

## ハイブリッド分岐方向予測機構の性能比較

斎藤史子<sup>1,2</sup> 北村健志<sup>1</sup> 山名早人<sup>1,2</sup>

<sup>1</sup> 早稲田大学理工学部 <sup>2</sup> アドバンスト並列化コンパイラ研究体

近年、分岐方向予測機構の予測正解率を向上させるための一手法として、複数の予測表を持つ分岐方向予測機構（本稿ではハイブリッド予測機構と呼ぶ）が提案されている。ハイブリッド予測機構は、(1) 予測の信頼性によって、予測結果を採用する予測表を選択する Global/Local 履歴対応予測機構と、(2) 分岐の偏向に応じてエントリを設ける予測表を選択する競合緩和予測機構に大別される。

従来の研究では、これら 2 種類の分岐方向予測機構の性能比較は充分とは言えない。そこで、本稿では、これらのハイブリッド予測機構のハードウェア容量を同一とした場合の予測正解率を比較した。また、我々は、今まで初期 1 億命令をシミュレーション対象としてきたのに対して、本稿では、より精度の高い結果を得るために、初期処理命令除去後の全命令を対象とした。さらに、(1) において競合緩和が考慮されていないことに着目し、予測表更新法を変更することによる競合緩和を試みた。その結果、SPECint95(gcc,ijpeg) において、(1) は、(2) に比較し予測正解率が 0.1~1% 優れることを確認した。また、(1) は、予測表を常時更新する場合と比較して、競合の多いプログラム (gcc) では、0.5% 予測正解率が向上した。以上から、(1) と (2) の予測正解率の差は、ほとんどないことと、競合の多いプログラムでは、(1) の予測表の更新法に改良の余地のあることが判明した。

## Hybrid Branch Predictors Evaluation on Prediction Accuracy

Fumiko Saito<sup>1,2</sup> Takeshi Kitamura<sup>1</sup> Hayato Yamana<sup>1,2</sup>

<sup>1</sup> School of Science and Engineering, Waseda Univ. <sup>2</sup> Advanced Parallelizing Compiler Project

In recent years, branch predictors with multiple prediction tables, which are called "Hybrid Predictor" in this paper, have been proposed. "Hybrid Predictor" is classified into two categories. One is "Combining Predictor", the other is "De-Aliased Predictor". The difference between "Combining Predictor" and "De-Aliased Predictor" is a means to select a prediction table. "Combining Predictor" select a prediction table by confidence. "De-Aliased Predictor" select a prediction table by branch direction bias. Although the prediction accuracy in "Combining Predictor" is the highest, "Combining Predictor" has been mostly compared with "De-Aliased Predictor".

In this paper, we compare the prediction accuracy in "Combining Predictor" with the one in "De-Aliased Predictor" on the same hardware buget. Although we have selected one hundred million first instructions, in this paper we select all instructions except input processing. We try to alleviate aliasing in "Combining Predictor" by update mechanism on "De-Aliased Predictor" (ex. partial update). As the result, in gcc and ijpeg(SPECint95) "Combining Predictor" predicts branch directions 0.1-1% better than "De-Aliased Predictor". In gcc, "Combining Predictor" with partial update predicts branch direction about 0.5% better than "Combining Predictor" with total update. These show that "Combining Predictor" and "De-Aliased Predictor" are little difference on prediction accuracy, and that "Combining Predictor" may predict better by improving update mechanism.

### 1 はじめに

近年、分岐方向予測機構における予測正解率を向上させるための一手法として、複数の予測表を持つ分岐方向予測機構が提案されている。本稿では、複数の予測表を持つ分岐方向予測機構をハイブリッド予測機構と呼ぶ。

ハイブリッド予測機構は、Global/Local 履歴対応予測機構<sup>1</sup>と、競合緩和予測機構に大別される。Global/Local 履歴対応予測機構では、予測の信頼性によって、予測結果を採用する予測表を決定する。Global/Local 履歴対応予測機構としては、1993 年

に McFarling によって提案された Combining 予測機構 [1] が挙げられる。競合緩和予測機構は、分岐の偏向によってエントリを設ける予測表を決定する。競合緩和予測機構は、予測正解率の向上を妨げる要因は競合であるという観点から、1997 年以降に提案され、1997 年に Michaud らの提案した Skewed 予測機構 [2]、Lee らの提案した Bi-Mode 予測機構 [3]、1998 年に Eden らの提案した YAGS(Yet Another Global Schme) 予測機構 [4] などが挙げられる。

本稿では、Global/Local 履歴対応予測機構が最も優れた予測機構であるにも関わらず、競合緩和予測機構と比較されていない点に着目して、これらの予測機構のハードウェア容量を同一にした場合の予測

<sup>1</sup>Global/Local 履歴対応予測機構は、一般にハイブリッド予測機構と呼ばれる。

表 1: SPECint95 の命令数

プログラム名	入力	初期命令	初期条件分岐命令	全命令	全条件分岐命令
099.go	null.in	974,832	123,094	36,136,596,014	4,188,219,317
124.m88ksim	ctl.in	65,514	7,570	64,578,245,993	9,146,618,664
126.gcc	varasm.i	725,523	104,424	251,477,759	38,175,283
129.compress	bigtest.in	1,645,656,980	192,936,246	42,389,729,773	5,759,951,247
130.li	*.lsp	205,901	19,644	55,691,376,022	8,493,525,398
132.jpeg	vigo.ppm	52,439,144	4,931,539	29,798,785,814	1,684,293,060
134.perl	primes.pl	38,524	5,360	14,484,690,488	1,738,590,139
147.vortex	vortex.in	136,988	31,779	76,691,376,022	9,542,280,537

正解率を比較する。また、我々は、今まで初期 1 億命令をシミュレーション対象としてきた [5] のに対して、本稿では、より精度の高い結果を得るために、初期処理命令除去後の全命令を対象とした。さらに、Global/Local 履歴対応予測機構において競合緩和が考慮されていないことに着目して、Global/Local 履歴対応予測機構に対する競合緩和予測機構の予測表更新法の適用による競合緩和を試みた。

以降、2 で、シミュレーション対象命令の選定、3 で、比較対象とした分岐予測機構の構成を説明する。4 では、実験環境と実験結果について述べ、5 で全体をまとめる。

## 2 シミュレーション対象命令の選定

シミュレーション対象命令の選定に先立って、シミュレーション対象命令時間の短縮のための課題を確認する。

課題 1 初期データ準備等の本来のコードとは、無関係な部分を除去すること。

課題 2 プログラム全体の挙動が、制限した部分で表現されていることが示されていること。

現在、多用されているシミュレーション時間の短縮方法として、「シミュレーション対象命令の制限」と、「入力セットの調整」が挙げられる。Skadron ら [6]、Sherwood ら [7] によってプログラムの周期性を利用した「シミュレーション対象命令の制限」が提案されているが、これらの手法では、単一の予測機構のみに対して実験を行っている。そのため、さまざまな予測機構の性能比較を行う際には、不適切である。また、「入力セットの調整」には、SPEC-lite<sup>2</sup> [8] のような調整した入力セットや、train 入力セットを採用する場合がある。しかし、文献 [9] では、SPEC2000<sup>3</sup>に

<sup>2</sup>SPEC2000 における簡易入力セット。

<sup>3</sup>art, gzip, vortex, vpr.

において、これらの手段では誤差のあることが示されている。

そこで、本稿では、シミュレーション時間を短縮する上で、課題 1 に着目し、入力処理命令を初期処理とみなして除去して、予測正解率を評価した。SPECint95 において初期処理命令と判断した命令数を表 1<sup>4</sup>に示す。

## 3 予測機構の構成

本稿では、ハイブリッド予測機構の構成要素となる予測機構を単体予測機構と呼ぶ。以下に、単体予測機構、ハイブリッド予測機構の構成について説明する。

### 3.1 単体予測機構

単体予測機構は、ハイブリッド予測機構の構成要素となる予測機構である。代表例として、Bimodal 予測機構、Gshare 予測機構 [1] を以下に示す (図 1)。

Bimodal 予測機構は、対象となる分岐命令の PC<sup>5</sup>をそのまま予測表のインデクスとする。これに対して、Gshare 予測機構では (PC XOR (Global 分岐履歴)) をインデクスとすることで、分岐方向遷移の分岐履歴依存を利用した予測を可能とする。

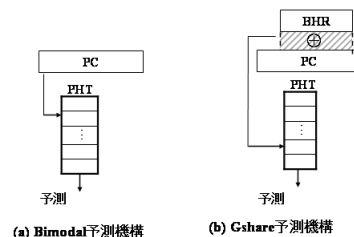


図 1: 単体予測機構

<sup>4</sup>gcc2.7.2.3 によって、-O2 -funroll-loops でコンパイル。Solaris で実行。  
<sup>5</sup>下位ビット。

### 3.2 ハイブリッド予測機構

ハイブリッド分岐方向予測機構は、複数の予測表で構成され、Global/Local 履歴対応予測機構と競合緩和予測機構に分類される。以下、これらの予測機構の構成について説明する。

#### 3.2.1 Global/Local 履歴対応予測機構

Global/Local 履歴対応予測機構は、複数の単体の予測機構を予測表として組み合わせた予測機構である(図2<sup>6</sup>)。予測の成功・失敗で遷移する2-bit 飽和カウンタ<sup>7</sup>の表である選択器によって、採用する予測表を決定する。予測表の採用・不採用に関わらず、常に予測表を更新する。

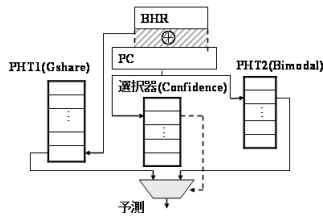


図 2: Global/Local 履歴対応予測機構 [1]

#### 3.2.2 競合緩和予測機構

競合緩和予測機構は、分岐の偏向によってエントリを設ける予測表が決定される。以下に、個々の競合緩和予測機構について説明する。

**Skewed 予測機構** Skewed 予測機構 [2] は、異なるインデクス法を採用した3つの予測表<sup>8</sup>で構成される(図3)。予測は、3つの予測表の予測結果の多数決によって決定する。予測表の更新は、部分的更新 (partial update) — 予測が正解した場合、予測が正

<sup>6</sup>本稿で実験対象とした Global/Local 予測機構は、Gshare 予測機構と Bimodal 予測機構を組み合わせた構成を示す。

<sup>7</sup>構成要素となる予測機構の予測が互いに異なる時、予測が正解した方に遷移する。

<sup>8</sup>インデクスに採用されたハッシュ関数  $f_0, f_1, f_2$  を以下に示す。

$$V = (\text{分岐命令アドレス}, \text{分岐履歴}) = (V_3, V_2, V_1)$$

$$H : (y_n, y_{n-1}, \dots, y_1) \mapsto (y_n \oplus y_1, y_n, y_{n-1}, \dots, y_3, y_2)$$

$$f_0 : (V_3, V_2, V_1) \mapsto H(V_1) \oplus H^{-1}(V_2) \oplus V_2$$

$$f_1 : (V_3, V_2, V_1) \mapsto H(V_1) \oplus H^{-1}(V_2) \oplus V_1$$

$$f_2 : (V_3, V_2, V_1) \mapsto H^{-1}(V_1) \oplus H(V_2) \oplus V_2$$

Global 履歴を参照する Skewed 予測機構は、特に、Gskewed 予測機構と呼ばれている。Gskewed 予測機構の中でも、予測表のひとつを Bimodal 予測機構と同様に分岐命令の PC でインデクス付けた予測機構は、Enhanced Gskewed 予測機構と呼ばれている。他の予測表に対するハッシュ関数には、 $f_1, f_2$  を用いる。

解した予測表のみ、予測が不正解だった場合、すべての予測表を更新する—によって行われる。

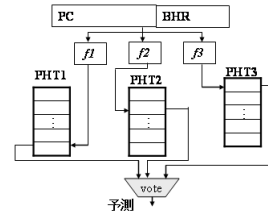


図 3: Skewed 予測機構 [2]

**Bi-Mode 予測機構** Bi-Mode 予測機構 [3] は、PC によってインデクス付けられた Choice PHT、(PC XOR Global 分岐履歴) によってインデクス付けられた、2つの Direction PHT で構成される(図4)。Choice PHT によって、Direction PHT を選択し、Direction PHT が予測を行う<sup>9</sup>。Choice PHT は、常に更新されるが、Direction PHT は、予測結果が採用された PHT のみ更新される。

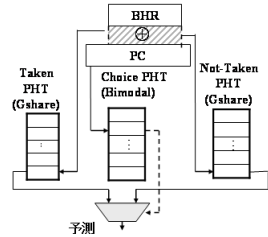


図 4: Bi-Mode 予測機構 [3]

**YAGS 予測機構** YAGS 予測機構 [4] は、Bi-Mode 予測機構における Direction PHT の冗長性<sup>10</sup>を排除するために考案された。YAGS 予測機構は、Direction PHT に PC の下位ビットを保持する以外は、Bi-Mode 予測機構と同じ構成である(図5)。Choice PHT の予測結果と反対の方向を名に冠した Direction PHT にエントリが存在しない限り、Choice PHT の予測結果を採用する。Choice PHT の予測結果と反対の方向を名に冠した Direction PHT にエントリが存在する

<sup>9</sup>Choice PHT において Taken と予測される時、Taken PHT における予測が採用され、Choice PHT において Not-Taken と予測される時、Not-Taken PHT における予測が採用される。

<sup>10</sup>ここで言う冗長性とは、Bi-Mode 予測機構において、同一分岐が、Taken PHT と Not-Taken PHT のそれぞれにエントリを保持する可能性のことを意味する。

場合には、Direction PHT の予測結果を採用する<sup>11</sup>。Choice PHT は、常に更新されるが、Direction PHT は、予測結果が採用された PHT のみ更新される。

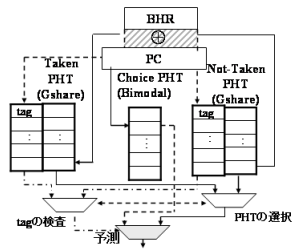


図 5: YAGS 予測機構 [4]

## 4 実験

本節では、実験環境と実験結果について述べる。比較対象として選択した予測機構の構成と予測アルゴリズムの詳細については、前節で紹介した。

### 4.1 実験環境

本稿では、条件分岐のみを対象に分岐予測正解率を算出するように改良を加えた SimpleScalar tool set Version 3.0c/PISA [10] の sim-bpred シミュレータ<sup>12</sup>を採用した。実験に用いたベンチマークは、SPECint95 の gcc, jpeg であり、表 1 に示した入力で、初期命令を除去して実験を行った。プログラムは、gcc 2.7.2.3 によって、-O2 -funroll-loops でコンパイルされている。比較対象として選択した分岐方向予測機構の容量は、0.5 ~ 128KB<sup>13</sup>である。表 2 に比較対象とした予測機構とその略号を示す。表内で履歴長について、特に触れられていない場合には、最長履歴長をとるものとする<sup>14</sup>。また、ハイブリッド予測機構の予測表の構成比を決定するために、初期命令を除去後の 10 億命令に対するシミュレーションを行った。この結果に基づいて、容量毎に最も高

<sup>11</sup>Choice PHT で Taken と予測された場合、Not-Taken PHT にタグを利用してエントリの有無を確認する。Direction PHT にエントリが存在しない場合には、Choice PHT の予測結果、つまり、Taken を最終的な予測として採用する。Direction PHT にエントリが存在する場合には、Direction PHT の予測結果を最終的な予測とする。Choice PHT で Not-Taken と予測された場合には、Taken PHT に対して、同様の操作を行う。

<sup>12</sup>sim-bpred シミュレータでは、命令は逐次実行されると想定され、前の分岐が解決するまでは次の分岐は実行されない。

<sup>13</sup>YAGS 予測機構に関しては、タグも予測機構の容量に含めた。全体の予測機構容量を 0.5 ~ 128KB に収まる範囲で、 $2^5 \sim 2^{19}$  エントリで構成される PHT のすべての組み合わせの試行を行った。

<sup>14</sup>Gshare 予測機構は、分岐履歴長を調整することが可能であるが、短い分岐履歴長では、Bimodal 予測機構との構成上の差異が小さくなってしまったため、最長の分岐履歴長をインデクスに採用した。特に、Global/Local 対応履歴予測機構では、構成要素となる予測機構同士の予測可能範囲が異なることが重要となる [11, 12]。

い予測正解率に達する構成を選択した。以降に示す予測正解率は、初期命令除去後の命令に対して算出している。

さらに、競合緩和予測機構の特徴が、インデクス法、採用予測表の選択法だけではなく、予測表の更新法にあることに着目し、Global/Local 履歴対応予測機構に競合緩和予測機構の更新法の適用を試みた (B\_Select, B\_Partial)。以下に、競合緩和予測機構の更新法についてまとめる。

**Select** 採用した予測表のみ更新する。Bi-Mode 予測機構、YAGS 予測機構の更新法と同等。

**Partial** 採用した予測が正しい場合は、正しく予測した予測表のみ更新する。採用した予測が間違えている場合は、すべての予測表を更新する。Skewed 予測機構の更新法と同等。

他に、Global/Local 履歴対応予測機構において、選択器を分岐履歴によってインデクス付けると予測正解率が向上すること [13] から、(PC XOR Global 分岐履歴) でインデクス付けられた、Bimodal 予測機構と同容量<sup>15</sup>の選択器を用いた機構 (G\_Always) も比較対象とした。

### 4.2 実験結果

まず、ハイブリッド予測機構の予測正解率を比較する (図 6(a), 図 7(a), 表 3)。最も高い予測正解率の予測機構は、gcc では、B\_Partial, jpeg では、G\_Always だった。これらの予測機構は、どちらも Global/Local 履歴対応予測機構である。競合緩和予測機構で最も高い予測正解率に達する Bi-Mode 予測機構よりも、B\_Partial, G\_Always の予測正解率は、0.5 ~ 1%程度高い。

次に、個々の競合緩和予測機構の予測正解率をみてる (図 6(b), 図 7(b), 表 3)。本実験では、Enhanced Gskewed 予測機構の予測正解率が他の文献 [15] に比べて向上しなかった。その原因には、予測表に対するインデクスを最適な履歴長に調整しなかったことが考えられる。Bi-Mode 予測機構の予測正解率が、Global/Local 履歴予測機構と比較して向上しなかった原因は、Choice PHT の分岐毎の偏向の判定にお

<sup>15</sup>構成要素である Gshare 予測機構と同容量の選択器では、Gshare 予測機構の容量を大きくする際に、選択器の容量も大きくなってしまった。そのため、ハードウェア容量を基準として比較する際、Gshare 予測機構に十分な容量が確保できず、予測正解率が向上しないことが観測された。

表 2: シミュレーション対象予測機構

略号	予測機構の説明
単体予測機構	
Bimodal	Bimodal 予測機構 .
Gshare	Gshare 予測機構 .
ハイブリッド予測機構	
B_Always	Bimodal 予測機構と Gshare 予測機構を組み合わせた Global/Local 履歴対応予測機構 . 選択器: Bimodal 予測機構と同容量 . 分岐命令アドレスの下位ビットでインデクス . 更新法: 常時 .
B_Select	B_Always と同じ構成 . 更新法: 予測を採用された予測表のみ更新 .
B_Partial	B_Always と同じ構成 . 更新法: 予測が正しい時は, 正しく予測した予測表のみ . 予測が間違えた時は, すべての予測表 .
G_Always	Bimodal 予測機構と Gshare 予測機構を組み合わせた Global/Local 履歴対応予測機構 . 選択器: Bimodal 予測機構と同容量 . 分岐命令アドレスの下位ビットと分岐履歴の XOR でインデクス . 更新法: 常時 .
Gskewed	Enhanced Gskewed 予測機構 . 分岐履歴長: $2^n$ エントリの予測表の場合, $n-3$ .
Bi-Mode	Bi-Mode 予測機構 .
YAGS	YAGS 予測機構 .

表 3: 予測正解率 (%)

126.gcc	予測機構容量 (KB)					132.jpeg	予測機構容量 (KB)				
	0.5	2	8	32	128		0.5	2	8	32	128
Bimodal	85.37	87.17	87.69	87.79	87.80	Bimodal	87.93	87.93	87.93	87.93	87.93
Gshare	81.30	86.95	90.11	92.23	94.27	Gshare	88.35	89.14	89.90	90.19	90.33
B_Always	85.04	90.41	92.86	94.21	95.45	B_Always	89.24	89.58	90.00	90.16	90.24
B_Select	84.02	89.28	91.66	93.54	94.68	B_Select	88.89	89.32	89.89	90.05	90.16
B_Partial	86.38	90.96	92.91	94.33	95.17	B_Partial	88.91	89.46	89.97	90.11	90.19
G_Always	84.58	90.33	92.78	94.13	95.24	G_Always	89.39	89.75	90.19	90.42	90.52
Gskewed	84.88	89.22	90.87	91.92	92.65	Gskewed	88.15	—	—	—	88.19
Bi-Mode	85.17	89.98	92.43	93.86	95.11	Bi-Mode	88.67	89.59	90.10	90.35	90.45
YAGS	83.44	86.32	87.54	87.82	87.89	YAGS	87.95	87.98	87.99	—	—

—は, それ以前の予測正解率を上回らなかったことを示す .

ける精度が不十分であったことが考えられる . YAGS 予測機構は, Bimodal 予測機構と Gshare 予測機構の中間の予測正解率であることが多い . YAGS 予測機構の予測正解率が向上しない理由は, PC でインデクス付けられた Choice PHT で予測が行われることが多いためと考えられる . 同様の結果が, 文献 [14] に示されている .

最後に, Global/Local 履歴対応予測機構において, 競合緩和予測機構の予測表の更新を適用した場合の効果についてまとめる (図 6(c), 図 7(c), 表 3) . Global/Local 履歴対応予測機構の予測ミスの原因には「競合」「選択器の選択ミス」「構成要素となる予測機構では予測不可能であること」などが考えられる . 本稿では, 競合緩和予測機構の更新法を Global/Local 履歴対応予測機構に適用することで「競合」による予測ミスが緩和されることを期待した . その結果, 予測機構の大容量化につれて予測正解率が向上する傾きが大きいため, 競合が大きいと判断されるプロ

グラム (gcc) では, Partial 更新法によって, 予測正解率が 0.1~1%程度向上したが, 競合が小さいと判断されるプログラム (jpeg) では, 予測正解率の向上は見られなかった . このことは, jpeg では「競合」よりも「選択器の選択ミス」や「構成要素となる予測機構では予測不可能であること」による予測ミスの方が頻繁であったためと判断できる .

## 5 おわりに

近年, 複数の予測表を持つさまざまなハイブリッド分岐方向予測機構が提案された . 本稿では, ハイブリッド分岐方向予測機構を, Global/Local 履歴対応予測機構と, 競合緩和予測機構とに大別した . 前者に分類される主な機構として, McFarling の combining 予測機構 [1], 後者に分類される主な機構として, Skewed 予測機構 [2], Bi-Mode 予測機構 [3], YAGS 予測機構 [4] が挙げられる . 本稿では, 0.5~128KB の容量の予測機構を対象に比較評価を行った . さらに, 競合緩和を考慮して, Global/Local 履歴対応予測機構

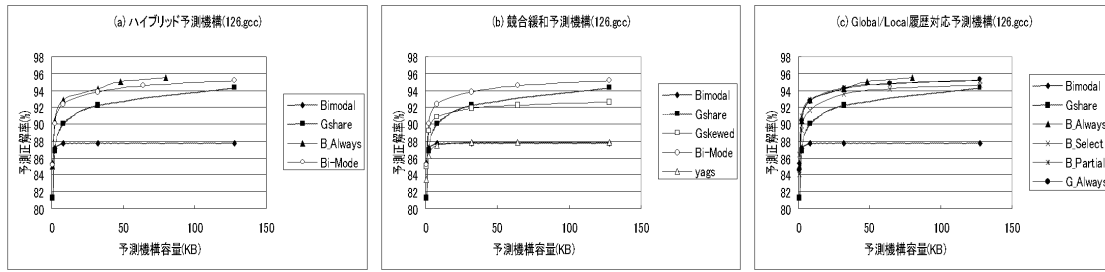


図 6: 126.gcc

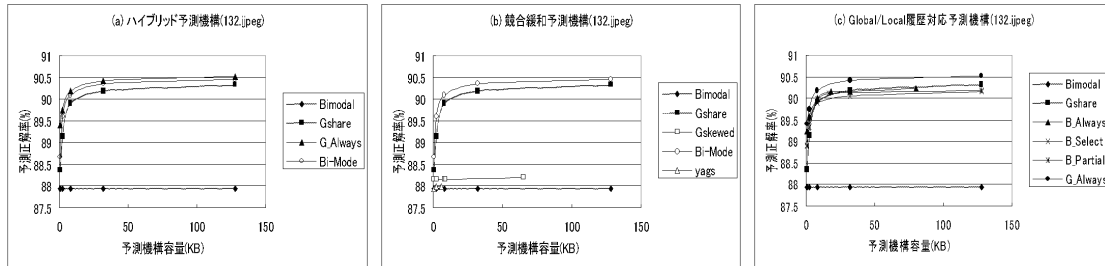


図 7: 132.jpeg

の予測表の更新法を工夫した。

その結果, SPECint95 の gcc, jpeg において, Global/Local 履歴対応予測機構は, 競合緩和予測機構間よりも, 予測正解率は, 0.1 ~ 1%程度だけ優れていた。また, Global/Local 履歴予測機構は, 競合の多いプログラム (jpeg) では, 従来の予測表の常時更新の場合と比較して, 0.5%程度予測正解率が向上した。このことから, Global/Local 履歴対応予測機構と競合緩和予測機構間の予測正解率の差は, ほとんどないことと, 競合の多いプログラムでは, Global/Local 履歴対応予測機構の更新法に改良の余地があることが判明した。

#### 参考文献

- [1] McFarling, S.: Combining branch predictors, Technical Report TN-36, Digital Western Research Laboratory (1993).
- [2] Michaud, P., Sez nec, A. and Uhlig, R.: Trading Conflict and Capacity Aliasing in Conditional Branch Predictors, *Proc. of 24th ISCA*, pp. 292–303 (1997).
- [3] Lee, C. C., Chen, I. K. and Mudge, T. N.: The Bi-Mode Branch Predictor, *Proc. of MICRO-30*, pp. 4–13 (1997).
- [4] Eden, A. N. and Mudge, T. N.: The YAGS Branch Prediction Scheme, *Proc. of MICRO-31*, pp. 69–77 (1998).
- [5] 齋藤, 蛭田, 山名: 複数の PHT を持つ分岐方向予測機構の性能比較, 情処研報 (2002-ARC-149), pp. 55–60 (2002).
- [6] Skadron, K., Ahuja, P. S. et al.: Branch Prediction, Instruction-Window Size, and Cache Size: Performance Trade-Offs and Simulation Techniques, *Trans. on Comput.*, Vol. 48, No. 11, pp. 1260–1281 (1999).
- [7] Sherwood, T. and Calder, B.: Time Varing Behavior of Programs, Technical report, Department of Computer Sciece and Engineering, University of California (1999).
- [8] KleinOowski, A., Flynn, J., Meares, N. and Lilja, D. J.: Adapting the SPEC2000 benchmark suite for simulation-based computer architecture research, *Proc. of WWC-3*, pp. 73–82 (2000).
- [9] J. W. Haskins, J. and Skadron, K.: Techniques for Accurate, Accelerated Processor Simulation: An Analysis of Reduced Inputs and Sampling, Technical report, University of Michigan (2002).
- [10] Burger, D. and Austin, T. M.: The SimpleScalar Tool Set, Version 2.0, Technical report (1997).
- [11] Evers, M., Patel, S. J. et al.: An Analysis of Correlation and Predictability: What Makes Two-Level Branch Predictors Work, *Proc. of 25th ISCA*, pp. 52–61 (1998).
- [12] 森, 小林ほか: 直行性を考慮したハイブリッド分岐予測機構, 情処研報 (1997-ARC-125), pp. 115–120 (1997).
- [13] Chang, P. Y., Hao, E. and Patt, Y. N.: Alternative Implementation of Hybrid Branch Predictors, *Proc. of MICRO-28*, pp. 252–257 (1995).
- [14] Chen, L. and Zhang, S.: Hot Set Detection and Hardware Optimization, <http://www.cs.wisc.edu/~chenl/cs752/paper.html> (2002).
- [15] Lee, C. C., Chen, I. C. and Mudge, T.: Design and Performance Evaluation of Global History Dynamic Branch Predictors, *Proc. of SCI*, pp. 664–671 (1998).