PHTエントリの破壊的競合を検知する手法とその効果

西岡 拓生† 孟 林†† 小柳 滋†
†立命館大学情報理工学部 ††立命館大学理工学研究科

1 はじめに

分岐予測とは，分岐命令の条件成立，不成立による分岐方向と分岐先アドレスを予測し，投機的に命令をフェッチすることである．もし分岐方向を決定するまで命令フェッチを停止した場合，フェッチ幅の広いプロセッサの性能を大きく低下させる．しかし分岐予測を誤った場合，予測ミスペナルティによりプロセッサの性能を大きく低下させる．近年のプロセッサはパイプラインが深くなる傾向にあるので，予測精度が低ければ，ミスペナルティはさらに増大する．

本論文では，動的分岐方向予測の精度を向上させることを目的として，分岐予測器の予測表の破壊的競合に着目した．予測表で発生した破壊的競合を検出し，解消することを試みた．

以下ではまず2章で動的分岐予測の概要と問題点を述べる．3章で提案手法を説明し，4章で実験結果を示す．5章でまとめを述べ，課題を検討する．

2 動的分岐予測の概要

動的分岐予測器は分岐の過去の分岐傾向から将来を予測する．過去の分岐傾向を記録するのに，分岐方向の決定で遷移する2bit飽和カウンタがよく用いられる．これは，過去と将来の分岐傾向の間には相関がある，という事実を利用している．各分岐命令は，2bitカウンタの集まりである予測表（以下，PHT）を参照する．しかし全ての分岐命令に2bitカウンタを割り当てるのはハードウェアのコストの問題から不可能なので，複数の命令で共有することになる．

PHTの大きさだけでなく，PHTのインデクス付けの方法も予測精度に大きく関わっている．インデクス付けの方法は，命令アドレスや分岐履歴の組み合わせによって様々ある．Bimodal予測器は分岐命令アドレスの上位ビットを用いてインデクス付けている．2レベル適応型分岐予測器（例えばGAp，PAp，gshare）は，過去の分岐履歴をシフトレジスタに保持し，分岐方向のパターンの繰り返しと命令アドレスを組み合わせてインデクス付けている[1]．

分岐予測精度を低下させる原因のひとつにPHTの破壊的競合がある．複数の分岐命令がPHTの一つのエントリを共有することを競合と呼ぶ．破壊的競合とは，エントリの共有により予測を誤る現象である．分岐命令の多くは分岐方向が偏っていることが知られているので，分岐の偏る方向が異なる命令がPHTを共有すると，破壊的競合が頻発する．図1に予測器を使用した分岐命令の内，破壊的競合を起こした命令の割合を示す．Bimodalとは違い，GApやgshare(BHR:8bit)は競合が起きやすくなる．この原因は，分岐間の相関を予測に反映するために，一つの分岐命令が複数のPHTエントリを使用することにある．

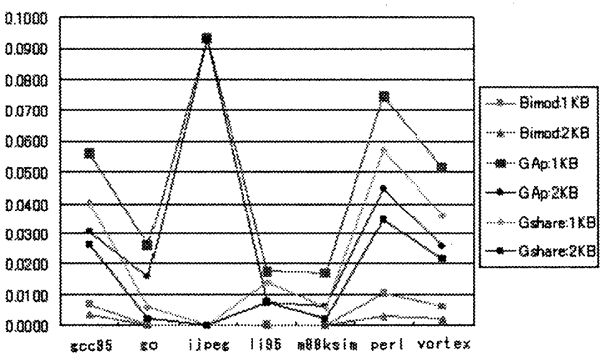

図1: 各予測器の破壊的競合率

3 提案手法

以上より，破壊的競合は「2bitカウンタの状態の往復」の形で現れると考えられる．そこで，カウンタの往復が発生しているエントリを使用する分岐命令に対して，通常用とはインデクス付けの異なる別種の予測器から予測値を得る機構を提案する．

3.1 破壊的競合検知機構

2bitカウンタの往復を検知するために，各エントリに5bitシフトレジスタ（以下SR）と1bitのDirection Bit(以下DB)を追加する（図2,3）．SRの初期値は0とし，DBの初期値はPHTの初期値が01の時0，10の時1とする．DBは，2bitカウンタが増加，もしくは減少するであろう方向を示している（つまり0なら減少，1なら増加する）．2bitカウンタがDBと同じ方向へ更新された時はSRを1bit左論理シ

A proposed of Detecting Destructive aliasing and its Effect
† Takuo Nishioka, Department of Information Science and Engineering, Ritsumeikan University
†† Lin Meng, Graduated School of Science and Engineering, Ritsumeikan University
† Shigeru Oyanagi

フトする．しかし，逆の方向へ更新された時は左論理シフトさせて 1 を加算し，DB を変更する．つまり SR は 2bit カウンタの更新の方向転換を記録している．SR の各ビットを加算した結果が 3 以上となった時に，破壊的競合が発生していると判断する．

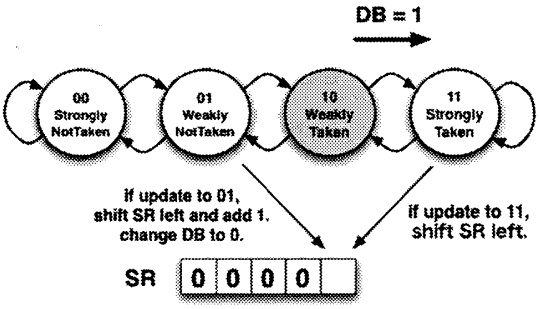

図 2: 破壊的競合の検知

3.2 監視リスト

競合を検知した時，緊急用の予測器から予測値を得る．そのために監視リストに通常用予測器の PHT エントリのインデックスを登録する (図 3)．しかし，監視できるエントリ数には限りがある．登録する時にリストが一杯だった場合，エントリを 1 つ解放する必要がある．解放するのは，最も使用されていないエントリとする (LRU)．また，SR が初期値の 0 に復帰した場合は，破壊的競合が起きていない事を意味するので，積極的に監視リストからエントリを解放するようにした．そのために，緊急用の予測器を使用している間も通常用の予測器の更新を行う．

その他，通常用と緊急用の予測器の選択を高速化するために PHT 選択ビットを用意した．監視リストに登録されているエントリのビットは 1 とする．

4 実験

simplescalar-3.0 を改造して提案手法を実現した．通常の予測器には破壊的競合率の高い GAp 予測器を，緊急用の予測器には破壊的競合率の低い Bimodal 予測器を選択した．提案手法と比較対象予測器の BHR を全て 8bit，監視テーブルエントリ数の上限値を 32，緊急用 PHT エントリ数を 128 に統一した．シミュレータは sim-outorder を，ベンチマークは SPECint95 中の 7 本を用いた．

図 4 に実験結果を示す．gcc95 で提案手法の GAp2048(PHT エントリ数) は，PAp8192 より予測率を 0.38%上昇させた．ijpeg で提案手法の GAp1024，2048 は，GAp と PAp4096，8192 より

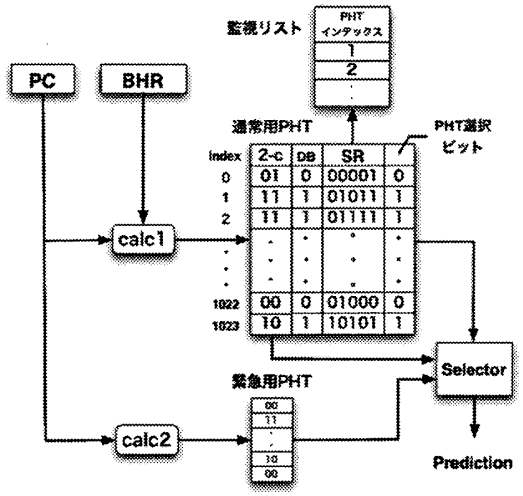

図 3: 提案手法全体図

1.53〜1.56%上昇させた．その他はあまり効果が得られていないか，他予測器より下回った．原因は，リストの検索速度の考慮から監視リストのエントリ数に上限があるので，リストへの追加と削除が頻繁に起こっているためである．また，DB と SR を追加したために，同じ PHT エントリ数の GAp と比べて提案予測器の容量が 4 倍以上となる．ほぼ同容量として比較したため，予測率の低下を避けられなかった．

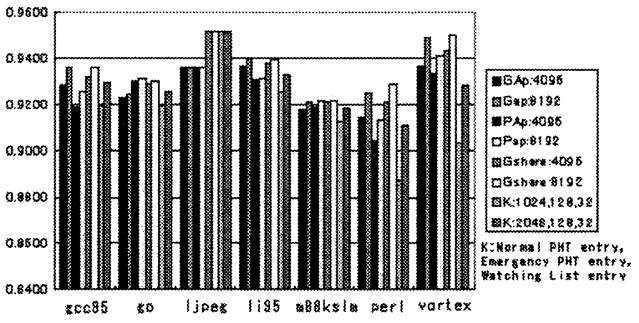

図 4: 実験結果

5 まとめ

提案手法は監視リストと全体容量の点で問題があることがわかった．課題は監視リストの登録，削除方法と，競合検知機構の容量の改善である．また，監視リストのエントリ数増加を検索速度の点から検討したい．

参考文献

[1] 安藤秀樹：命令レベル並列処理，コロナ社，2005
[2]Mike Johnson，村上和彰：スーパースカラ・プロセッサ，日系 BP 出版センター，1994