

## 2パス限定投機システムにおけるパス予測器の検討

金海 和宏<sup>†</sup> 大津 金光<sup>†</sup> 横田 隆史<sup>†</sup> 馬場 敬信<sup>†</sup><sup>†</sup>宇都宮大学工学部情報工学科

## 1 はじめに

プログラムのループ中には、条件分岐の結果の組み合わせにより様々な実行経路（以下、パス）が存在する。ループにおいて次に実行されるパスを予測し、投機的にマルチスレッド実行していくことで実行時間の短縮が望める。

我々が提案した2パス限定投機方式 [1] では、高々2本のパスの実行頻度がループ中において支配的であることを明らかにした。上位2本の各パス（以下、実行頻度の高い順に#1パス、#2パス）を予測対象として、投機実行に特化した#1パスと#2パスのコードを作成する。この2つのコードを投機的にマルチスレッド実行することでプログラム実行の高速化を行う。

我々は当方式をハードウェア上で実現するための2パス限定投機システム PALS を提案している [2]。当システム中においてパスの予測を行う機構を実装しており、システムの性能は予測成功率により大きく左右される。しかし、現在搭載されているパス予測器の妥当性については十分な検討が行われていない。本稿では、幅広い分岐予測機構をもとに、予測成功率の向上を目指してパス予測器の検討を行う。

## 2 パス予測機構

PALS では、パスの投機を行う前に、#1パスかそれ以外のパスかを予測する予測器が実装されている。

現状のパス予測器では2レベル分岐予測機構の GAg を採用している。2レベル分岐予測器の機構にはこの他にも GAp, PAg, PAp, Gshare[3] が存在し、これらの機構は今日の分岐予測器において広く活用されている。これらの機構を実際にパス予測器に実装して評価を行い、パス予測器に関して各機構がどれほどの妥当性を持つか検討する。

パス予測器はパス履歴レジスタ（以下、履歴レジスタ）とカウンタテーブルから成り、履歴レジスタをインデックスとしてカウンタテーブルを参照し、格納内容に基づいて次のパスの予測を行う。

履歴レジスタは、#1パスを1、それ以外のパスを0として1ビット上位シフトの後に最下位ビットへの数値の格納を繰り返すことでパスの履歴を保存する。nビット分の履歴をカウンタテーブルへのインデックスとして使用する。

予測ミスが発生した場合、発生した段階まで履歴レジスタを巻き戻し、正しい予測結果を格納した後に再

び予測を開始する。このため、履歴のロールバックを行うために、既の実行が確定したnビットの記憶用レジスタを履歴レジスタとは別に保持している。予測が失敗した際に履歴レジスタに記憶用レジスタの内容をロードし、正しいパス情報のビットを履歴レジスタの最下位ビットに格納した上で予測を再開する。実際のパス予測に用いる履歴レジスタはnビット以上の領域を用意しており、カウンタテーブルを参照するために最新の下位nビットを有効としている。この履歴レジスタ長を変更することによる予測成功率の変化も評価の対象となる。

カウンタテーブルの各テーブルは2ビットの飽和カウンタを持つ。各カウンタの閾値および初期値を  $(10)_2$  として、閾値以上である場合には#1パスへの予測を行い、閾値未満である場合はそれ以外のパスへの予測を行う。予測後に正しいパスが#1であればカウンタがインクリメントされ、それ以外へのパスであればデクリメントされる。従来の2レベル分岐予測器では上記に示した通り2ビットの飽和カウンタを用いるが、このカウンタのビット数を変化させて評価を行う。

図1にパス予測器 (GAg) の構成を示す。

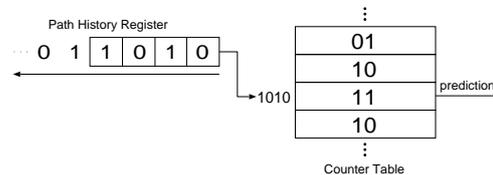


図 1: 2レベルパス予測器 (GAg)

## 3 評価環境

使用するループは SPEC CINT2000 の 164.zip 内に出現する3つのループとし、少なくとも2本以上のパスが出現すること、ループ中で別の関数を呼び出さないことを選定条件としている。

データセットは test を用い、評価対象とした3つのループのイテレーション（繰り返し）数と#1パスの出現頻度をまとめたものを表1に示す。

表 1: 各ループの実行状況

	ループ回数	#1パスの出現頻度
1	16495105	50.050 %
2	10321605	62.922 %
3	10321605	61.962 %

ループアドレスの下1桁は必ず0か8であり、下位3ビットは000で固定される。このため、Gshareにおいては、下位4ビットから7ビットまでの数値とレジスタのXORを取り、インデックスとしている。

Consideration of Path Predictor in Two-Path Limited Speculation System

<sup>†</sup>Kazuhiro Kinkai, Kanemitsu Ootsu, Takashi Yokota and Takanobu Baba

Department of Information Science, Faculty of Engineering, Utsunomiya University (†)

本来プログラム中で予測ミスが起こった場合でも継続的に投機実行されるため、ただちに予測ミスを判定することはできない。しかし、本稿の評価では1イテレーションごとに予測が正しいかどうかを判断し、結果を得ている。

4 評価結果

履歴レジスタ長を変化させた評価結果を図2に、カウンタテーブルにおける各テーブルの飽和カウンタのビット数を変化させた評価結果を図3に示す。

図2はカウンタテーブルのカウンタを2ビットの飽和カウンタとし、履歴レジスタ長を0ビットから28ビットまで変化させた評価結果となる。履歴レジスタが0ビットのとき、カウンタテーブルは1つだけテーブルを持つ。このとき、パス予測器は履歴レジスタの内容にかかわらずにただ1つのテーブルのみを参照して予測を行う。

図3では履歴レジスタ長を28ビットとし、飽和カウンタのビット数を1ビットから8ビットまで変化させている。

図中の数値はプログラム終了時の全てのループを通じた成功率となっており、小数点以下3桁目を四捨五入している。

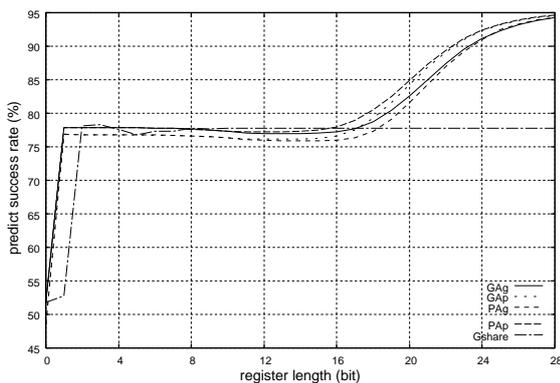


図 2: 予測成功率 (履歴レジスタ長 変化)

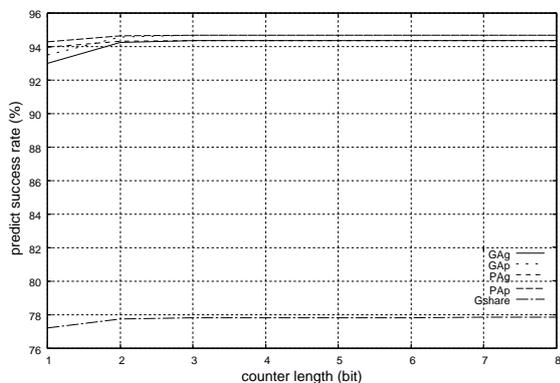


図 3: 予測成功率 (飽和カウンタ長 変化)

履歴レジスタ長が16ビットを超えるとPApが最も高い予測成功率を持つ。特に、ループアドレスごとにカウンタテーブルを用意するGAp, PApが他のパス予測器に対して高い評価結果を残した。

履歴レジスタのビット長の変化に伴って予測成功率も変化するが、Gshareとそれ以外のパス予測器には評価結果の推移に違いが生じる。Gshareは約78%の予測成功率で飽和するが、その他のパス予測器は、履歴レジスタのビット長がGshareよりも大きいときに予測成功率の変曲点が訪れており、飽和した際の予測成功率もGshareに比べて大きい。

飽和カウンタを0ビットとすると、全てのイテレーションに関して#1パスでの予測を行う。このため、どのパス予測器も予測成功率はループに発生する#1パスの頻度とほぼ同じとなる。各パス予測器は飽和カウンタが3ビットのときに予測成功率が飽和する。

5 おわりに

本稿では、2パス限定投機システムにおけるパス予測器の予測成功率向上を目指し、2レベル分岐予測器の機構をもとにしたパス予測器の評価を行った。履歴レジスタとカウンタテーブル内の飽和カウンタのビット長を変えた評価も同時に行っている。

本研究ではPApが最も予測成功率が高かった。カウンタテーブルの飽和カウンタ長を変化させることでGApもPApと同程度の予測成功率をもつ。またGshare以外のパス予測器は、履歴レジスタのビット長を増やすことで予測成功率を向上できる。しかし、PApは2レベルパス予測器の機構の中でハードウェア量が最も多く必要であり、履歴レジスタのビット長を増やすと必要なハードウェア量もまた増加する。このため、性能とハードウェア量の検討が必要となる。

今後は、本稿で用いたもの以外のSPEC CINT2000ベンチマークプログラムによる評価を行い、より詳細なデータを取った上で最も予測成功率の高いパス予測器を検討していく予定である。

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

参考文献

- [1] 横田 隆史, 斎藤 盛幸, 大津 金光, 古川 文人, 馬場 敬信: “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, 大津 金光, 横田 隆史, 馬場 敬信: “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] Yu Yeh and Yale N. Patt: “A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History”, The 20th Annual International Symposium on Computer Architecture, pp.257-266, 1993.
- [4] Scott McFarling: “Combining Branch Predictors”, Digital Equipment Corporation Western Research Laboratories, Technical Note TN-36 1993.