

プログラム解析に基づく分岐予測機構に関する問題点の検討

3K-5

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

半導体技術の進歩により単一のプロセッサに数多くの機能ユニットを組み込めるようになってきており、スーパースカラ・プロセッサが広く使われるようになった。しかしながらスーパースカラ・プロセッサはその構成上、複数の有効な命令が同時に実行できてはじめてその機能を余すところなく発揮するものであって、分岐命令が多く含まれる汎用プログラムでは必ずしもその性能を十分に発揮できているとはいえない。このような問題点を隠蔽するため、これまでに数多くの分岐予測機構が提案され実装されてきたが、その結果既存の分岐予測機構の枠組では性能限界が見えてきた。本研究では分岐命令に関わる様々な特性を調査・解析し、いま一度分岐命令の扱い方の基本から検討すべくプログラム解析を行ない、現状の分岐予測機構における問題点に関する考察を行った。

2 目的と背景

スーパースカラ・プロセッサなどのように命令レベルの並列性を利用する場合には、その性能を十分活かすために命令実行ユニットに意味のある命令を連続的に供給することが重要である。しかし一般的にプログラム・コード中には10~30%程度の割合で分岐命令が含まれているため、分岐先を正しく予測し、その予測に従って命令をフェッチし続けなければならない。よって分岐先の予測に失敗するとその分岐命令にまで遡って正しいパスをフェッチし直さなければならない。このために実行ユニットに命令が供給できなくなり、パイプライン・ハザードが起る。特に out-of-order 機構などをもった最新のマイクロプロセッサでは、数十の命令をバッファリングするために、このペナルティーが増大する傾向にある。バッファリング・サイズの上限を決定する要因の1つが分岐予測性能にあるといえる。

表1は最近のマイクロプロセッサが out-of-order のために備える re-order バッファのサイズと同時発行できる命令数を示したものである。一般に re-order バッファのサイズが大きくなれば命令レベルの並列性を抽出しやすくなるが、分岐予測性能によるペナルティーが高くなる。例えば、PowerPC620 では最大16命令を同時に

プロセッサ	re-order バッファ数	同時命令発行数
PA-8000	56	4
Pentium Pro	40	5
K5	16	5
PowerPC620	16	4

表1: 最近のマイクロプロセッサにおける命令レベル並列性利用のためのハードウェア・サイズ

バッファリングするため、分岐命令の出現頻度を20%と仮定すると約3命令が分岐命令であることになる。分岐予測成功率が90%であると仮定した場合、この16命令がすべて正しいパスからフェッチされたものである確率は72.9%である。ところがPA-8000の場合最大56命令を同時にバッファリングするため、分岐命令の出現頻度が同じとすると、約11命令が分岐命令で、同様の確率で正しいパスをフェッチするには、97.2%もの高確率で分岐予測を成功させなければならない。このように命令レベルの並列性を利用しようとすると、必ず分岐命令によるペナルティを考慮しなければならない。

本研究ではこれまでに提案・実装されてきた分岐履歴を使った分岐予測機構について、プログラム解析に基づいて分岐命令の特性に関するデータを収集することで、その問題点を調査した。

3 分岐履歴に基づいた分岐予測機構

最近のマイクロプロセッサで採用されている分岐予測機構のほとんどは分岐履歴に基づいて予測を行なうものである。通常1~3-bit程度の分岐履歴を動的に保存し、次の分岐方向を予測するものである。

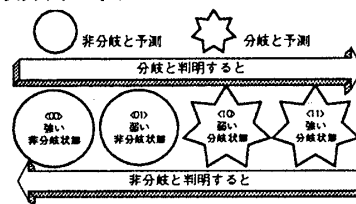


図1: PowerPC604/620 の分岐履歴情報

例えばPowerPC604/620の場合、図1のような2-bitの分岐履歴ビットを用いて予測を行なう。

4 分岐命令特性

4.1 プログラム解析の概要

本研究ではSPARCシミュレータを用いてプログラム解析を行なった。実験に使用したサンプル・プログラムは表2の通りである。

表2から、分岐命令が20%程度というかなり高い確率で存在することが分かる。また、中でも条件分岐の占める割合が相当に高く、分岐命令の70%程度が条件

A Study of Problem of Branch Prediction Strategies using Program Analysis

Tomohiro NAKAMURA[†], Kenji KISE[†], Kazuhiro KANAZASHI[‡], Hidehiko TANAKA[†],

[†]University of Tokyo, Department of Engineering, [‡]Fujitsu LTD.

名称	実行命令数	分岐命令数	条件分岐数
calc	33,647 (100.00%)	7,140 (21.22%)	5,207 (15.48%)
compress	3,000,000 (100.00%)	393,473 (13.12%)	278,099 (9.27%)
dhrystone	2,072,456 (100.00%)	436,841 (21.08%)	289,135 (13.95%)
espresso	3,000,000 (100.00%)	544,837 (18.16%)	386,713 (12.89%)
grep	101,354 (100.00%)	22,859 (22.55%)	18,159 (17.92%)
jhd	668,186 (100.00%)	135,140 (20.22%)	111,673 (16.71%)

表 2: サンプルプログラムの概要

分岐であることも分かる。つまり分岐制御、とりわけ条件分岐の制御がマイクロプロセッサの高性能化には欠かせないといえる。

4.2 分岐予測ミス原因

分岐履歴を利用した分岐予測機構の性能は BHT (Branch History Table) のサイズを十分に大きくしても 90% 程度で飽和してしまう。この原因として主に「初期分岐予測ミス」、「構造上予測ミス」の 2 つが考えられる。初期分岐ミスとは、履歴情報が揃っていないような初めて実行する分岐に対する予測ミスである。構造上ミスとは、分岐予測機構の構造上の問題 (限界) によって起こる予測ミスのことである。後者には履歴情報の不足によるミスや、履歴情報を保存するためのエンタリ不足によるものなどが含まれる。本稿では、初期分岐、後方への条件分岐パターンをプログラム解析により求めることで、これらのミスの頻度を調査し、その対策を考える。

4.3 初期分岐予測ミス

図 2 は PowerPC604 の分岐予測機構を用いて初期分岐の予測を行なった場合の予測成功率を表している。PowerPC604 の分岐予測機構では分岐履歴がない場合には分岐不成立と予測する (ただし無条件分岐は成立と予測する)。分岐全体に対する分岐予測成功率はおよそ 90% であるから、明らかに初期分岐に関しては分岐予測成功率が低い。しかし図 3 から分かるように十分 BHT エンタリがあれば初期分岐に相当する分岐は非常に少なくなり、性能への影響は小さいと考える。基本的に初期分岐は静的な分岐予測を行なうことで予測成功率をあげるのが妥当な対策である。

4.4 後方への条件分岐パターン

ループの終端には分岐命令が存在する。通常この分岐命令は後方への分岐であり、複数回分岐が成立しループを回った後で分岐が不成立となりループから抜け出る。このような分岐を PowerPC604/620 のような 2-bit の分岐履歴情報をもとにして正しく予測することは困難である。そこでまずこのような後方への条件分岐の分岐パターンを調べることでその制御方法を考える。

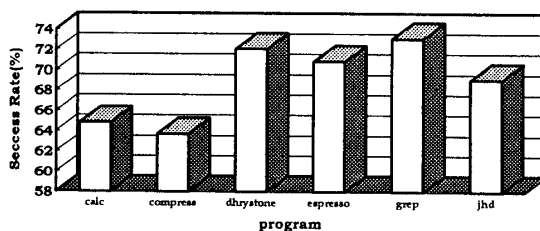


図 2: 初期分岐予測性能

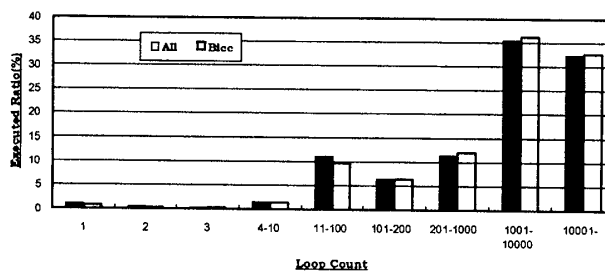


図 3: 分岐命令の繰り返し回数頻度

表 3 は後方への分岐のパターンに関する偏りを示している。ここで集中度とは、1 つの後方分岐命令が同じ回数だけループを回ってから抜け出る確率で、例えばある 1 つの後方分岐命令が、「3 回ループを回って (3 回分岐成立) から抜け出る (1 回分岐不成立)」というパターンを 9 回繰り返す、その後で「1 回もループを回らずに抜け出る」という場合には、3 回ループを回るという可能性が最も高く、その集中度は 90% となる。よって集中度が高いほどループを回る回数が毎回一定である可能性が高いといえる。表 3 によれば、いずれのプログラムでもこの集中度が 97~99% となっており、後方への条件分岐に関してはこのパターンをつかみ、予測することで大幅な性能改善が可能である。

calc	compress	dhrystone	espresso	grep	jhd
98.75	99.99	99.94	98.62	97.61	99.21

表 3: 後方分岐のパターン集中度 (%)

5 おわりに

プログラム解析に基づいて、初期分岐予測、後方への条件分岐予測に関する考察を行なった。特に後方への条件分岐はループ終端などに多く見られ、表 3 に示した特性を利用して分岐予測を行なえば、かなりの性能向上が見込める。これを効率良く実現するためのハードウェア、ならびにソフトウェア支援の検討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] Po-Yung Chang, Yale Patt, "Branch Classification: A New Mechanism for Improving Branch Predictor Performance", 27th International Symposium on Microarchitecture, pp.22-31, 1994
- [2] 中村友洋, 吉瀬謙二, 金指和幸, 田中英彦, "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", 第 51 回情報処理学会全国大会, pp.6-13-6-14, 1995