

遷移確率に基づく分岐予測手法の提案

梅沢 晃 山名 早人

早稲田大学理工学部

プロセッサの処理速度を向上させるために、今日のプロセッサではパイプラインを利用して命令を並列に実行させる技術が使われている。しかし、条件分岐が存在すると次の命令を確定することができない。条件分岐の分岐成立の有無を予測し、分岐方向が確定する前に実行を開始する技術を分岐予測と言い、予測精度向上のため様々な研究が行われている。

本稿では、従来の動的な分岐予測手法では予測が困難と分類される分岐をターゲットとし、分岐の同一方向への連続分岐回数の偏りに基づいた分岐予測手法を提案する。本予測手法を SPECint95 に適用したところ、特定の分岐において Bimodal Predictor と比べ、予測ミス率が最大 33.49% 減少した。

A Proposal of the Branch Prediction Technique based on the Transition Rate

Akira Umezawa Hayato Yamana
School of Science and Engineering, Waseda Univ.

In order to raise the processing speed of a processor, in today's processor, the technique of pipelining is adapted to extract the instruction level parallelism. However, a pipeline stall occurs when a conditional branch exists. Various researches have been done, in order to raise the accuracy of prediction.

In this paper, we propose a new branch prediction technique based on the transition rate, which is specifically the number of succession branch times for the same direction. The proposed scheme targets the branches that are classified into difficult prediction branch. We applied the proposed scheme to some specific branches in SPECint95. As a result, we confirmed that the miss prediction rate decreased a maximum of 33.49% compared with that of Bimodal Predictor.

1 はじめに

近年のプロセッサは深いパイプラインを持つため、分岐予測精度がプロセッサの性能に大きな影響を与える。分岐予測には、コンパイル時に予測方向を決定する静的な分岐予測 [1][2]、実行中に予測方向を決定する動的な分岐予測がある。現在は複雑な動的な分岐予測方式が提案され、高い予測精度を示している。しかし、年々プロセッサのパイプライン段数は深くなる傾向があり、そのようなプロセッサでは予測ミスからの復帰処理にかかるコストが高くなり、より高精度な予測が必要とされている。

動的な分岐予測は、

- (1) 精度を上げるための分岐予測方式の研究 [3]。
- (2) 有限であるハードウェア量を用いた時の、高精度な予測方法に関する研究 [4]。

の2種類の研究に分類される。(1)は2レベル適応型分岐予測機構のように、グローバルまたはアドレスごとの分岐履歴を分岐予測に利用することにより、予測の精度自体を上げようというものである。また、(2)は分岐同士のエイリアシングにより予測精度が制限される問題に関して、動的や静的なプロファイル情報を用いて、ハードウェアを効率的に利用する方法等の研究である。

本稿では、これらの内(1)を研究対象とした。以下では、まず、2節で従来の分岐予測手法では予測困難な分岐の特徴について述べる。2節の結果に基

づき、3節で予測困難な分岐に対する予測手法を提案し、4節に評価結果を示す。

2 予測困難な分岐の解析

プログラム全体としての分岐予測精度を向上させるためには、全体の予測精度への影響が大きく、既存の方法では予測精度の向上が困難である分岐の予測精度を向上させることが重要である。以下では、そのような予測困難と判断される分岐の特徴解析を行う。

解析対象は、SPECint95 から m88ksim(マイクロプロセッサのエミュレーション)、gcc(C言語コンパイラ)、compress(テキスト・ファイルの圧縮と解凍)、li(Lispのインタプリタ)、jpeg(jpeg方式による画像の符号化、複合化)、perl(Perl言語の文字列の並べ替え)とする。入力データセットは test、train とする (Perl は test のみ)。

これらのプログラムの各分岐に対して、2bit 飽和カウンタ (Bimodal Predictor)[6]、及び2レベル適応型分岐予測機構である PAp(Per-address Adaptive Branch Prediction using per-address pattern history tables)、GAp(Global Adaptive Branch Prediction using per-address pattern history tables)[7]を適用した際の予測ミス率を SimpleScalar Tool Set Version 2.0[8]で計測する (厳密には SimpleScalar の PAp、GAp は、分岐アドレスの下位ビットによる

インデックス付けを行うので、Yaleらが[8]で分類した PAp、GApとは異なり、SAs(Per-Set Adaptive Branch Prediction using per-set address pattern history tables)、GAs(Global Adaptive Branch Prediction using per-set pattern history tables)と等しい。)。なお、分岐同士のエイリアシングが発生すると、真に予測し難い分岐を特定することができないので、分岐予測テーブルサイズを十分に大きく用意した(表1)。

表 1: 分岐予測方式ごとのテーブルサイズ

予測方式	一層目のエントリ数	履歴長	飽和カウンタの数
Bimodal			2^{30}
PAp	2^{15}	15	2^{30}
GAp	1	15	2^{30}

2.1 予測ミス率への影響が大きい分岐

図1はそれぞれのプログラムで、全分岐予測ミスに占める割合が大きい順に、上位10%の静的分岐を対象に、(1)分岐全体に占める割合(動的分岐数)、(2)予測ミス数に占める割合を示す。それぞれのプログラムでの分岐予測方式は、表1の3方式の中で最も精度の高い予測方式を用いた。図1より、多くのプログラムでは僅かな割合の分岐が、分岐全体の実行回数の割合、及び分岐全体のミス数の割合の大部分を占めることがわかる。これらの分岐の精度を改善することができれば、プログラムを通しての予測精度にも大きな影響を与えることになると考えられる。

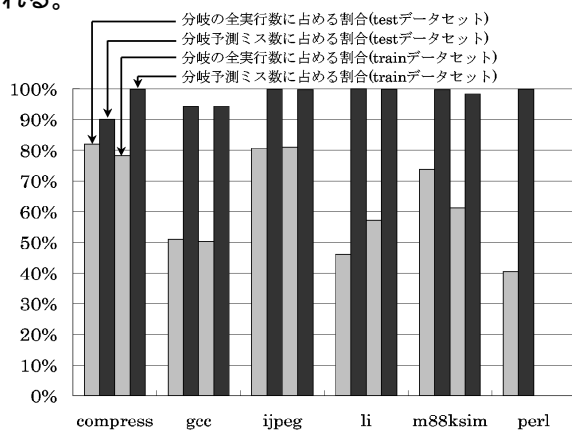


図 1: 全ミス数に占める割合が上位10%の静的な分岐が占める動的な分岐実行数・予測ミス数の割合

2.2 分岐履歴による分類

動的分岐予測では、分岐同士のエイリアシングにより予測精度の低下が生じる。分岐同士のエイリアシングを避けるためには、分岐に対して予測ハードウェアを効率的に割り当てる必要があり、そのために分岐履歴で分岐を分類する方法が提案されている[4][5]。例えば[4]では分岐履歴により、容易に予測が可能と分類された分岐は静的に予測を行い、予測が困難と分類された分岐には複雑な動的予測機構を割り当てる。

以下では、分岐の分類を行うために一般に用い

られている、Taken Rate、Transition Rate[5]により分岐を分類する。図2~図4は、分岐予測ミスに占める割合が上位30位までの分岐を Taken Rate、Transition Rate ごとに分類し、各分岐予測方式を使用した時の精度を示したものである。Taken Rateは分岐の Taken 方向への偏りを示し、Taken 方向への分岐回数をプログラムを通しての分岐回数で割った値であり、Transition Rateは分岐方向が変わった回数をプログラムを通しての分岐回数で割った値である。

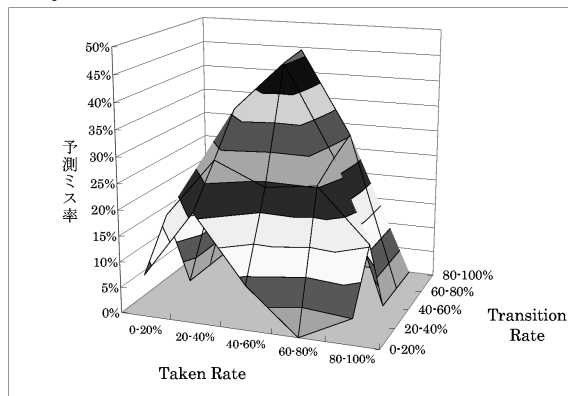


図 2: Bimodal Predictor での予測ミス率

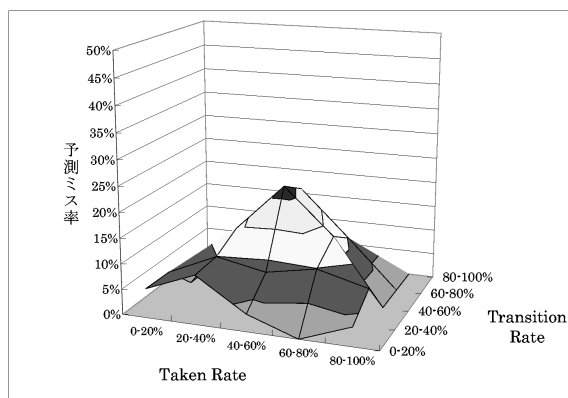


図 3: PAp での予測ミス率

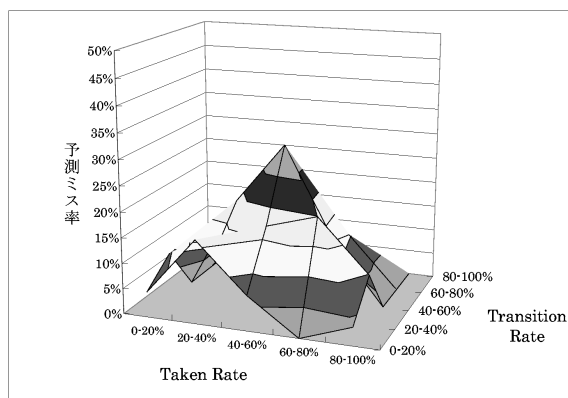


図 4: GAp での予測ミス率

図2~図4からわかるように、どの予測方式でも、Taken Rate、Transition Rateの両方が40%~60%の分岐で、予測ミス率が高いという結果になっている。これは、分岐方向の偏りが少なく、頻繁に分岐方向の変更が起こる分岐では、予測ミスが多く

表 2: 各分岐における同一方向への連続分岐回数の割合の積算値 (%)

	1回まで	2回まで	3回まで	4回まで	5回まで	6回まで	7回まで	8回まで	9回まで	10回まで
compress	46.66	73.43	85.32	92.18	98.33	99.21	100.00			
gcc	47.40	72.89	84.42	90.49	93.82	95.79	97.04	98.17	98.86	99.32
ijpeg	50.29	74.59	87.05	93.25	96.35	98.13	99.06	99.45	99.74	99.84
li	55.44	90.20	93.72	93.99	99.72	99.72	99.72	99.72	100.00	
m88ksim	43.60	89.25	95.19	98.15	99.17	99.79	99.79	100.00		
perl	49.93	74.86	87.44	93.81	96.90	98.47	99.24	99.62	99.81	99.91

なるためと考えられる。

2.3 予測困難である分岐履歴の解析

予測が困難である分岐の履歴の特徴を調べるため、それぞれのプログラムの Taken Rate、Transition Rate が共に 40% ~ 60% 分岐の中から、最も予測精度が低い分岐を各プログラム 1 つずつ選択し、その挙動を調べる。予測精度は、従来提案されている各種の分岐予測手法がプリミティブとして備える Bimodal Predictor を基準とした。

表 2 は、分岐の同一方向への連続分岐回数の偏りを表し、それぞれの回数まで連続して分岐する割合の積算値を示す。表 2 より、どのプログラムにおいても同一方向への分岐が 4 回連続した後、さらに同じ方向へ分岐する確率が 10% 以下であることがわかる。また、表 2 の li を例に取ると、1 回までの実行で分岐方向が変更する分岐と比較して、2 回までの実行で分岐方向が変更する分岐の割合が約 35% 増加していることがわかる。このような分岐では、同一方向への 2 回連続分岐後は、次の分岐時に逆方向へ予測すると、高い予測精度が実現されると考えられる。

3 同一方向への連続分岐回数に基づく分岐予測手法の提案

本節では、Bimodal Predictor に「一定回数連続して同じ方向に分岐した時は、次の実行時に逆方向に予測する」という法則を適用する分岐予測手法の提案を行う。

3.1 本手法の分岐への適用方法

本稿での「一定回数連続して同じ方向に分岐」とは、ある分岐が複数回実行される時、同一方向へ連続して分岐した回数である。例えば、これから同じ方向に連続して 10 回分岐する時に、3 回連続して同じ方向に分岐したら逆方向に予測するという法則を Bimodal Predictor に適用するとする。最初の 2 回の分岐までは分岐結果で 2bit カウンタを更新していくが、3 回目の分岐終了時には、2bit カウンタを分岐結果で更新せずに、4 回目の分岐時に逆方向への分岐を予測するように更新する。4 回目以降は、その後 3 回連続して同じ方向に分岐しても、逆方向への予測は行わずに、実際に分岐結果で 2bit カウンタを更新していく。また、逆方向に予測する方法は、一定回数連続して同じ方向に分岐した後に、2bit カウンタの状態を 1 つ前の分岐とは逆方向に「Strong Biased」に設定する方法と、「Weak Biased」に設定する 2 種類の方法が考えられる (図 5)。

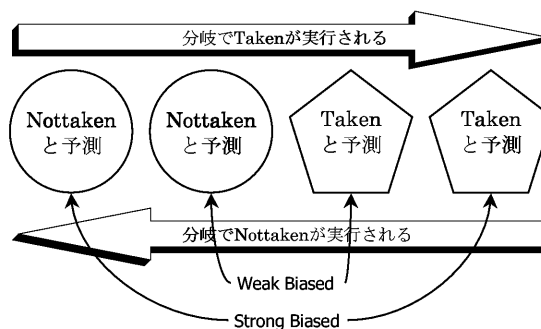


図 5: 2bit カウンタ

3.2 複数の連続分岐回数に対する予測の適用方法

表 2 の値から、分岐によっては、同一方向への連続分岐回数の割合が、一つ前の値と比較して大きく増加している箇所が複数あることがわかる。つまり、「3 回同一方向へ連続分岐後に逆方向へ予測し、予測が失敗すれば、さらに 5 回同一方向へ連続分岐後に逆方向へ予測する」のように、複数の連続分岐回数において本手法を適用することが効果的であると考えられる。

しかし、本手法を適用する同一方向への連続分岐回数を複数にした場合、過去の予測結果が未来の予測に悪影響を与えることが予想される。そのため、次の 2 種類の順で、複数の連続分岐回数に対して本手法を適用することを考える。

- (a) 本手法を各分岐に対して適用時に、予測ミス率が Bimodal Predictor と比較して低下する「同一方向への連続分岐回数」を調べる。その結果から、予測ミス率の低下が大きい順に、複数の同一方向への連続分岐回数に対して同時に適用する方法。
- (b) 一定回数同じ方向へ連続して分岐した時に、次の分岐方向を逆に予測したときの条件付き予測ミス率を分岐履歴から求める。その結果から、予測ミス率の低い順に、複数の同一方向への連続分岐回数に対して同時に適用する方法。

(a) の方法を図 6 を用いて説明する。図 6 は表 2 の compress に対して、3.1 で述べた手法を、同一方向へ n 回連続して分岐後に適用した時の、予測ミス率の推移を示している。(a) の方法では、従来の Bimodal Predictor と比較して、予測ミス率が低下している同一方向への連続分岐回数に基づく予測を、予測ミス率の低下が大きい連続分岐回数から順番に適用する。図 6 の例だと、Strong Biased に設定する方法では、(a)、(b)、(c) の順番に適用する。

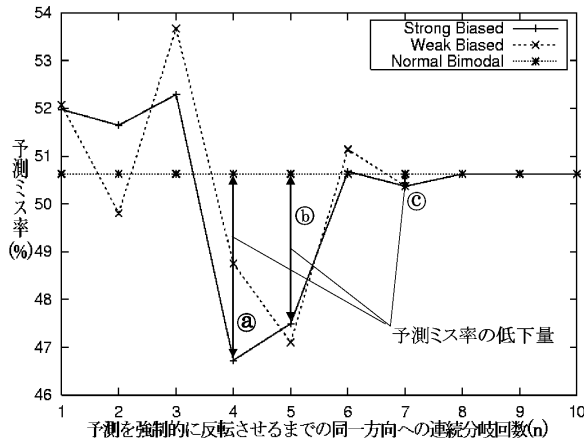


図 6: 予測ミス率の推移

(b)の方法を図7を用いて説明する。図7は表2の compress の値をグラフにしたものである。図7の X 軸の値が 2 を例にとると、①がそこまで到達する同一方向への連続分岐回数の割合(すなわち、2 回以上の連続分岐回数の割合)で、②が③より割合が大きいときは、2 回連続して同じ方向に分岐した後は予測方向を逆にした方が精度が上がることを示している。2 回連続して同じ方向に分岐した後に予測方向を逆にした場合の予測ミス率は式 (1) で求められる。

$$\text{予測ミス率 (\%)} = \frac{\text{③}}{\text{①}} \times 100 \quad (1)$$

(b)の方法では、表2の値から、一定回数同じ方向へ連続して分岐した時に、次の分岐方向を逆に予測したときの条件付き予測ミス率を式(1)で求める。式(1)で求めた条件付き予測ミス率が50%以下の同一方向への連続分岐後の予測のみ、予測ミス率の低い順に適用する。

表3で本節で提案した分岐予測手法をまとめる。

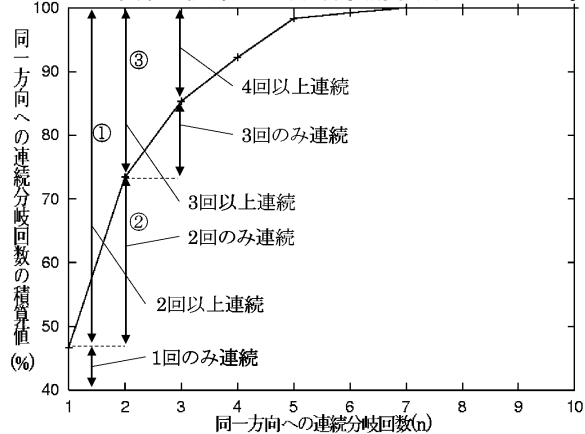


図 7: 同一方向への連続分岐回数の割合の積算値

4 提案手法の評価

本節では分岐の同一方向への連続分岐回数に基づく予測の評価を行う。

表 3: 提案手法の説明

提案手法	説明
3.1	一定回数の同一方向への連続分岐後に、Bimodal Predictor の予測方向を強制的に反転させる手法。
3.2(a)	3.1 の手法の結果から、予測ミス率が低下している連続分岐回数を求め、低下率が大きい順に複数の連続分岐回数に対して同時に適用する方法。
3.2(b)	分岐履歴から、一定回数同じ方向へ連続して分岐した時に、次の分岐方向を逆に予測したときの条件付き予測ミス率を求め、予測ミス率が低い順に、複数の同一方向への連続分岐回数に対して同時に適用する方法。

4.1 同一方向への連続分岐回数が n を超えた時に予測を反転させる場合

図8~図13は、Bimodal Predictor に3.1の手法を適用した時の、予測ミス率の推移を示している。n は予測方向を反転する同一方向への連続分岐回数を示す。

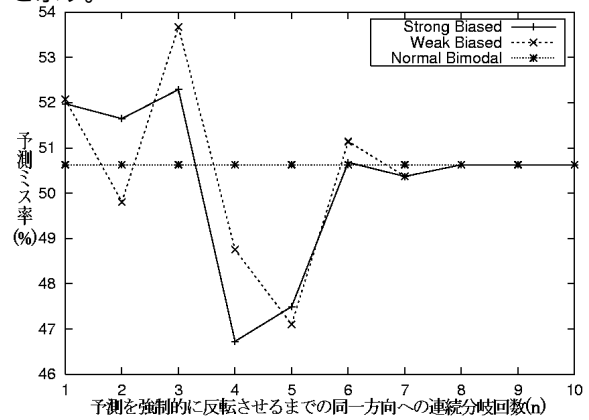


図 8: 予測ミス率の推移 (compress)

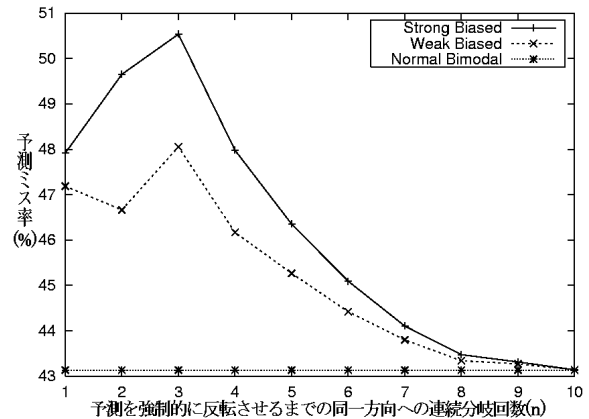


図 9: 予測ミス率の推移 (gcc)

図8~図13から、予測精度が改善する分岐の、最も有効な同一方向への連続分岐回数は4回以内が多く、値が大きくなるにつれ従来のBimodal Predictorの値に収束していくことがわかる。これは表2からわかるように、全ての分岐で、同一方向への連続分岐回数が4回以内で90%に到達し、それ以上の同一方向への連続分岐回数を予測に利用しても効果が低いからである。Strong Biased と Weak Biased の推移は、ほぼ同じ傾向を示している。

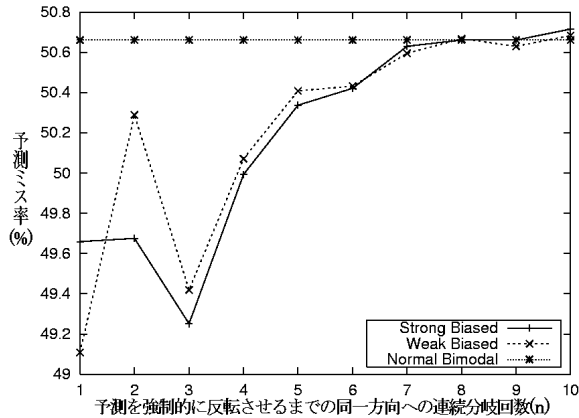


図 10: 予測ミス率の推移 (ijpeg)

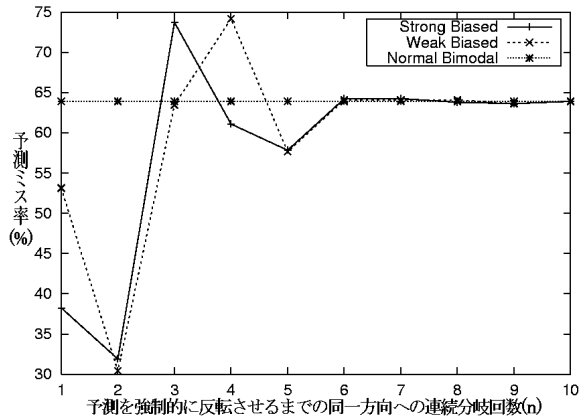


図 11: 予測ミス率の推移 (li)

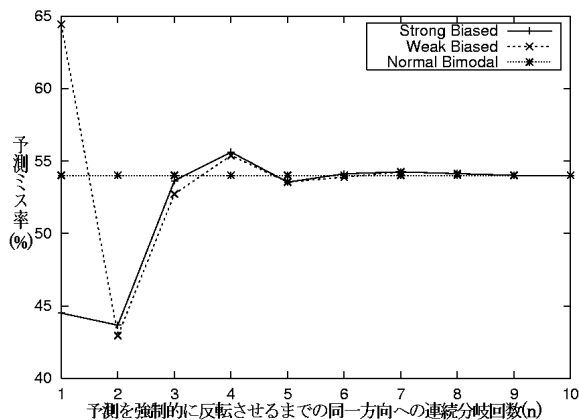


図 12: 予測ミス率の推移 (m88ksim)

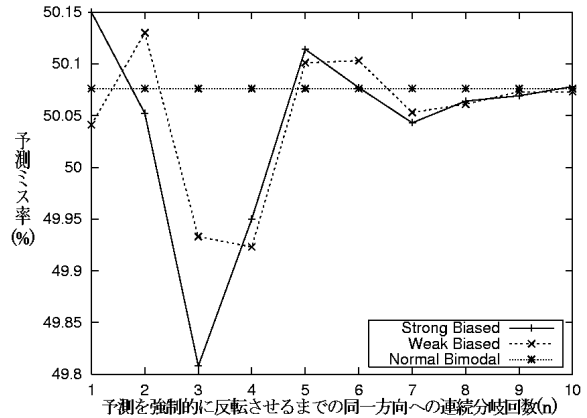


図 13: 予測ミス率の推移 (perl)

4.2 同一方向への連続分岐回数が n_1, n_2, \dots, n_q を超える各時点で予測を反転させる場合

次に同一方向への連続分岐回数に基づく予測を、3.2で示した方法で、複数同時に適用した時の予測ミス率の推移を、表4、表5に示す。表4は3.2(a)の手法、表5は3.2(b)の手法を適用した結果である。表の「 n 位まで」は、3.2(a)、3.2(b)の各手法により選択された上位 n 位までの連続分岐回数に、3.1の手法を適用したことを示す。また、表4の「1位まで」の予測ミス率は、標準の Bimodal Predictorと比較して最も精度が改善している3.1の手法による予測ミス率と等しい。各分岐における逆方向への予測方式ごとの最も精度の高い予測を太字で示した。

表4から、3.2(a)の手法は3.1の手法に比較し高い効果を示す分岐が存在することがわかる。しかし、liのように複数同時に適用することにより、予測精度が大きく低下する分岐も存在する。

表5は、表4で示した予測精度の推移と違い、複数の「連続分岐回数」に対して3.1の手法を適用するほど予測精度が向上している。また、複数の連続分岐回数に適用したことによって生じる、過去の予測結果による未来の予測への悪影響も、表4と比べて少ない。

表6に、表4、表5から得られた最も精度の高い予測と、従来の Bimodal Predictor を使用した時の予測精度をまとめ、それぞれの分岐に対して最も精度の高い予測を太字で示した。表6より、Bimodal Predictor に分岐の同一方向への連続分岐回数に基づく予測を適用することにより、6個の分岐のうち5個の分岐で予測精度が向上している。適用の順番では、3.2(a)の手法では少しの連続分岐回数に対して3.1の手法を適用するだけで高い効果を示すが、過去の予測結果による未来の予測への悪影響が3.2(b)に比べて大きい。3.2(b)の手法では、3.2(a)並みの予測精度を出すには、多くの連続分岐回数に対して3.1の手法を適用する必要がある。一方、全ての分岐で Weak Biased による予測精度が、Strong Biased による予測精度と同等か上回っており、3.1の手法を、複数の連続分岐回数に対して同時に適用する時には Weak Biased による予測方法を使った方が有効であると考えられる。

表 4: 3.2(a) 適用時の予測ミス率の推移 (%)

	方式	Bimodal	1位まで (3.1)	2位まで	3位まで	4位まで	5位まで	6位まで	7位まで	8位まで
compress	Strong	50.63	46.73	47.12	46.65					
compress	Weak	50.63	47.10	46.73	45.90	45.67				
gcc	Strong	43.13								
gcc	Weak	43.13								
jpeg	Strong	50.66	49.25	50.17	50.17	50.44	50.49	50.49	50.58	
jpeg	Weak	50.66	49.11	49.35	49.62	50.17	50.44	50.49	50.49	50.47
li	Strong	63.90	31.90	39.89	36.81	40.23	40.06	40.23		
li	Weak	63.90	30.41	38.24	35.00	36.65	36.48			
m88ksim	Strong	54.01	43.67	43.85	43.51	42.88				
m88ksim	Weak	54.01	42.94	43.67	43.55	43.43				
perl	Strong	50.08	49.81	49.83	49.80	49.90	49.90	49.90		
perl	Weak	50.08	49.93	49.81	49.94	49.94	49.93	49.93	49.93	

表 5: 3.2(b) 適用時の予測ミス率の推移 (%)

	方式	Bimodal	1位まで	2位まで	3位まで	4位まで	5位まで	6位まで
compress	Strong	50.63	50.63	50.37	47.03	47.03	47.89	
compress	Weak	50.63	50.63	50.37	46.84	47.03	46.05	
gcc	Strong	43.13						
gcc	Weak	43.13						
jpeg	Strong	50.66	50.66	49.67				
jpeg	Weak	50.66	50.63	49.09				
li	Strong	63.90	63.62	57.56	35.36	36.65		
li	Weak	63.90	63.62	57.39	33.71	34.83		
m88ksim	Strong	54.01	54.12	43.78	43.65	44.11	44.11	44.11
m88ksim	Weak	54.01	54.12	43.05	42.81	43.96	43.87	43.99
perl	Strong	50.08	50.07	49.94	49.93	49.93	49.93	49.81
perl	Weak	50.08	50.07	49.92	49.95	49.94	49.92	49.80

表 6: Bimodal Predictor、Bimodal Predictor + 提案手法による予測ミス率 (%) の比較

	方式	Bimodal	3.2(a)	3.2(b)
compress	Strong	50.63	46.65	47.03
compress	Weak	50.63	45.67	46.05
gcc	Strong	43.13		
gcc	Weak	43.13		
jpeg	Strong	50.66	49.25	49.67
jpeg	Weak	50.66	49.11	49.09
li	Strong	63.90	31.90	35.36
li	Weak	63.90	30.41	33.71
m88ksim	Strong	54.01	42.88	43.65
m88ksim	Weak	54.01	42.94	42.81
perl	Strong	50.08	49.80	49.81
perl	Weak	50.08	49.81	49.80

5 おわりに

本稿では、Taken Rate、Transition Rate により予測が困難と分類される分岐に対して、同一方向への連続分岐回数の偏りに基づいた予測の有効性について考察を行った。

同一方向への連続分岐回数の偏りから、予測困難な分岐の分岐方向の変化を予測する方法は、Bimodal Predictor との併用時には効果的であることがわかった。

この方法は Bimodal Predictor に限らず、GAP、PAP 等にも適用できると考えられる。履歴を基にするそれらの予測方法は、履歴により適切に結び付けられた 2bit カウンタでは高い予測精度を示す。しかし、特定の履歴の基では全く予測精度を上げることができない分岐も存在すると考えられ、そのような履歴が分岐全体としての予測精度を低下させていると考えられる。

本稿で提案した方法は、そのような履歴によって結び付けられた 2bit カウンタに対して、単純な方法で予測精度を上げることができることを示している。今後は、同一方向への連続分岐回数から、予測困難な分岐の分岐方向の変化を予測する方法が、実際に Bimodal Predictor 以外の動的な分岐予測機構に対して有効であるかの調査をすることが必要である。

謝辞 本研究にあたり助言を頂いた、山名研究室の斎藤史子氏に感謝の意を表する。

参考文献

- [1] Joseph A. Fisher, Stefan M. Freudenberger: "Predicting Conditional Branch Directions From Previous Runs of a Program", Proc. of ASPLOS-V, pp.85-95, 1992
- [2] JT. Ball, J. R. Larus: "Branch prediction for free", Proc. of PLDI'93, pp.300-313, 1993.6
- [3] JS. McFarling: "Combining branch predictors", Technical Report TN-36, Digital Western Research Laboratory, 1993.6
- [4] JP. Chang, E. Hao, T. Yeh, and Y. Patt: "Branch classification: a new mechanism for improving branch predictor performance", Proc. of MICRO-27, pp.22-31, 1994.11
- [5] JM. Haungs, P. Sallee, M. Farrens: "Branch Transition Rate: A New Metric for Improved Branch Classification Analysis", Proc. of 6th HPCA, pp.241-250, 2000
- [6] J.E.Smith: "A Study of Branch Prediction Strategies", Proc. of 8th ISCA, pp.135-148, 1981
- [7] T-Y. Yeh, Y. N. Patt: "A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History", ISCA'93, pp.257-266, 1993
- [8] JD. Burger, T. Austin: "The SimpleScalar Tool Set Version 2.0", University of Wisconsin-Madison Computer Sciences Tech Report #1342, 1997.6