

極端な偏りを利用する Bimode++ 分岐予測器の提案

吉瀬 謙^{†,††} 片桐 孝洋^{†,††}
本多 弘樹[†] 弓場 敏嗣[†]

命令発行幅の増大と命令パイプライン長の増大により、プロセッサ性能に与える分岐予測器の重要性が増している。予測精度を向上させるために、過去数十年の間に様々な分岐予測器が提案されている。本稿では、Bimode 分岐予測器の精度を向上させる幾つかの手法を議論し、新しい構成の分岐予測器として Bimode++ 分岐予測器を提案する。SpecFP, SpecINT, マルチメディア, サーバの領域を含むベンチマークプログラムを用いた評価結果から、Bimode++ 分岐予測器の優れた予測精度を確認する。

The Bimode++ Branch Predictor Using the Feature of Extremely Biased Branches

KENJI KISE,^{†,††} TAKAHIRO KATAGIRI,^{†,††} HIROKI HONDA[†]
and TOSHITSUGU YUBA[†]

Accurate branch prediction has come to play an important role for modern microprocessors. In order to improve its prediction accuracy, many branch predictors have been investigated in the past few decades. In this paper, the Bimode++ branch predictor is proposed as an enhanced version of Bimode branch predictor. The Bimode++ branch predictor is unique by using the feature of extremely biased branches. Our experimental results using benchmarks from SpecFP, SpecINT, multi-media and server area show that Bimode++ branch predictor gives better prediction accuracy than Bimode branch predictor.

1. はじめに

命令発行幅の増大と命令パイプライン長の増大により、プロセッサ性能に与える分岐予測器の重要性が増している。予測精度を向上させるために、過去数十年の間に様々な分岐予測器が提案されている。

本稿では、極端な偏りのある分岐命令の特徴⁴⁾を利用する分岐予測器の精度向上手法を議論する。極端な偏りのある分岐命令の特徴に加えて、幾つかの発見を組み合わせた形の分岐予測器として Bimode++ 分岐予測器を提案する。Bimode++ 分岐予測器は、我々が 2003 年に発表した Bimode-Plus 分岐予測器⁶⁾の改良型である。

SpecFP, SpecINT, マルチメディア, サーバの領域を含むベンチマークプログラムを用いた評価結果から、Bimode++ 分岐予測器の優れた予測精度を確認する。

2. 関連研究

Gshare 予測器²⁾において、複数の分岐命令の結果を一つの飽和型カウンタに格納することで予測精度が低下するという問題(破壊的競合)を緩和するために、Lee ら¹⁾により提案された Bimode 予測器の構成を図 1 に示す。Bimode 予測器は Choice PHT, Taken PHT, Untaken PHT という 3 つのテーブルを利用する。ここで、Taken PHT と Untaken PHT のことを Direction PHT と呼ぶ。Choice PHT は 2 ビットカウンタの配列で、分岐命令のアドレス(プログラムカウンタ)の一部を用いて生成するインデックスで参照される。Direction PHT も 2 ビットカウンタの配列で、グローバルな分岐履歴レジスタ(BHR)と分岐命令のアドレスとの排他的論理和により生成されるインデックスで参照される。Choice PHT が分岐成立と予測した場合には Taken PHT を、そうでない場合には Untaken PHT を用いて予測をおこなう。

我々は、Bimode 分岐予測器の改良型として、過去の分岐方向の履歴が全て同様の挙動を示す「極端な偏りのある分岐命令」の存在に注目した Bimode-Plus 分

† 電気通信大学 大学院情報システム学研究所

Graduate School of Information Systems, The University of Electro-Communications

†† 独立行政法人科学技術振興機構 さきがけ

“Information infrastructure and applications”,

PRESTO, Japan Science and Technology Agency(JST)

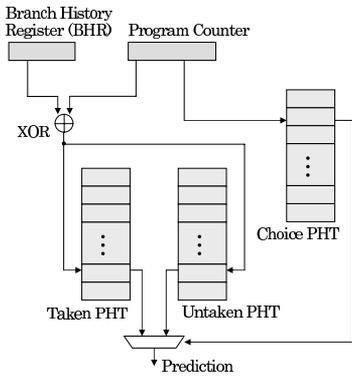


図 1 Bimode 分岐予測器

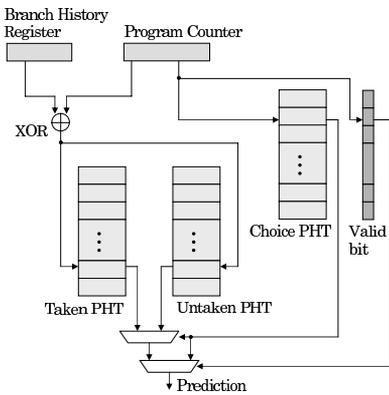


図 2 Bimode-Plus 分岐予測器

岐予測器 (図 2) を提案してきた。本稿では、Bimode-Plus 分岐予測器に幾つかの改良を施すことで予測精度の向上を狙う。改良を施した予測器として Bimode++ 分岐予測器を提案し、その予測精度を評価する。

提案手法は BTB に変更を加える必要はない。また、BTB のエントリ有無を参照した分岐予測器⁷⁾ のように、予測結果を得るために BTB を参照する必要はない。

3. Bimode 予測器に対する幾つかの改良

3.1 不安定状態の特徴を利用する機構

Bimode 予測器の Direction PHT では、ある閾値を設定しその閾値より飽和型カウンタの値が大きい場合に分岐成立、そうでない場合に不成立と予測する。

図 3 の状態遷移に基づく 2 ビット飽和型カウンタの場合には、閾値として値 1 が利用される。2 ビット飽和型カウンタの値が 1 より大きい場合に分岐成立、1 以下の場合に不成立と予測される。しかしながら、例えば、予測結果が分岐成立の場合でも、閾値に近い値の場合 (2 ビットの場合には値 2 の Likely Taken) と、閾値から遠い値の場合 (2 ビットの場合には値 3

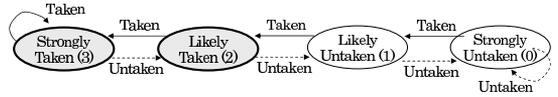


図 3 2 ビット飽和型カウンタの状態遷移

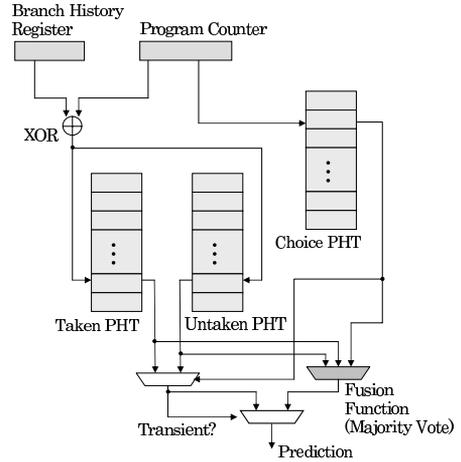


図 4 不安定状態の特徴を利用する Bimode 分岐予測器 (Bimode-Transient)

の Strongly Taken) における予測の精度が同じとは考えにくい。ここで、飽和型カウンタが閾値に近い値を持つ状態のことを、近い将来に予測結果が反転する可能性が高いという点に注目し、「不安定状態」と呼ぶことにする。

トレース駆動シミュレータを用いて、Bimode 予測器の Direction PHT が不安定状態とそうでない場合の予測精度を測定したところ、明らかに、不安定状態の場合の予測精度が悪いことが明らかになった。本節では、この点に注目して、不安定状態の特徴を利用する機構⁵⁾ を提案する。

従来手法の Bimode 予測器では、予測結果を生成する際に、Taken PHT あるいは Untaken PHT のどちらかを選択するためだけに Choice PHT が利用される。また、Taken PHT が利用される時に Untaken PHT は参照されない。同様に、Untaken PHT が利用される時には、Taken PHT の内容が予測結果に影響を与えることはない。

しかしながら、例えば、Untaken PHT が利用され、値が不安定状態である場合には、短い時間の後に Taken PHT が利用される可能性が高い。このため、不安定状態の場合には、Taken PHT と Untaken PHT の両方のパターン履歴表に当該分岐命令の情報が格納されている確率が高いと考えられる。

これらを考慮し、不安定状態と判定された場合には、Choice PHT, Taken PHT, Untaken PHT の 3 つの出力を入力とする関数 (Fusion function) により予測結果を生成する Bimode-Transient(図 4) 方式を提案

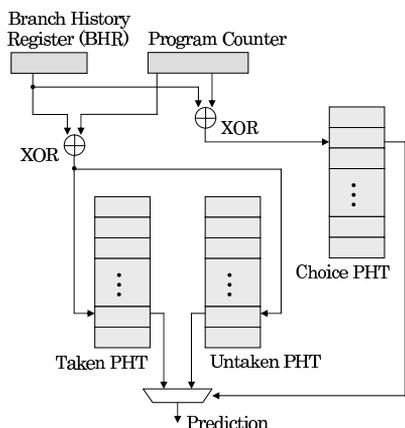


図5 分岐履歴を用いて Choice PHT を参照する Bimode 分岐予測器 (Bimode-BHRC)

する。予測値を生成する関数として、3つのパターン履歴表の予測結果において、2つ以上のパターン履歴表が生成した出力を最終的な予測値とする多数決回路を採用することで予測精度の向上が得られることを確認している。その他の関数の検討は今後の課題である。

3.2 分岐履歴を用いた Choice PHT インデックス

従来手法の Bimode 予測器では、プログラムカウンタのみを利用して Choice PHT のインデックスを作成する。Direction PHT(Taken PHT, Untaken PHT)と同様に、分岐履歴レジスタ (BHR) とプログラムカウンタとの排他的論理和を利用してインデックスを生成することで、幾つかのベンチマークにおける分岐予測精度が向上する。

このように変更を加えた予測器 (Bimode-BHRC) の構成を図5に示す。基本的に、静的な分岐命令の挙動を記憶することが Choice PHT の目的であるため、Choice PHT のインデックス作成に利用する BHR のビット数はそれほど大きくする必要はない。ハードウェア量が 8KB の場合には、3 ビット程度の分岐履歴を利用することで、最も高い予測精度を達成する。

3.3 極端な偏りを利用する分岐予測器

個々の分岐命令の挙動に注目して調査をおこなったところ、プログラムの開始から終了までの全ての実行において必ず分岐成立する分岐命令、あるいは必ず不成立である分岐命令が多数存在することが判明した。これら、全て同様の挙動を示す分岐命令のことを「極端な偏りのある分岐命令」と呼ぶことにする⁴⁾。その中で、必ず成立する分岐命令のことを「極端な偏りのある成立分岐命令」、必ず不成立する分岐命令のことを「極端な偏りのある不成立分岐命令」と呼ぶことにする。

幾つかの方式を評価した結果、8KB のハードウェア量では、極端な偏りのある不成立分岐命令のみを利用する方式が最も高い予測精度を達成することが判明

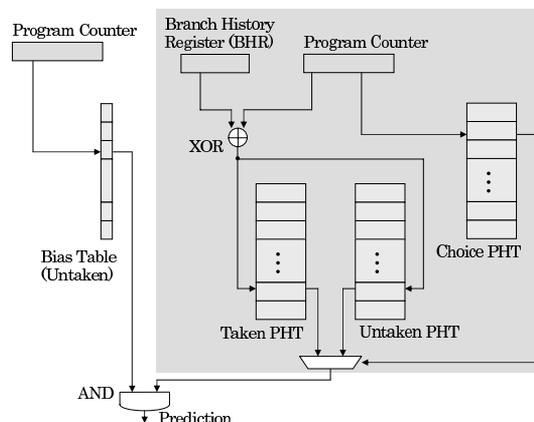


図6 Bimode-Plus(Untaken) 分岐予測器

した。ただし、極端な偏りのある成立分岐命令と極端な偏りのある不成立分岐命令とを利用する Bimode-Plus 予測器との予測精度の差は非常に小さい。本稿では、極端な偏りのある不成立分岐命令のみを利用する Bimode-Plus 分岐予測器を利用する。

極端な偏りのある不成立分岐命令を利用する Bimode-Plus 予測器の構成を図6に示す。図右に示した網掛けの部分が Bimode 予測器である。左側に示した1ビットの配列として構成される Bias テーブルには、過去の分岐方向が全て不成立であるか否かという情報を、それぞれの分岐命令毎に格納する。

予測の方法を示す。プログラムカウンタからインデックスを生成し、予測すべき分岐命令に対応する Bias テーブルのエントリを参照する。もし、極端な偏りのある不成立分岐命令と判断された場合には、分岐不成立と予測する。そうでなければ、右側に示した Bimode 予測器から得られる出力を用いて予測をおこなう。

分岐履歴レジスタ (BHR) の更新方式を検討する。Bimode 予測器と同様に、全ての条件分岐命令の結果を用いて分岐履歴レジスタを更新する方式を考える。このとき、特に、極端な偏りのある分岐命令が頻繁に出現する場合に、分岐履歴レジスタに格納されたビットのパターンが、前回の実行におけるビットのパターンと同様の傾向を示すようになり、グローバルなパス情報としての役割を果たさない可能性が生じる。このことを考慮し、全ての条件分岐命令の結果を用いて分岐履歴レジスタを更新する方式と、極端な偏りのある分岐命令を除く条件分岐命令の結果を用いて更新する方式を提案する。前者を BHR_ALL 方式、後者を BHR_NOB 方式と呼ぶことにする。

3.4 改良を施した極端な偏りを利用する予測器

BHR_ALL 方式と BHR_NOB 方式とを動的に切り替えることで、予測精度の向上を達成できる。これを実現するために、全ての条件分岐命令の結果を用いて更新される BHR_ALL と、極端な偏りのある分岐

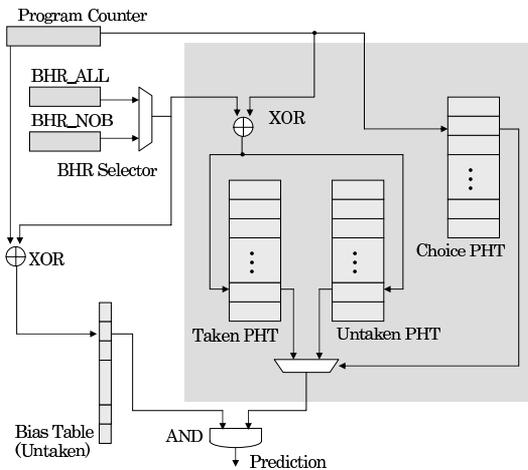


図 7 改良を施した Bimode-Plus2(Untaken) 分岐予測器

命令を除く条件分岐命令の結果を用いて更新される BHR_NOB という 2 種類の分岐履歴レジスタを用意し、選択回路 (BHR Selector) を用いて分岐履歴レジスタを動的に切り替える方式を提案する。

調査の結果、サーバ系のベンチマークのように、極端な偏りをもつ分岐命令と判断される命令の数が多い場合に、BHR_ALL 方式が適しているという傾向を確認した。このため、2 つの分岐履歴レジスタを切り替える方式として、Bias テーブルの値が書き換わった回数を保持するカウンタを追加し、この値がある閾値より大きい場合には BHR_ALL を、そうでない場合には、BHR_NOB を利用する選択方式を採用する。

また、Choice PHT、Direction PHT と同様に、Bias テーブルのインデックス生成に関しても、分岐履歴レジスタとの排他的論理和を用いることで予測精度が向上することを確認している。

これらの改良を施した Bimode-Plus2 分岐予測器の構成を図 7 に示す。左上に 2 つの分岐履歴レジスタと選択回路が追加されている。左下の Bias テーブルは、プログラムカウンタと分岐履歴レジスタとの排他的論理和を用いて参照されるように変更されている。

4. Bimode++分岐予測器の提案

前章で議論した方式を組み合わせる新しい分岐予測器として、Bimode++分岐予測器を提案する。Bimode++分岐予測器の構成を図 8 に示す。改良を施した Bimode-Plus 分岐予測器 (図 7) に、「分岐履歴を用いて Choice PHT インデックスを生成する方式」と、「不安定状態の特徴を利用する機構」を組み合わせた構成を持つ。

予測方法は次のとおりである。BHR_Selector により、BHR_ALL と BHR_NOB から適切な分岐履歴レジスタを選択する。選択した分岐履歴レジスタとプ

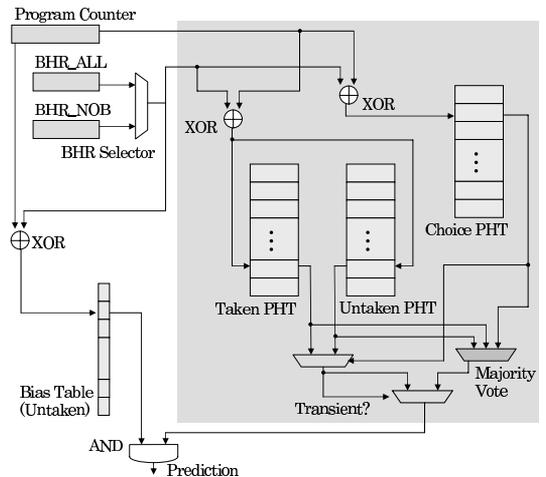


図 8 Bimode++分岐予測器

ログラムカウンタとの排他的論理和を用いてインデックスを生成し、Bias テーブルのエントリを参照する。もし、極端な偏りのある不成立分岐命令と判断された場合には、分岐不成立と予測する。そうでない場合には、右側の網掛けに示した不安定状態の特徴を利用する Bimode 予測器の出力を利用する。

履歴テーブルの更新方法は次のとおりである。分岐結果が成立の場合に、bias テーブルのエントリをリセットする。Direction PHT は、その出力を利用して予測値を生成した場合のみ更新される。Choice PHT は、極端な偏りのある分岐命令と判断されなかった場合に、Bimode 予測器と同様に更新される。BHR_ALL は、全ての条件分岐命令の結果を利用して更新される。BHR_NOB は、極端な偏りのある分岐命令を除く条件分岐命令の結果を利用して更新される。

5. 評価

5.1 評価方法

トレース駆動のシミュレータを利用して分岐予測器の精度を評価する。分岐予測器は、分岐の成立あるいは不成立のどちらか (1 ビットの出力) を予測する条件分岐命令のみを予測の対象とする。ただし、分岐履歴レジスタの更新には CALL 命令の結果を利用することがある。

ベンチマークプログラムは、Intel MRL と IEEE TC-uARCH の支援で開催された分岐予測器のコンテスト Championship Branch Prediction のために作成された 20 本のトレースと枠組 (Version 3) を利用する。これら 20 本のトレースは、SpecFP (FP), SpecINT (INT), マルチメディア (MM), サーバ (SERV) という異なる 4 つの応用分野の実行履歴を集めたものである。個々のトレースは、アプリケーションの実行履歴に加えて、オペレーティングシステ

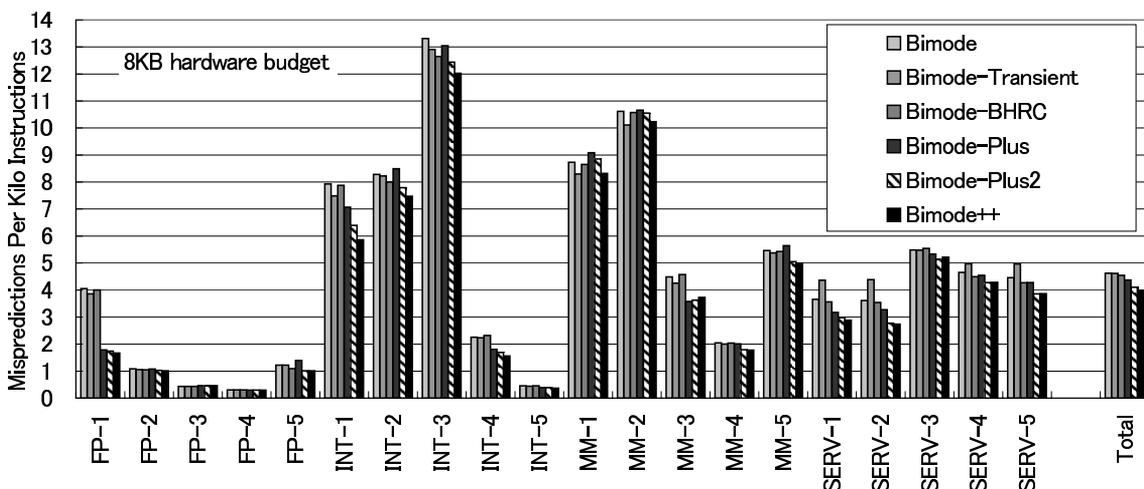


図 9 8KB のハードウェア量における分岐予測器の予測精度の比較

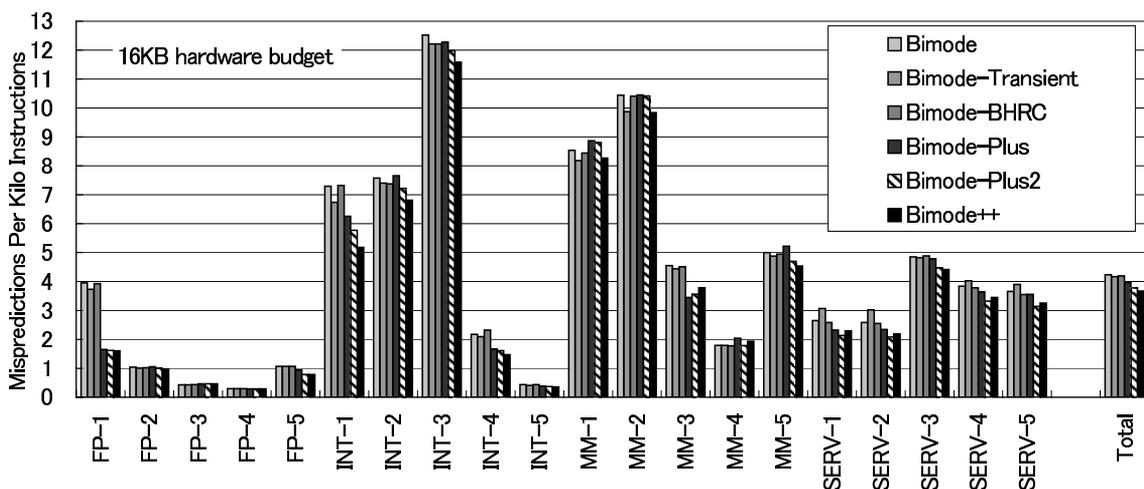


図 10 16KB のハードウェア量における分岐予測器の予測精度の比較

の実行履歴を含んでいる。個々のトレースの総実行命令数は、3 千万命令程度に調整されている。

分岐予測器の予測精度を示す尺度として、1000 命令当たりの分岐予測ミス回数 (Mispredictions Per Kilo Instructions) を利用する。

5.2 分岐予測器の構成

それぞれの分岐予測器は、構成を決める多くのパラメータを持つ。予測精度が高くなるように調整した結果から、これらのパラメータとして次に示す構成を利用する。

Bimode 予測器、不安定状態の特徴を利用する Bimode 分岐予測器、分岐履歴を用いて Choice PHT を参照する Bimode 分岐予測器は、Choice PHT のエントリ数を Direction PHT のエントリ数の 2 倍に設定する。例えば、8KB の予測器では、Choice PHT 4KB, Taken PHT 2KB, Untaken PHT 2KB の合計 8KB となる。

Bimode-Plus 予測器は、Choice PHT のエントリ数と Direction PHT のエントリ数を同一に、Bias テーブルのエントリ数をこれらの倍に設定する。8KB の予測器では、Choice PHT 2KB, Taken PHT 2KB, Untaken PHT 2KB, Bias テーブル 2KB の合計 8KB となる。

改良を施した Bimode-Plus 予測器と Bimode++ 予測器は、Taken PHT のエントリ数を Choice PHT のエントリ数の 4 倍、Untaken PHT のエントリ数を Choice PHT のエントリ数の 2 倍、Bias テーブルのエントリ数を Choice PHT のエントリ数の 2 倍に設定する。8KB の予測器では、Choice PHT 1KB, Taken PHT 4KB, Untaken PHT 2KB, Bias テーブル 1KB の合計 8KB となる。

5.3 評価結果

ハードウェア量を 8KB, 16KB に設定した場合のそれぞれのベンチマークと平均 (Total) の予測精度を

図 9, 図 10 に示す。グラフには, Bimode 分岐予測器 (Bimode), 不安定状態の特徴を利用する Bimode 分岐予測器 (Bimode-Transient), 分岐履歴を用いた Choice PHT インデックスを利用する Bimode 分岐予測器 (Bimode-BHRC), Bimode-Plus 分岐予測器 (Bimode-Plus), 改良を施した Bimode-Plus 分岐予測器 (Bimode-Plus2), Bimode++分岐予測器 (Bimode++) という 6 個のデータを示す。

図 9, 8KB のハードウェア量の結果を議論する。この時, ベースとなる Bimode の 1000 命令当たりの予測ミスの回数は 4.62 である。この値に対する予測ミスの削減率を計算すると, Bimode-Transient が 0.17%, Bimode-BHRC が 1.81%, Bimode-Plus が 5.54%, Bimode-Plus2 が 11.2%, **Bimode++** が 13.7% である。

図 10, 16KB のハードウェア量の結果も 8KB の時と同様の傾向を示す。16KB のハードウェア量の時, ベースとなる Bimode の 1000 命令当たりの予測ミスの回数は 4.23 である。この値に対する予測ミスの削減率を計算すると, Bimode-Transient が 1.52%, Bimode-BHRC が 0.96%, Bimode-Plus が 6.35%, Bimode-Plus2 が 10.7%, **Bimode++** が 13.2% である。

このように, 提案手法の Bimode++分岐予測器は, Bimode 分岐予測器と比較して, 13%を超える予測ミスの削減を達成する。

6. 議論

6.1 パラメータチューニング

本稿では, 8KB のハードウェア量の構成において高い予測精度を達成するようにパラメータをチューニングした構成の予測器を利用した。

特に, Choice PHT, Taken PHT, Untaken PHT という 3 つのテーブルへのハードウェア資源の割り当ては性能に大きな影響を及ぼす。また, こららのパラメータはベンチマークごとに異なる最適値を取りうる事が考えられる。これら 3 つの履歴テーブルを 1 つの大きなテーブルとみなし, 動的に資源を分割することで, 予測精度が向上できる可能性がある。これらハードウェア資源の割り当て方式を含むパラメータチューニングの検討は今後の課題である。

6.2 プロセッサ性能

本稿の評価では, 提案手法の Bimode++予測器の予測精度を測定し, 従来手法と比較して高い予測精度を達成できることを示した。

今後, Bimode++分岐予測器をクロックレベルのプロセッサシミュレータに組み込み, 様々なモデルのプロセッサ性能に与える影響を検討する必要がある。チップマルチプロセッサや SMT の環境では, 分岐予測のテーブルを共有することによる性能低下が問題となる。ユニプロセッサの環境だけではなく, さまざま

なプロセッサアーキテクチャを前提とした検討と評価は今後の課題である。

7. おわりに

命令発行幅の増大と命令パイプライン長の増大により, プロセッサ性能に与える分岐予測器の重要性が増している。

本稿では, 極端な偏りのある分岐命令の特徴を利用する分岐予測器の精度向上手法を議論した。極端な偏りのある分岐命令の特徴に加えて, 不安定状態の特徴を利用する機構, 分岐履歴を用いた Choice PHT インデックス方式を提案し, これらを組み合わせる形の予測器として Bimode++分岐予測器を提案した。

SpecFP, SpecINT, マルチメディア, サーバの領域を含むベンチマークプログラムを用いて予測精度を評価した。従来手法の Bimode 予測器と比較して, Bimode++分岐予測器は, 8KB のハードウェア量のとき 13.7%の予測ミスを, 16KB のハードウェア量のとき 13.2%の予測ミスを削減することを明らかにした。

謝辞

電気通信大学大学院情報システム学研究科の森田啓義助教授, 長岡浩司助教授には刺激的なコメントを頂きました。ここに感謝いたします。本研究の一部は, 科学技術振興機構さきがけプログラムの助成による。

参考文献

- 1) Lee, C.-C., Chen, I.-C. K. and Mudge, T. N.: The bi-mode branch predictor, *Proceedings of the 30th Annual ACM/IEEE International Symposium on Microarchitecture*, pp. 4-13 (1997).
- 2) McFarling, S.: Combining branch predictors, Technical report, Digital Equipment Corporation WRL TN-36 (1993).
- 3) Yeh, T.-Y. and Patt, Y. N.: Two-level adaptive training branch prediction, *Proceedings of the 24th Annual International Symposium on Microarchitecture*, pp. 51-61 (1991).
- 4) 吉瀬謙二: 極端な偏りに基づく分岐予測器, それを組み込んだプロセッサ及びハードウェア予測器, 特願 2003-271395 (2003-07-07).
- 5) 吉瀬謙二: 不安定状態を利用する予測器, プロセッサ, 特願 2004-350584 (2004-12-03).
- 6) 吉瀬謙二, 片桐孝洋, 本多弘樹, 弓場敏嗣: Bimode-Plus 分岐予測器の提案, 電子情報通信学会技術研究報告, CPSY-2003-10, pp. 25-30 (2003).
- 7) 斉藤史子, 山名早人: BTB のエントリ有無を参照した分岐予測器, 情報処理学会論文誌, Vol. 45, No. SIG 11(ACS 7), pp. 71-79 (2004).