スレッドレベル投機実行を支援する キャッシュ機構に関する研究

越前 智史 (210036)

北陸先端科学技術大学院大学 情報科学研究科

2004 年 2 月 13 日

キーワード: キャッシュ，スレッドレベル投機実行，同期.

1 はじめに

現在，スーパースカラプロセッサによって命令レベルの並列性を利用したプログラム実行が主流であるが，近傍の命令間の依存関係により並列度に限界がある．そこで，プログラムからスレッドレベルの並列性を抽出するアーキテクチャとして，スレッドレベル投機実行型チップマルチプロセッサの研究がある．これにより，従来並列処理を行うことが困難であった逐次プログラムに対し，依存関係が完全に解決されていないスレッドを投機的に実行することによりスレッドレベルの並列実行が可能となる．スレッド投機実行を行う機構では，依存関係に違反する処理が検出された場合は，そのスレッドの処理を取り消すことでプログラムの実行の正しさを保証する．本稿では，投機メモリを支援するためのアクセス違反検出機構を備えた分散キャッシュを検討し，効率の良いメモリシステムを提案する．

2 スレッド投機実行

スレッド投機実行を行うアーキテクチャのキャッシュ機構は，キャッシュ本来の機能に加え，投機的なメモリアクセスによる依存関係の違反を検出する役割がある．検出機構の一例として，既存のキャッシュラインにデータが投機的に読み出されたことを記録する R ビットとデータの投機度を表す Spn が追加されたキャッシュ構成がある．投機度 Spn は値が大きいほどより投機状態であることを示し，投機度 0 の場合は Head と呼ばれる投機ではない状態を示す．分散スヌープキャッシュにおけるデータ依存違反の検出は，ストア情報を他のキャッシュに伝搬することによって行われる．ストアが伝搬される際，あるラインに関して書き込みの投機度が既存の Spn 値より低く，R がセットされている場合に当該キャッシュを使用するスレッドは投機メモリアクセス違反となる．

3　キャッシュアーキテクチャの使用設計

3.1　データ依存の同期

投機スレッドが依存違反を起こした場合，違反を起こしたスレッドとともに，それを親とする全ての子スレッドも違反を起こしたものとして破棄（squash）される．squash が発生した後，スレッドの親子間のストア伝搬履歴を使用して更なる squash を抑制する機構を提案する．キャッシュのラインに新たな制御フィールド（Curr，Prev，Sq）を設ける．これらの制御フィールドはスレッド間のデータ依存に関する同期をとるために使用する．squash が発生した後の実行では，親スレッドは当該ラインへのストアの回数（Curr）が squash 前の実行時のストア回数（Prev）より小さい場合，そのラインに対して同期不成立となりストアの伝搬を行わない．一方，子スレッドにおいて Sq ビット（squash 時にセット）がセットされたラインにロードアクセスした場合，実行がストールする．親スレッドからのストアの伝搬があったとき Sq ビットはクリアされ，ストールが解除されて，このスレッド間データ依存の同期機構により squash の回数が軽減される．

3.2　冗長キャッシュによる効率化手法

従来の投機実行では，Head スレッドの終了後に新たなスレッドを開始するが，これらのスレッド間には並列度に相当する投機度の差が存在するため，実行開始前に当該キャッシュの全てのラインを無効化および必要ならばライトバックする必要がある．この無効化およびライトバック処理のオーバヘッドを削減するために，冗長な分散キャッシュを構成する．スレッド実行に割り当てられていないキャッシュは待機キャッシュとして振舞い，ストアの伝搬による最新の情報をスヌープし，Head のスレッドが終了した際，次の新しいスレッドが直ちにそのキャッシュを利用して実行を開始する．終了した Head スレッドが使用したキャッシュに関する無効化／ライトバック処理は他のスレッド実行のバックグラウンドで行われる．

4　まとめ

本稿ではデータ依存違反による squash 回数を軽減する手法，およびキャッシュミス率を軽減させる手法を提案した．squash 回数を軽減する手法では予測可能なデータ依存違反を回避し，スループットの向上と共有バスのトラフィック量の抑制が可能となった．また，冗長なキャッシュを用意することにより，直ちに新しいスレッドを開始し，そのスレッドのキャッシュミス率軽減が可能となった．