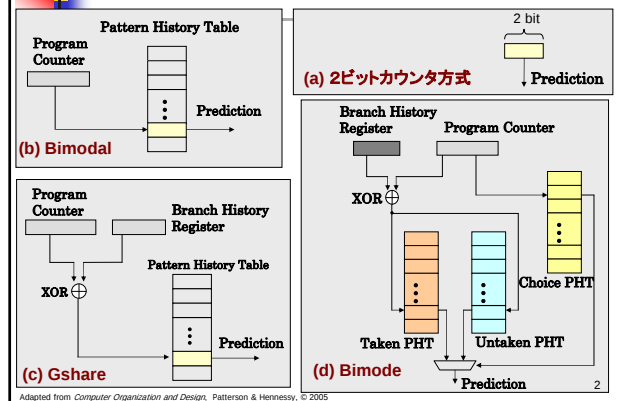


計算機アーキテクチャ特論 (Advanced Computer Architectures)

7. 分岐予測(2)

1

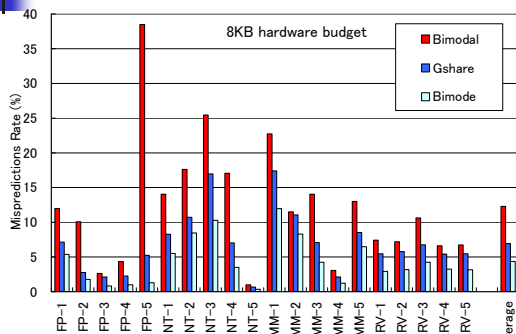
ここまでの分岐予測器



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005.

2

予測精度(予測ミス率)



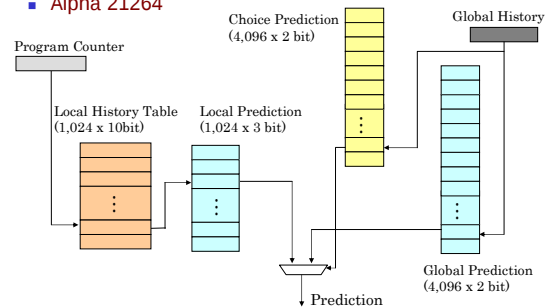
Benchmark for CBP(2004) by Intel MRL and IEEE TC uARCH.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005.

3

ハイブリッド方式 Alpha 21264

- 複数の分岐予測の利点
- Alpha 21264

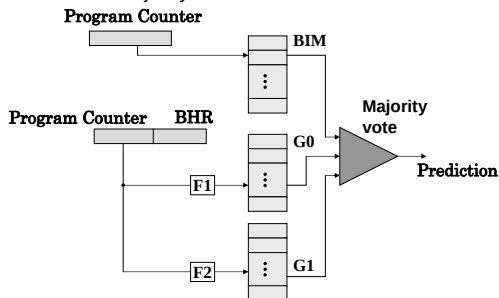


Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005.

4

ハイブリッド方式 E-gskew (ISCA 1997)

- 多数決(Majority vote)により最終予測を決定

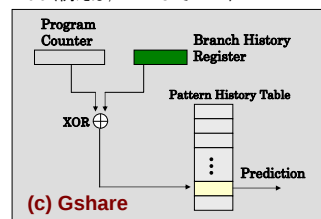


Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005.

5

ハードウェア量と分岐履歴の長さ

- パターン履歴表のハードウェアが支配的
- 2ビットカウンタのNエントリのテーブル
 - $N \times 2 \text{ bit}$ (例えば, 32Kエントリで8KB)
- 分岐履歴レジスタのとりうる最大長を $n \text{ bit}$ とすると
 - $2^n \times 2 \text{ bit}$ (例えば, $n = 15$ で8KB)



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005.

6

パーセプトロン

■ パーセプトロンモデル

- x_1 から x_n までの n ビットの分岐履歴を入力とする。
- y を計算する。
 - w は符号付きの整数で表現
 - y の値がある閾値より高い場合に成立と予測する。

The diagram illustrates a perceptron model. At the top, there are several input nodes represented by circles. The first node is labeled '1' and has a weight 'w0' below it. The next node is labeled 'x1' and has a weight 'w1' below it. This is followed by 'x2' with weight 'w2', and then an ellipsis '...' indicating more nodes, followed by 'xn' with weight 'wn'. Arrows from each of these input nodes point downwards to a single output node labeled 'y'.

$$y = w_0 + \sum_{i=1}^n x_i w_i.$$

Perceptron Model

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

7

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

7

Perceptron (HPCA 2001)

- 分岐アドレスによりパーセプトロンを選択
- グローバル分岐履歴を用いて予測
 - 8KBのハードウェア量で、34ビットの分岐履歴
- 分岐ミス、 $|y| < \theta$ の時にトレーニング

Program Counter

Branch History (x)

Selected Perceptron

Compute y

Prediction

Table of Perceptrons (w)

$$y = w_0 + \sum_{i=1}^n x_i w_i.$$

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

8

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

8

パターンマッチング

分岐履歴 予測: 0 or 1?

最長一致パターン

Prediction

- 過去の履歴との最長一致のパターンを見つける。
- それぞれの最長一致パターンの次に出現する0と1の回数を数え、多い方を予測値とする。

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

9

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

9

パターンマッチングの可能性

マッチングの対象となる分岐履歴の長さ N

$N = 1024 \times 1024$ (one million) に設定

$M = 16, 32, 64, 128, 256, 512, 1024$ に設定

M	16	32	64	128	256	512	1024
SpecFP	5.32	2.34	1.59	0.98	0.54	0.46	0.43
SpecINT	9.92	4.13	2.75	2.39	2.30	2.26	2.16
MultiMedia	8.90	6.51	5.43	5.09	4.99	4.94	4.93
Server	9.20	3.91	2.25	1.91	1.78	1.75	1.73
Average	8.34	4.22	3.00	2.59	2.40	2.35	2.31

- テーブルのエントリ数を 224 と非常に大きく設定した Gshare の予測ミス率は 4.71

吉瀬, 岩田: 分岐予測の精度と履歴情報との関係について, 信学会ソサイエティ大会, 2005

10

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

0

Partial Pattern Matching (CBP 2004)

Table 4 Table 3 Table 2 Table 1 Table 0

pc h[0:9] pc h[0:19] pc h[0:39] pc h[0:79]

12 10 10 10 10

3b ctr 8 bit tag 8 bit u 3b ctr 8 bit tag 8 bit u 3b ctr 8 bit tag 8 bit u 3b ctr 8 bit tag 8 bit u 3b ctr 8 bit tag 8 bit u

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

prediction 0/1

CBP2004プレゼンテーションスライド

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

11

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

11

Benchmark	Bimodal	Gshare	Bimode	PPM
FP-1	12	7	5	2
FP-2	10	2	1	1
FP-3	2	2	1	1
FP-4	4	2	1	1
FP-5	20	5	1	1
INT-1	14	8	5	1
INT-2	17	10	6	7
INT-3	20	17	17	7
INT-4	17	7	3	1
INT-5	5	1	1	1
MM-1	20	17	12	10
MM-2	11	11	8	8
MM-3	14	7	4	1
MM-4	3	1	1	1
MM-5	13	8	6	6
SERV-1	7	6	3	2
SERV-2	7	6	3	2
SERV-3	10	6	4	3
SERV-4	6	5	3	2
SERV-5	6	5	3	2
Average	12	7	4	3

Benchmark for CBP(2004) by Intel MRL and IEEE TC uARCH.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

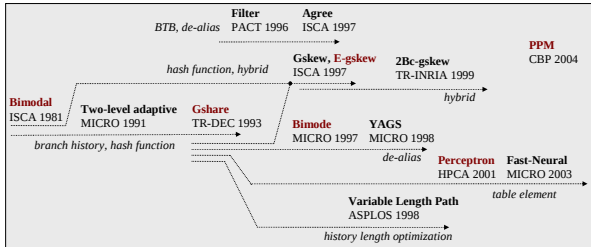
Benchmark for CBP(2004) by Intel MRL and IEEE TC uARCH.

Adapted from *Computer Organization and Design*, Patterson & Hennessy, © 2005

2

分岐予測のまとめ

- 重要な分岐予測の根底にあるアイデアとその構成を解説



ISCA (International Symposium on Computer Architecture)
MICRO (International Symposium on Microarchitecture)
PACT (International Conference on Parallel Architectures and Compilation Techniques)
ASPLOS (International Conference on Architectural Support for Programming Languages and Operating Systems)
Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

13

分岐予測のまとめ

- Bimodal
 - 2ビットカウンタの配列, プログラムカウンタ
- Gshare
 - グローバル分岐履歴の利用
- Bimode
 - 偏りを用いて表を分離, メタな予測器
- ハイブリッド方式
 - 複数の予測方式の利用, 多数決
- パーセプトロン
 - パーセプトロンモデル, 長い分岐履歴の利用
- パターンマッチング (PPM)
 - 最長一致パターンの利用

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

14

分岐予測に関する今後の展望

- 新しい予測方式の開発
 - 予測精度の向上
- 分岐予測の精度の計算機性能
 - 投機的な更新
- パーセプトロンの実現
 - 高い予測精度, 複雑なハードウェア
- 分岐予測の精度と計算機アーキテクチャ
- 他の予測への適用

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

15

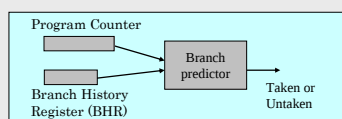
Prophet-Critic Hybrid Branch Prediction

Ayose Falcon, UPC, Jared Stark, Intel, Alex Ramirez, UPC, Konrad Lai, Intel, Mateo Valero, UPC
ISCA-31 pp. 250-261 (2004)

16

研究の目的と概要

- 今までにない, 斬新なアイデアの分岐予測器を提案
- 分岐予測器の予測精度の向上, プロセッサの性能向上を目指す.
- 予測ミス率を大幅に低減する.
 - Gccの場合には, 予測ミス率を 3.11% から 1.23%に削減
- Intel Pentium 4 プロセッサをベースとして, uPC (Uops Per Cycle) で5.2%の性能向上 (Perceptron分岐予測の場合).



従来の分岐予測器の仕組み

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

17

Prophet/critic hybrid の構造

- Prophet: 予言者
- Critic: 批判家, 鑑定家

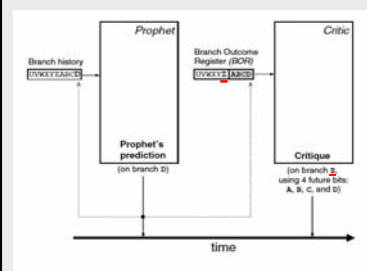


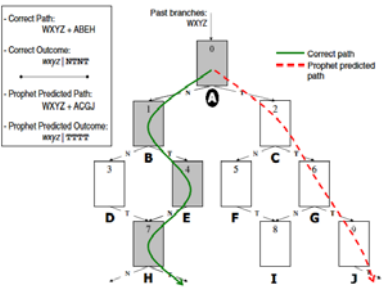
Figure 1. Structure of a prophet/critic hybrid.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

18

- Prophetは, 分岐命令Dの方向を, ...XYZABCの履歴を用いて予測する.
- Criticは, 分岐命令Zの方向を, ...XYZABCDの情報をを用いて鑑定する.
- Criticは, Zの予測を鑑定するが, それには, Zの予測時刻より後の(Zを基準として未来の)情報を利用できる.
- ある予測を前提に処理を進めたところ矛盾が生じる. 前提がおかしいと判断.

Prophet/critic hybrid の予測の例



- 基本ブロック 0, 1, 4, 7 が正しい制御の流れとする。
- 分岐 A の予測を考える。
- Branch History: W, X, Y, Z
- Branch Future: A, C, G and J
- 過去の履歴において, W, X, Y, Z | A=T, C=T, G=T, J=T の場合には, 分岐予測ミスが多いことを記録し, これを用いて鑑定する。

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

19

Decoupled front-end architecture, /SCA-1999

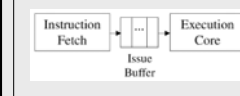


Fig. 1. Two processing engines: the processor pipeline at a high level. The instruction fetch unit prepares and decodes instructions and supplies them to the issue buffer. The execution core consumes instructions from the issue buffer and then orchestrates their execution and retirement. The instruction fetch unit is a fundamental bottleneck in the pipeline. The execution core can only execute instructions as fast as the instruction fetch unit can prepare them.



Fig. 2. The decoupled front-end design at a high level. The fetch target queue (FTQ) buffers fetch addresses provided from the branch predictor. They are queued in the FTQ until they are consumed by the instruction fetch unit, which in turn produces instructions as in an ordinary pipeline. The FTQ allows the branch predictor to continue predicting in the face of an instruction cache miss. It also provides the opportunity for a number of optimizations, including multilevel branch predictor designs and fetch directed cache prefetching and predictor lookup.

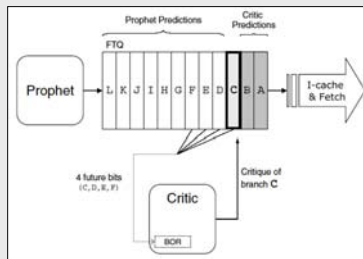
Optimizations Enabled by a Decoupled Front-End Architecture

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

[24] G. Reinman, T. Austin, and B. Calder. A scalable front-end architecture for fast instruction delivery. In *Proceedings of the 26th Annual Int. Symposium on Computer Architecture*, pages 234–245, May 1999.

20

Decoupled front-end architecture を利用した Prophet/critic hybrid の実装



- Prophet, Critic それぞれの予測機構には, 従来の分岐予測機構の任意の組み合わせを利用できる。

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

21

評価環境

- 現実的な, サイクル当たり6命令を処理する高性能マイクロプロセッサ。

Processor Frequency	3.8 GHz
Fetch/Issue/Retire Width	6 uops
Branch Mispredict Penalty	30 cycles
BTB	4096 entries, 4-way
FTQ Size	32 entries
Instruction Window Size	2048 uops
Scheduling Window Size	256 int, 128 mem, 384 fp (sizes in uops)
Load/Store Buffer Sizes	768 load, 512 store (sizes in uops)
Functional Units	6 int, 4 mem, 2 fp
Hardware Data Prefetcher	Stream-based (16 streams)
Instruction Cache	64 KB, 8-way, 64-byte line
L1 Data Cache	32 KB, 16-way, 64-byte line, 3 cycle hit
L2 Unified Cache	2 MB, 16-way, 64-byte line, 16 cycle hit
Memory Latency	100 ns

Table 2. Simulation parameters.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

22

Future bit の重要性, 何回先の予測結果まで利用すればよいか?

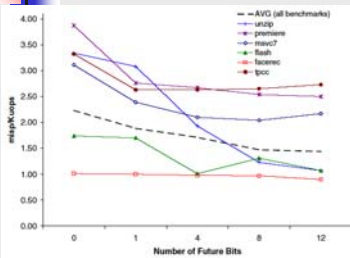
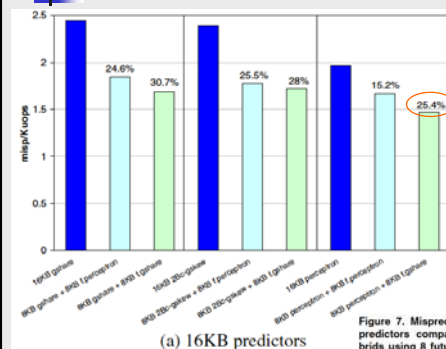


Figure 5. Effect of varying the number of future bits used by the critic on prediction accuracy for selected benchmarks. (prophet: 8KB perceptron; critic: 8KB tagged gshare)

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

23

予測精度の比較 (16KB のハードウェア資源)



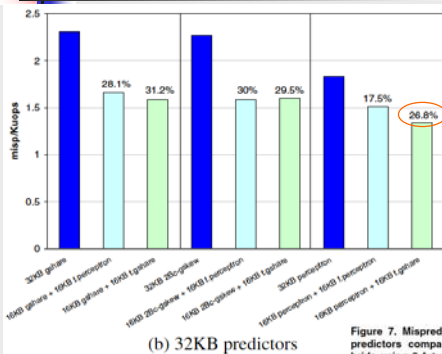
(a) 16KB predictors

Figure 7. Mispredict rates of conventional predictors compared to prophet/critic hybrids using 8 future bits. Numbers indicate percent reduction in mispredict rate.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

24

予測精度の比較 (32KBのハードウェア資源)



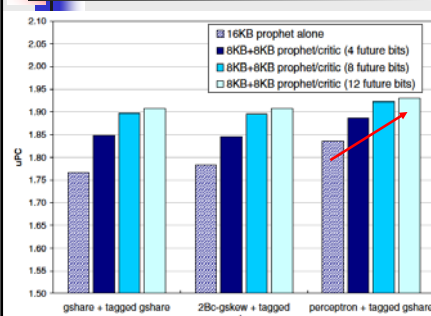
- 単体では perceptron がベスト.
- 更に、提案手法により 26.8%の予測ミスを削減

Figure 7. Mispredict rates of conventional predictors compared to prophet/critic hybrids using 8 future bits. Numbers indicate percent reduction in mispredict rate.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

25

プロセッサの性能向上の比較



- Perceptron がベスト.
- Perceptron, 12 future bitの場合には、従来手法と比較して、5.2%の性能向上

Figure 9. Average uPC of 16KB conventional predictors compared to 8KB+8KB prophet/critic hybrids using 4, 8, and 12 future bits.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

26

Prophet-Critic Branch Prediction まとめとコメント

- 今までにない、斬新なアイデアの分岐予測の提案と評価.
- 予測ミス率を大幅に低減する.
 - Gccの場合には、予測ミス率を 3.11% から 1.23%に削減することができる.
- Intel Pentium 4 プロセッサをベースとして、uPC (Uops Per Cycle) で 5.2%の性能向上 (Perceptron分岐予測の場合).
- コメント
 - アイデアとして面白い.
 - ハードウェアの規模と複雑さも問題にならないレベル.
 - アイデアは分岐予測以外の分野でも利用できるのではないか.

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

27

Completion Time Multiple Branch Prediction for Enhancing Trace Cache Performance

Ryan Rakvic et al.
Carnegie Mellon University
ISCA-2000 pp.47-58

28

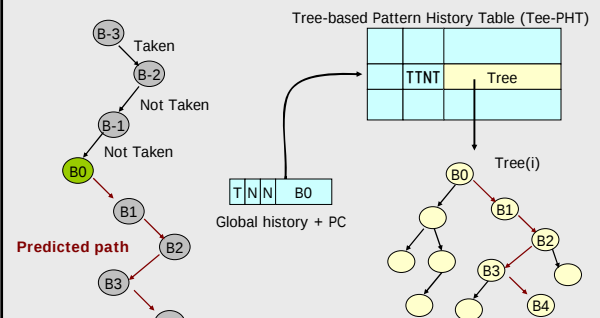
複数分岐予測 (Multiple Branch Predictor)

- これまでの分岐予測は、1サイクルに1つの分岐方向を予測
- ハードウェア資源の増加にともなって、大規模なプロセッサが実現可能
- フェッチする命令数を増やすためには、1サイクルに**複数の分岐方向を予測**することが必要
 - 複数分岐予測 (Multiple Branch Predictor)

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

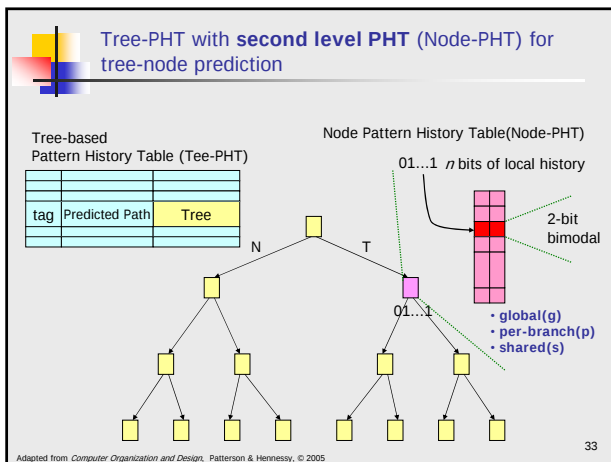
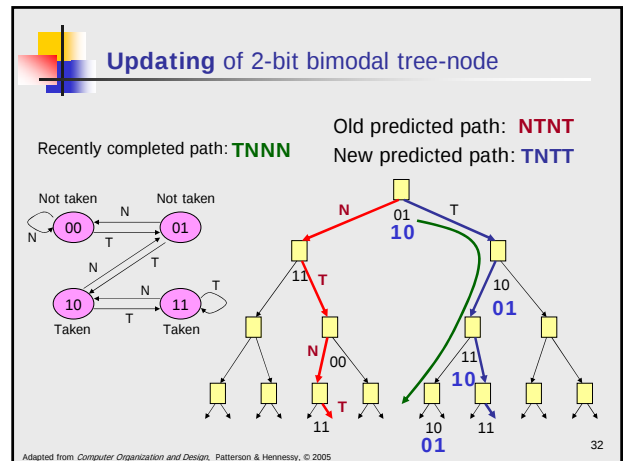
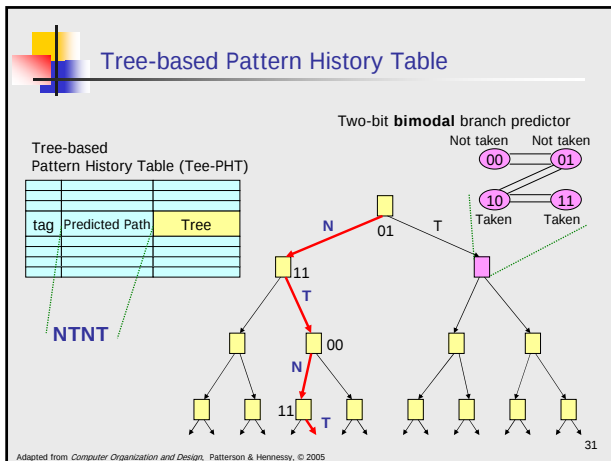
29

Tree-Based Multiple Branch Predictor (TMP)



Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005

30



アナウンス

- 講義スライド, 講義スケジュール
- www.arch.cs.titech.ac.jp

Adapted from Computer Organization and Design, Patterson & Hennessy, © 2005