

トレース・ドリブン・シミュレーションによる
分岐予測機構の検討

4P-7

中村 友洋[†], 吉瀬 謙二[†], 金指 和幸[‡], 田中 英彦[†]
[†]東京大学工学部, [‡]富士通(株)

1 はじめに

半導体技術とプロセッサアーキテクチャの改良によってマイクロプロセッサの性能は飛躍的に進歩している。特にスーパーパイプライン構造は処理を複数のステージに分けることで実効的な処理のスループットを上げ、スーパースカラ構造は命令レベルの並列性を抽出して実行する能力を持ち、最近の主流となっている。これらの技術は一連の命令列を連続もしくは同時に実行するために、分岐命令のような制御フローの変更を行なう処理によって命令のストールが生じることがあり、特に複数命令を同時に実行するスーパースカラ構造では大きなペナルティとなる。そこで分岐命令の結果を予測する機構(分岐予測機構)を採り入れてペナルティの軽減を行なうことが常識となってきている。本稿では最近のマイクロプロセッサで採用されている分岐予測機構をトレース・ドリブン・シミュレーションにより定量的に評価する。

2 目的と背景

スーパーパイプライン構造や、スーパースカラ構造をもつ場合には、分岐命令などによる制御フローの変更によって、大きなペナルティが課せられることがあるため、分岐予測機構を設け、正しい制御フローを予測し、このペナルティを軽減している。いくつかのプロセッサにおけるペナルティの大きさを表1にまとめる。表1か

プロセッサ	ペナルティ サイクル	命令数
Alpha 21064	4	8
Pentium	3	6
PowerPC604	3	12

表 1: 分岐予測ミスによるペナルティ

ら分かるように、スーパーパイプライン構造のために、1つの分岐予測ミスによって複数サイクルにわたるパイプライン・ハザードが生じ、さらに複数命令を同時に発行するためにペナルティを命令数に換算すると、さらに2~4倍となってしまふ。分岐命令は全実行コードの10~30%を占めるといわれており、命令レベルの並列性

A Study of Branch Prediction Strategies using Trace Driven Simulation

Tomohiro NAKAMURA[†], Kenji KISE[†], Kazuhiro KANAZASHI[†], Hidehiko TANAKA[†],
[†]University of Tokyo, Department of Engineering,
[‡]Fujitsu LTD.

を抽出し処理を続けるには、分岐予測機構の強化が必須となっている。本稿では多く使われている分岐履歴情報をもとにした分岐予測機構の性能を、分岐履歴情報を保存するエントリ数などをパラメータとして評価する。

3 主な分岐予測機構

最近のマイクロプロセッサで採用されている主な分岐予測機構としては次のようなものがある。

1. 常に分岐成立, 後方分岐

静的な分岐予測, 後方分岐はループによる後方分岐の成功率が高いことを利用, PA-RISCなどで採用

2. 分岐履歴情報に基づく予測

1~3-bit 程度の分岐履歴を各分岐命令に対して保存して動的に予測を行なう, PowerPC604で採用されている2-bitの分岐履歴は図1のような情報をもつ, Pentium, Alpha, PowerPCなどでも採用

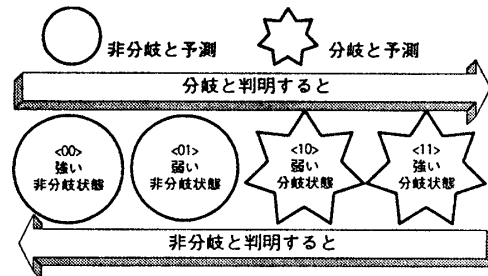


図 1: PowerPC604 の分岐履歴情報

本稿では、PowerPC604 の分岐予測機構を基本として図 2のような分岐予測機構の評価を行なう。BTAC(Branch Target Address Cache) は分岐命令のアドレスと分岐先を保存する。BTAC中にある分岐命令は分岐成立と予測される。次にBAC(Branch Always Correction)では無条件分岐命令の予測を修正する。最後にBHT(Branch History Table)では分岐履歴を保存し、これに基づいて予測を行なう。

4 分岐予測機構シミュレータ

4.1 実装方法

本研究では、SPARCシミュレータを用いてトレースデータを取り、これを分岐予測機構のシミュレータに通すことで評価を行った。コードはSPARCの命令セットであるが、シミュレート対象としたプログラムはSPARCチップ用の最適化を施さずにコンパイルした。分岐予測機構シミュレータは、図2の構成を基本としたが、BTACのエントリ数は64、BHTの履歴情報は2-bit、BHTの

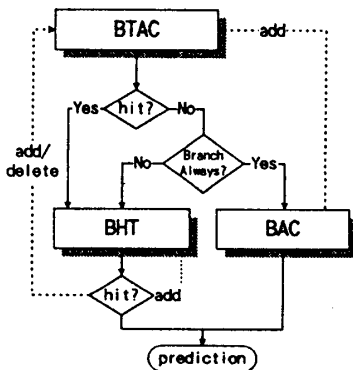


図 2: 3 ステージ分岐予測機構

エントリ数は1~4096の範囲で可変とした。またBTACやBHTのエントリが溢れた場合の置き換え戦略は、シミュレータを簡単にするため、疑似FIFO¹とした。

4.2 実験内容

SPARCシミュレータから得られるトレースデータをもとに、実際に行われた分岐命令(Bicc, CALL, CBccc, FBfcc, JMPL, RETT, Ticc)についての分岐状況を抽出し、4.1節で述べた分岐予測機構シミュレータを使って分岐予測機構の動作をトレースした。今回サンプルとして使用したプログラムは次の通り。

名称	実行命令数	分岐命令数	無条件分岐
calc	95,920 (100.00%)	21,129 (22.03%)	2,158 (2.25%)
compress	636,151 (100.00%)	90,183 (14.18%)	18,712 (2.94%)
dhystone	622,019 (100.00%)	131,936 (21.21%)	8,990 (1.45%)
jhd	667,674 (100.00%)	134,974 (20.22%)	8,978 (1.34%)
whetstone	2,116,651 (100.00%)	224,124 (10.59%)	56,073 (2.65%)

表 2: サンプルプログラムの概要

5 評価

図3は、1~4096の範囲におけるBHTエントリ数と分岐予測成功率の関係を示したものである。グラフは各サンプルプログラムの結果の平均をとっており、BACの制御がある場合とない場合を示している。プログラムのサイズにもよるが、ほぼBHTエントリ数が32~96の付近で分岐予測成功率が大きく上がることが分かる。つまりこれ以下のBHTエントリ数では、履歴情報が有効に活用される前にエントリの置換が行なわれる確率が高いということが分かる。また、BACの制御が無い場合は、有る場合に比べていずれのエントリ数においても性能が低くなっているが、これはBAC制御が無いとBHTに無条件分岐も登録することになり、それだけBHTエントリが有効に活用されにくくなるためである。図4は、

¹FAFO(First Assign First Out):エントリが割り当てられた順に置き換える。同一エントリの更新による再割り当ては考えない。

BHTエントリ数とBHTへのヒット率の関係を示したものである。BACの制御がある場合は、無条件分岐命令がBHTを使わないので、見かけ上BHTへのヒット率が低下するが、無条件分岐命令をヒットしたものと扱うとBACの制御がない場合に比べて全体的なヒット率がある。これもBACの利用によりBHTエントリがより有効に活用されるためである。

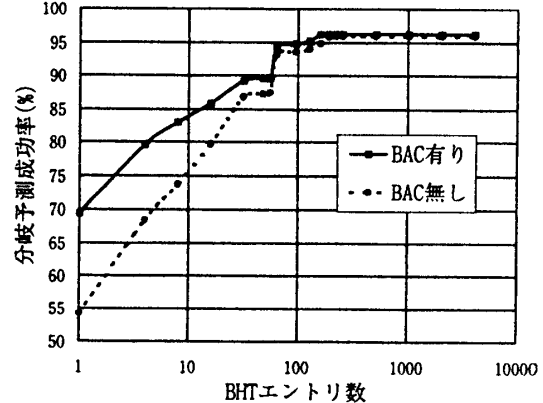


図 3: BHT エントリ数-分岐予測成功率

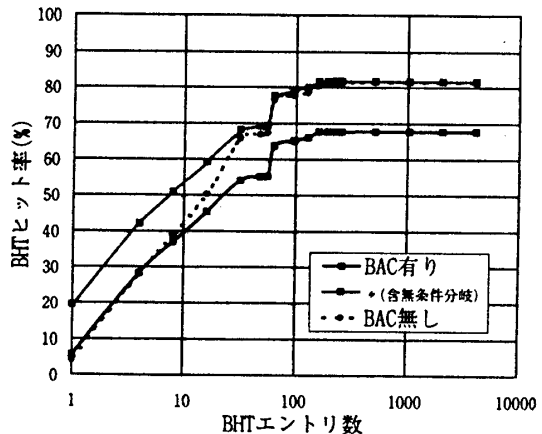


図 4: BHT エントリ数-BHT ヒット率

6 おわりに

BHTエントリ数が64前後で分岐予測成功率に大きな差があることを示した。この値はプログラムによって前後するが、少なくともBHTエントリが不足した場合にはかなりのペナルティがかかるといえる。シミュレータでは他にも多くのデータが得られており、バッファ置換戦略の評価をはじめとする各種解析が今後の課題である。

謝辞

本研究の一部は文部省科学研究費(一般研究(B)課題番号07458052「大規模データベースプロセッサの研究」)による。

参考文献

[1] Scott A. Mahlke, Richard E. Hank, Roger A. Bringham and John C. Gyllenhaal, "Characterizing the Impact of Predicated Execution on Branch Prediction", 27th International Symposium on Microarchitecture, pp.217-227, 1994
 [2] "PowerPCTM 604 RISC Microprocessor Technical Summary", Motorola Inc., 1994