

辻 秀典, 中村 友洋, 吉瀬 謙二, 安島雄一郎, 田中 英彦
 東京大学大学院 工学系研究科

1 はじめに

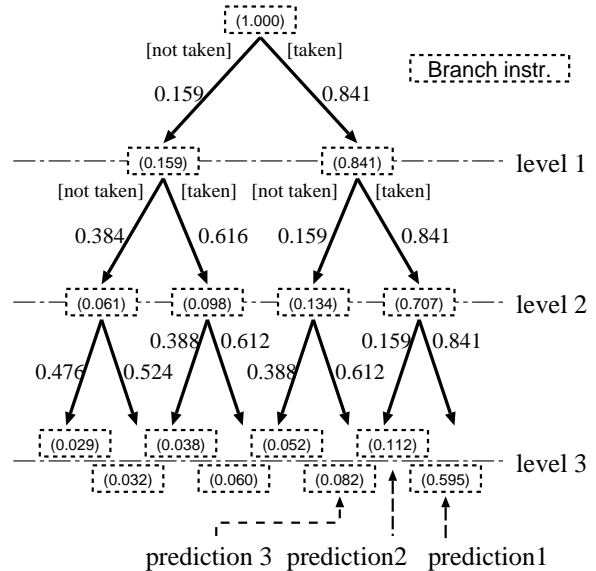
命令レベル並列性を利用したマイクロプロセッサが一般的となった現在、さらなる性能向上を求めて次世代のマイクロプロセッサに関する研究が盛んに行われている。それらは大規模な投機的処理を目指しているものが多く、そのひとつとして大規模データパス・プロセッサ [1] が提案されている。大規模な投機的処理を行うためには効率的な分岐制御が必要であるが、そのために大規模データパス・プロセッサでは、コントロールフロー・ツリーに基づきパス別の分岐確率情報を管理することによって、多段の分岐を越えた投機的フェッチを行う。

本稿では、そこで処理される分岐確率情報の伝播手法の検討とその評価を行う。

2 コントロールフロー・ツリーに基づくパス予測

通常、分岐予測といえば分岐命令毎に分岐先の予測を行う。そのため、複数の分岐命令を越えたパス予測を既存の方法を用いて行う場合には、分岐命令毎の予測を繰り返すしかない。この場合、予測できるパスは単一である。分岐予測を繰り返すことによって分岐予測の成功率は落ちることに加え、予測が外れた場合の別のパスをフェッチし直すペナルティが大きくなる。そのため、単一のパスだけを予測候補とするのではなく、複数のパスを予測候補とすることにより予測が外れた場合のペナルティを削減し予測確率低下を補う。その方法として、コントロールフロー・ツリーを構築し、これに基づき複数の分岐命令を統合的に処理してパス予測を行う。

具体的には図1に示すようなコントロールフロー・ツリーを構築する。まず、各分岐命令の分岐先すべてに重み付けを行う。図中、各矢印は分岐先候補を意味しその脇にある値がその重みである。そして、複数のその分岐先候補によって構成されるパスは、その重み値の積によって求められる確率値をもつ。図中ボックス内の値がこれに該当し、この値が大きなものから順に実行される確率の高いパスとする。このようにして予測を行った場合の予測成功率を図2に示す。



この図から分かるように、実行される確率が高いパスに偏りがあるため実際にすべての命令を展開する必要性はない [2]。そのため大規模データベース・プロセッサでは、レベルを固定した展開は行わず確率の高いパスを優先的にフェッチする形をとる。

グラフ中の直線は、同じパス予測を分岐命令毎に予測を繰り返すことによって行った場合の予測成功率である。複数の候補をあげることができないため、予測候補数が 1 の場合の値で直線としてある。ここで予測候補数が 1 の場合に注目すると、コントロールフロー・ツリーに基づく予測よりもこの単一予測の方が若干予測成功率が高いことがわかる。すべてのパスを展開して情報管理をしているにもかかわらず、若干なりとも予測成功率が低下する理由として、その情報管理、つまり確率伝播手法に問題があることがまず考えられる。

3 確率伝播手法の検討と評価

分岐命令毎の予測を繰り返す場合とコントロールフロー・ツリーに基づく予測の違いについては、前者の予測が離散的な値 (taken/not taken のみ) として扱われることによる情報のばらつき、および予測がレベル毎に区切られることの 2 点が考えられる。そのため、(1) 確率情報伝播においてレベルを考慮にいれた方法 と、(2) taken/not taken 方向の確率値の差を広げる方法の 2 手法について検討し評価を行った。

通常は 図 3 の上に示すように、分岐先の重み付けを履歴に基づいて taken/not taken の割合で割り振る。これに対して 図 3 の下で示した方法では、定数 (Const) を均等に割付けた残りを taken/not taken 方向で割り振る。(1) の方法では、レベルの小さい方向の定数を順に大きくする type 1、レベルの大きい方向の定数を順に大きくする type 2 を評価した。(2) の方法では、taken/not taken の値が近くなることが少なくなるように、指数関数による写像を用いて taken/not taken の方向の値の差を広げる type 3 を評価した。この 3 種類と従来の手法 (normal) に関して、cc1 と 合成ベンチマーク whetstone の 1000 万命令実行における予測成功率を求めた。レベルは同様に 5 とし、その結果を 図 5 に示す。

