2H-7

2パス限定投機システムにおける投機的メモリアクセスの解析

関口 祐司 † 十鳥 弘泰 †† 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 †† 宇都宮大学大学院工学研究科情報システム科学専攻

1 はじめに

近年，マルチコアプロセッサが普及し，マルチスレッドを使用したプログラム高速化手法が注目されている．

我々は，プログラム中のループの実行頻度の高い2つの実行経路（以下，パス）に最適化を行い，ループ中のパスを対象に投機的にマルチスレッドで実行していく2パス限定投機方式を提案し，その実装をアーキテクチャ上で実施するため，2パス限定投機方式システム PALS を開発している.[1][2]

PALSの投機的メモリアクセスを適切に行うため，各スレッドユニットに Memory Buffer (MB) を設置し，PALSの並列実行時の投機データを格納する．しかし，スレッドがMBの容量を超えるストア命令を行った場合，データを格納することができず，プログラムを実行できない.（詳細は後述する）

そこで本稿では，その問題解決を目指したメモリアクセス機構を検討するために，PALS 上でプログラムを実行した際の投機的メモリアクセスの解析を行う．

2 2パス限定投機システム PALS

PALS ではプログラム内のループで実行頻度の多いパス上位二つを投機実行の対象にし，最適化された投機コードをあらかじめ作成する．そして，次に実行するパスを動的に予測し，予測パスに基づいて対応するパスの投機コードを実行する．図1に PALS の構成を示す．TU は計算を行うプロセッサであり，TMU は TU の制御やパス予測などを行う機構である．

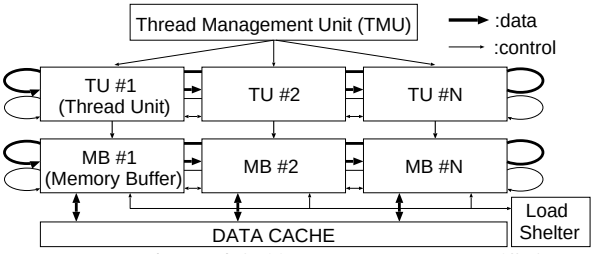

図 1: 2パス限定投機システム PALS の構成

2.1 PALS のメモリシステム

PALS においてパスの投機実行が失敗した場合，投機スレッドの実行を破棄し，メモリ上のデータを投機失敗前の実行状態に戻して実行を再開する．この投機失敗時の復帰作業はオーバヘッドとなる．また，投機実行中のメモリアクセスは逐次処理中のメモリアクセス処理の実行が異なり依存違反が発生する場合があるので，投機実行中のメモリアクセスを適切に行うメモリシステムが必要になる．

そこで PALS では，各 TU にローカルメモリとして Memory Buffer (MB) を備える．投機ミスした場合に MB のデータを消去するだけで，スレッドの実行を破棄することができる．MB のメモリアクセスのデータ整合性を保つメカニズムは ARB をもとにしている.[3]

PALS では MB にアドレスとデータ1組に制御ビットを付加したものを1つのエントリとして記憶する．図2にメモリ依存違反を検出した時の状態を示す．ロード命令実行時，MB は後続 TU の MB から読み込まれたアドレスを制御ビットと共に記憶される．そして，当該エントリに新たなストアが行われた場合にメモリ依存違反を検出し，後続スレッドを破棄する．ハードウェアで検出を行うため，オーバヘッドを最小限にすることができる．

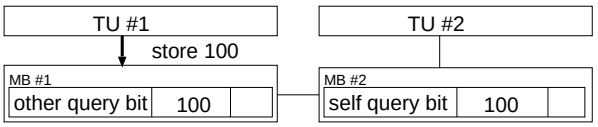

図 2: メモリ依存違反を検出した時

しかし，自 TU や後続の MB からのロード命令実行時にエントリを作成することにより，スレッドのストアするデータサイズが MB の容量以下でもエントリが MB に格納できない問題がある．先頭の MB でエントリを格納できない場合，先頭スレッドの実行ができずプログラムの実行が停止する．図3にエントリを格納できなかった場合の様子を示す．

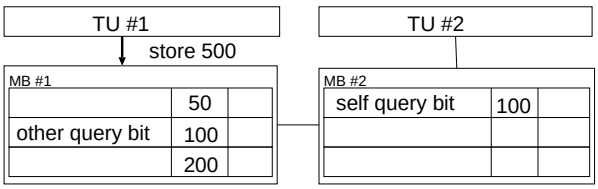

図 3: 先頭 TU の MB にエントリが格納できなかった場合

この問題を解決するためには，先頭のスレッドの実行を停止させないことが重要である．後続スレッドからのロード要求のエントリを退避するための記憶装置として Load Shelter (LS) を用意する．LS は先頭 TU の MB のみ使用することができる．LS で図3の状態になる確率を減らすことができるが，エントリを退避することでオーバーヘッドが大きくなる．

Analysis of Speculative Memory Access for Two-Path Limited Speculation System
†Yuji Sekiguchi
††Hiroyoshi Jutori, Kanemitsu Ootsu, Takashi Yokota, Takanobu Baba
Department of Infomation Science, Faculty of Engineering, Utsunomiya University (†)
Department of Information Systems Science, Graduate School of Engineering, Utsunomiya University (††)

ハードウェア資源量（MBの容量）を抑えるために，どの程度のエントリ数が必要になるかを調べる．そこで，PALSにおいてループをマルチスレッド実行する際に，実際に必要となる最大のエントリ数を算出する．プログラムを高速化するためには，実行頻度の高いループ（以下，ホットループ）を対象にすることが効果的である．そこで，本稿ではプログラム全体におけるループの実行割合のうち，90%を占めるループを対象に調査する．

3 スレッド当たりのメモリアクセスの解析

MBの容量をどの程度にすればよいのか検討するために，プログラムをPALS上でマルチスレッド実行した際に実際に使用されるエントリ数の調査を行う．

調査はSPECベンチマークプログラムを用いて行う．各ループの実行頻度を多いものから順に，実行頻度の合計が90%を超えるまで加算していく．この方法に従って164.gzipと256.bzipのホットループを集計した．実行頻度はプログラム全体におけるループの実行回数から算出した．164.gzipは実行頻度上位9種類のループ，256.bzipは上位4種類のループの実行頻度を合計すると約90%となる．このプログラムをPALSにおいて4スレッド並列実行することを想定して評価を行う．実行中のメモリアクセスで使用されるエントリ数をループイテレーション毎にカウントすることで行う．

3.1 評価環境

プログラムをPALS上でマルチスレッド実行時にメモリアクセスで使用するエントリ数の計算は，命令実行時で使用されるデータのアドレスから算出した．メモリアクセスでは，新しいアドレスであるならばアドレスをMBに記憶し，新しいエントリを作成する．ロード命令実行時にアドレスが当該MBにない場合，先行するMBにアドレスの値を送り，同じアドレスがなかったらそのアドレスを記憶する新しいエントリを作成する．この時，記憶されたアドレスのエントリの合計が並列実行中に格納したエントリ数である．

評価対象のループでMBに格納できる全メモリアクセスのエントリ数とストアデータのエントリ数の割合を図4に示す．図中のストアデータとは，ループ中のパスで，最もストア命令の実行数が多いパスのストア命令数である．このストアデータが記憶できないとプログラムの実行ができず停止してしまう．ストア命令のエントリは必ずMBに格納できるように，必要なエントリを見積りをしなければならない．

3.2 評価結果

PALSでは，格納できないエントリがあると並列実行の結果を保証できない．そのため，最大どの程度のエントリ数を見積もると並列実行できるかを164.gzipと256.bzip中のホットループを対象に算出した．

図4で，どの程度エントリを見積もれば全メモリアクセスのエントリを格納できるのかを算出した．

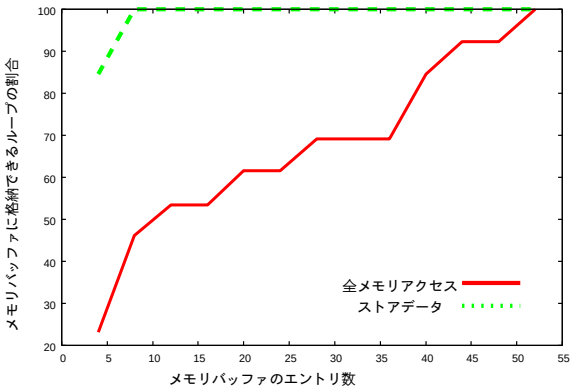

図4: メモリバッファに格納できるループの割合

この評価では，ストアデータを格納するためにはエントリが8必要であることが分かった．エントリ8である時，5割前後のループ中のマルチスレッド実行時での全メモリアクセスのエントリを格納することができる．なので，大部分のループのメモリアクセスは比較的少ないことが分かる．

MBのエントリ数を30と見積もると7割前後のループのマルチスレッド実行を保証できる．しかし，ホットループ実行中のストア命令の数は8である．大部分のエントリはLSに退避させるので，大部分のループでオーバーヘッドが大きくなってしまう．

4 おわりに

本論文ではホットループ中の実行時に格納するエントリ数を算出し，SPEC CINT2000の2つのプログラムに対して、MBのエントリ数とそのMBに格納することのできるホットループの割合を調べた．その結果，ストアデータのみ格納する場合は，8エントリ必要であることが分かった．一方，全メモリアクセスを格納するには52エントリ必要であることが分かった．

今後は，他のSPECベンチマークプログラムを対象として調査を行い，MBのエントリ数とストアデータを格納できない割合の関係を調査する．その結果をもとに,MBのエントリ数を超えたストアを行ってもプログラムの実行を保証するメモリシステムを検討する．

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 (C)20500047，同 (C)21500049，同 (C)21500050) および，宇都宮大学若手萌芽的研究プロジェクトの援助による．

参考文献

[1] 横田隆史ほか: “2パス限定投機方式の提案”, 情報処理学会論文誌:コンピューティングシステム, Vol.46, No.SIG 16(ACS 12), pp.1-13, 2005.

[2] 十鳥弘泰ほか: “2パス限定投機方式を実現するマルチコアプロセッサPALSの提案”, 信学技報, Vol.109, No.319(CPSY2009-46), pp.19-24, 2009.

[3] M Franklin ほか: “ARB:A Hardware Mechanism for Dynamic Memory System”, In Proceedings of the 22nd Annual International Symposium on Computer Architecture, June 22-24, pp.414-425, 1995.