

投機的にコヒーレンス制御を行う DSM システムの性能評価*

鈴木 圭介 多田野 陽介 乗貞 由華 古川 文人
 大津 金光 横田 隆史 馬場 敬信 †
 宇都宮大学工学部情報工学科 ‡

1 はじめに

我々は、DSM システムにおけるリモートメモリへのアクセス(以下、リモートアクセス)によるレイテンシ(以下、リモートレイテンシ)を短縮することを目的とした、新たな投機的コヒーレンス制御機構^[1]を提案してきた。

これは、従来までの read アクセスのみを投機の対象とした投機的コヒーレンス制御と違い、write と read の両方のメモリアクセスにより発生するリモートレイテンシを短縮することを目的としたものである。

本稿では、2つのマイクロベンチマークと SPLASH-2 の Barnes, Raytrace を用いて本機構のシミュレーションを行うことにより性能評価を行う。

2 投機的コヒーレンス制御機構

2.1 DSM システムの構成

我々が提案している DSM システムの構成を図 1 に示す。この DSM システムは、複数のノードがネットワークで結合されたものである。各ノードは、プロセッサコア、キャッシュ、DSM コントローラ、ディレクトリおよびメモリ、ネットワークインターフェース、および投機的コヒーレンス機構(以下、SCCM(Speculative Coherence Control Mechanism))から構成される。

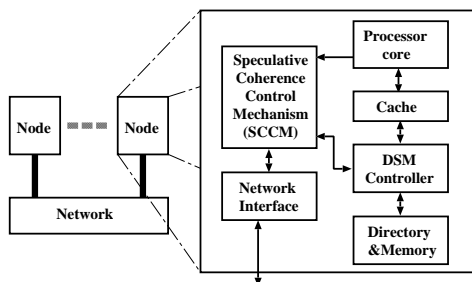


図 1: DSM システムとノード内の構成

キャッシュコヒーレンスプロトコルは無効化型である。キャッシュは Modified, Exclusive, Shared, Invalid(以下、M, E, Sc, I)の4状態をとる。また、メモリブロックは, Uncached, Shared, Private, Busy-Shared, Busy-Private (以下、U, Sm, P, BS, BP)の5状態をとる。

* Performance Evaluation of the Speculative Coherence Controller for a DSM system

† Keisuke Suzuki, Yousuke Tadano, Yuka Norisada, Fumihito Furukawa, Kanemitsu Ootsu, Takashi Yokota, and Takanobu Baba

‡ Department of Information Science, Faculty of Engineering, Utsunomiya University

2.2 SCCMによる投機的コヒーレンス処理

SCCMは、以下の5つの投機的コヒーレンス処理を行う。

- (1) speculative self-downgrade
- (2) speculative self-invalidate
- (3) speculative send in shared state
- (4) speculative send in exclusive state
- (5) speculative upgrade

これらの処理は、producer-consumer sharing(以下、PC共有状態)と migratory sharing の2つの共有状態にあるデータへのメモリアクセスにおいて、read および write 両方のメモリアクセスに要するコヒーレンス処理を投機的に行うものである。

SCCM内では、VMSP^[2](Vector Memory Sharing Predictor)とLTP^[3](Last-Touch Predictor)をそれぞれ独自に拡張することにより、投機的コヒーレンス処理の種類、開始のタイミングを予測する。

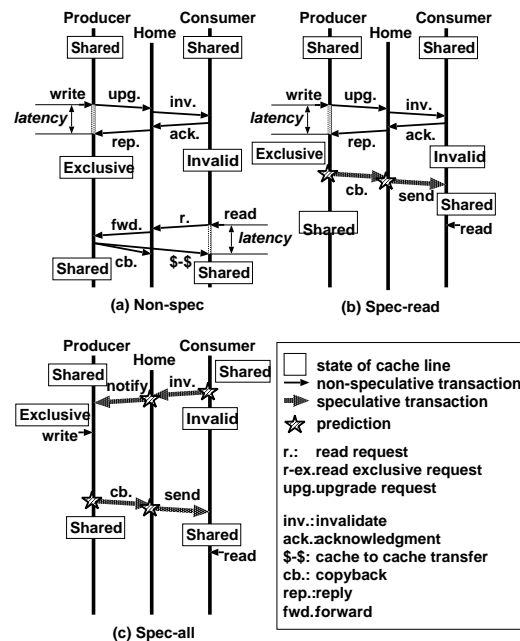


図 2: PC 共有状態における投機的コヒーレンス処理の例

図2は、コヒーレンス処理の流れを示したものである。PC共有状態とはあるメモリブロックに対し、Producer ノードが write を行い、その後 Consumer ノードが read を行う共有パターンである。

図2(a)は、PC共有状態にあるデータ(以下、PC共有データ)へのメモリアクセスを従来のコヒーレンス処理(以下、non-spec)で解決する実行例である。この例

では、Producerによるwriteアクセスが開始された後、Consumerの所有するラインの無効化が始めて開始される。また、Consumerによるreadアクセスが開始された後、Producerへのread要求が初めて開始される。

図2(b)は、readアクセスのみ投機の対象とした場合(以下、spec-read)である。Consumerによるreadアクセスが開始される前に予測をし、予め該当するラインをScに遷移させている。これにより、readアクセスによるレイテンシは存在しないが、writeアクセスによるレイテンシは残る。

一方で、図2(c)は、write、read両方のアクセスと対象とした場合(以下、spec-all)である。Producerによるwriteアクセスが開始される前に予測し、該当するラインをEに遷移することによりレイテンシを削減している。

図で示す通り、spec-allではwrite、readアクセスによるレイテンシは削減されているが、予測の回数が増えていることがわかる。

3 評価

3.1 評価方法

評価の目的は、non-spec、spec-readと比べて、spec-allの性能向上を実験により明らかにすることである。

本稿では、マイクロベンチマークのpcとmgrとSPLASH-2のBarnes、RaytraceをCC-NUMA型並列計算機シミュレータRSIMによりnon-spec、spec-read、SCCMを実装したRSIM上においてシミュレーションすることにより評価を行った。pcとmgrはそれぞれPC共有状態とM共有状態を繰り返すプログラムである。評価パラメータを表1に示す。

表 1: 評価パラメータ

ノード数		32台
プロセッサコア		4-way スーパースカラ
データキャッシュ(ノンブロッキング)		
サイズ	L1	64kバイト
	L2	8Mバイト
アクセス レイテンシ	L1	タグ
		データ
	L2	タグ
メモリアクセスレイテンシ		80サイクル
ネットワーク		
パケットヘッダサイズ		8バイト
トポロジ		2D-メッシュ
リンク幅		2バイト
1ホップ	データなし	88サイクル
	データ付き	184サイクル

3.2 評価結果

図3、図4にSCCMでの、pc、mgr、Barnes及びRaytraceの実行時における性能向上比及び、readアクセス、writeアクセスにおける平均メモリアクセスレイテンシを示す。それぞれ、non-specの実行速度を基準とした。また、平均メモリアクセスレイテンシは、PC共有データまたはM共有データへのメモリアクセスの平均レイテンシとしている。この結果から、spec-allがnon-specに対し、pcで1.59倍、mgrで1.36倍、Barnesで1.01倍、Raytraceで1.02倍となった。また、spec-readに対してpcにおいて1.14倍となった。この理由として、non-spec

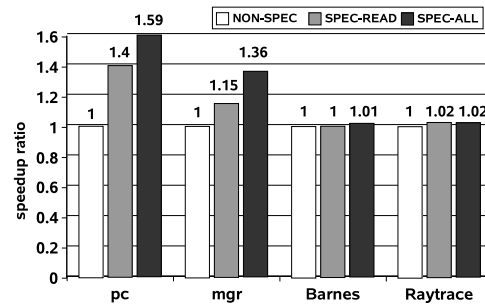


図 3: 性能向上比

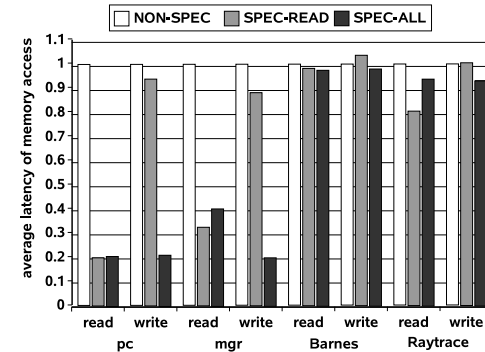


図 4: 平均アクセスレイテンシ

に比べ、spec-allではread、writeの平均メモリアクセスレイテンシがpcでは77%、Raytraceでは6%削減されていることが挙げられる。

4 おわりに

本論文では、readとwriteアクセスを投機の対象とする投機的コヒーレンス制御機構SCCMを備えたCC-DSMシステムを評価した。

その結果、従来のコヒーレンス制御に対して、性能向上比は最大1.59倍になることがわかった。

今後の課題は、投機の予測成功率に関する調査と、プロセッサの台数を増やして評価を行うことである。

謝辞 本研究は、一部日本学術振興会科学研究費補助金(基盤研究(B)14380135, 同(C) 14580362, 若手研究14780186)の援助による。

参考文献

- [1] 古川文人, 他:DSMシステムにおける投機的コヒーレンス制御機構の提案と評価, 情報処理学会研究報告 2001-ARC-142 2001-HP-85, pp.169-174(2001)
- [2] A. Lai, et al. :Memory Sharing Predictor: The Key to a Speculative Coherent DSM, Proc. of the 26th Ann. Int. Symposium on Computer Architecture (1999)
- [3] A. Lai, et al. :Selective, Accurate and Timely Self-Invalidation Using Last-Touch Prediction, Proc. of the 27th Ann. Int. Symposium on Computer Architecture(2000)