性能は、約8%から15%向上した。

シミュレーションでは、マルチプロセッサバスの性能がボトルネックとなるシステムモデルで実施しているため、実際のシステムとは、若干異なる設定がある。今後、実際のシステムにより近いモデルでシミュレーションを行い、両方式の詳細な性能分析を行っていく予定である。