

## 分岐予測機構を応用したループパス予測器の初期検討

金海 和宏<sup>†</sup> 大川 猛<sup>†</sup> 大津 金光<sup>†</sup> 横田 隆史<sup>†</sup> 馬場 敬信<sup>†</sup>

<sup>†</sup>宇都宮大学工学部情報工学科

### 1 はじめに

プログラムのループ中には、条件分岐の結果の組み合わせにより様々な実行処理経路 (パス) が存在する。我々は、ループ内で次のイテレーションにおいて実行されるパス (以下、ループパスと呼ぶ) を予測し、予測パスに応じて最適化したプログラムコードを投機的にマルチスレッド実行していくことでプログラム実行の高速化を実現する2パス限定投機方式 [1] を提案した。本方式での投機実行によるプログラムの高速化にはパスの予測成功率が大きく影響する。しかし、パス予測器の妥当性についての十分な検討は行われていない。本稿では高い予測成功率を目指し、2レベル適応型分岐予測器、パーセプトロン分岐予測機構 [2]、O-GEHL 分岐予測機構 [3] をループパス予測器に応用し、各予測器ごとに予測精度について評価を行う。

### 2 ループパス予測器

2パス限定投機方式では、1つのループ内において高々2本のパスの実行頻度が支配的であることを明らかにしており、上位2本のパス (実行頻度が高い順に #1パス、#2パス) を投機的にマルチスレッド実行することで実行時間の短縮を実現する。ループパス予測器はループ内の各イテレーションにおいて#パスが実行されるかどうか予測を行っているが、同じく2通りの予測を行う分岐予測機構に着目し、パスの予測に分岐予測機構を応用した際の妥当性を予測精度の面で評価する。

本稿では分岐予測機構で最も主流である2レベル適応型予測機構と、最先端の分岐予測機構の研究で多く用いられているパーセプトロン分岐予測機構、および Championship Branch Prediction (CBP)[4] で特に高い成績をおさめた O-GEHL 分岐予測機構 (それぞれ、パーセプトロンパス予測器、O-GEHL パス予測器) をループパス予測器に実装して予測精度の比較を行う。評価結果をもとに、各機構がパスの予測精度に関してどれほどの性能を達成するかを評価する。

### 2レベルパス予測器

ループパス履歴レジスタ (以下、履歴レジスタ) とカウンタテーブルで構成し、履歴レジスタをインデックスとしてカウンタテーブルを参照する。参照先の飽和カウンタの値に基づいて次に実行するパスを予測する。図1に2レベルパス予測器の構成を示す。

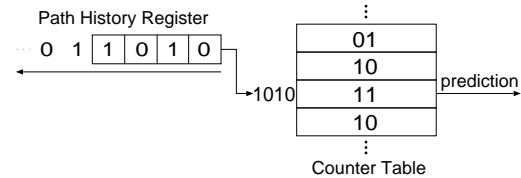


図 1: 2レベルパス予測器

履歴レジスタは、#1パスを1、それ以外のパスを0として1ビット上位シフトの後に最下位ビットへの数値の格納を繰り返すことでパスの履歴を保存する。nビット分の履歴をカウンタテーブルへのインデックスとして使用する。

カウンタテーブルの各テーブルの飽和カウンタは標準の設定である2ビットとし、閾値  $(10)_2$  を各カウンタの初期値とする。カウンタの値が閾値以上である場合には#1パス、閾値未満であればそれ以外のパスが実行される可能性が高いという予測を行う。

### パーセプトロンパス予測器

図2にパーセプトロンパス予測器の構成を示す。履歴レジスタとバイアスビット、および履歴レジスタの各ビットとバイアスビットに対応する重みを格納するパーセプトロンテーブルより構成する。各分岐命令アドレス (ループアドレス) に対応するパーセプトロンテーブルを参照し、履歴レジスタの各ビットと、対応する重みとの内積を求める。内積が0以上であれば#1パス、0未満であればそれ以外のパスが実行される可能性が高いと予測する。

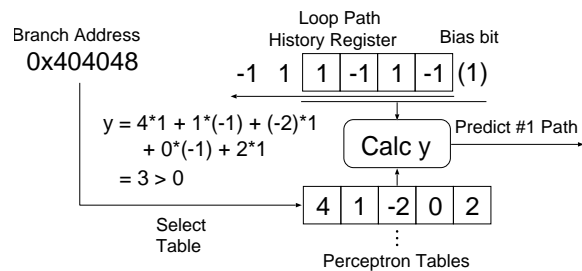


図 2: パーセプトロンパス予測器

### O-GEHL パス予測器

図3に O-GEHL パス予測器の構成を示す。実行パスの履歴を格納する履歴レジスタと、重みを格納したパーセプトロンテーブル、およびループアドレスの LSB の履歴を格納する (非ループパスの) パス履歴レジスタを用いる。予測の際には全てのパーセプトロンテーブルを参照するが、参照するパーセプトロンテーブルによって、各レジスタの LSB からの参照ビット数の範囲が異なる。各レジスタとループアドレスをもとに 3n

Preliminary Consideration of Loop Path Predictor based on Branch Prediction Mechanisms

<sup>†</sup>Kazuhiro Kinkai, Takeshi Ohkawa, Kanemitsu Ootsu, Takashi Yokota and Takanobu Baba

Department of Information Science, Faculty of Engineering, Utsunomiya University (†)

ビットのビット列を生成した後、 $n$ ビットに3等分する。分割したそれぞれのビット列内においてビットの並び替えを行い、3つのビット列を3エントリのXORを用いて $n$ ビットにハッシュし、パーセプトロンテーブルの各重みへのインデックスを生成する。インデックスをもとに各テーブルから1つずつ重みを抽出して総和を求め、総和が0以上であれば#1パス、0未満であればそれ以外のパスが実行される可能性が高いと予測する。

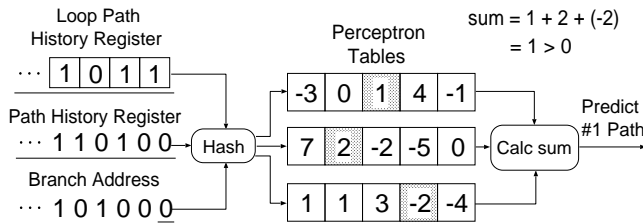


図 3: O-GEHL パス予測器

### 3 予測性能の比較

評価の対象とするループは、SPEC CINT2000の 164.bzip, 175.vpr, 176.gcc, 181.mcf, 186.crafty, 255.vortex, 256.bzip2, 300.twolf 内のループで、データセットは train を用いる。少なくとも 2 本以上のパスが出現すること、ループ中で関数の呼び出しを含まないものについて上位 5 つのループを評価の対象とする。

実行パスの履歴を格納する履歴レジスタのビット数(履歴長)を 1 ビットから 200 ビットまで変化させた評価結果を図 4 に示す。ただし、2 レベルパス予測器はハードウェア量が履歴長に対して急激に増大することから 26 ビットまでとし、O-GEHL パス予測器の(ループアドレスの LSB の履歴を格納する)パス履歴長は最大でも 16 ビットまでとする。

2 レベルパス予測器は履歴長を大きくすることで予測失敗率が低下する。履歴長をさらに大きくすることで予測失敗率がより減少する可能性があるが、ハードウェア資源量が指数関数的に増えていくため、これ以上の履歴長の増加は困難である。

パーセプトロンパス予測器は履歴長 20 ビット以降、

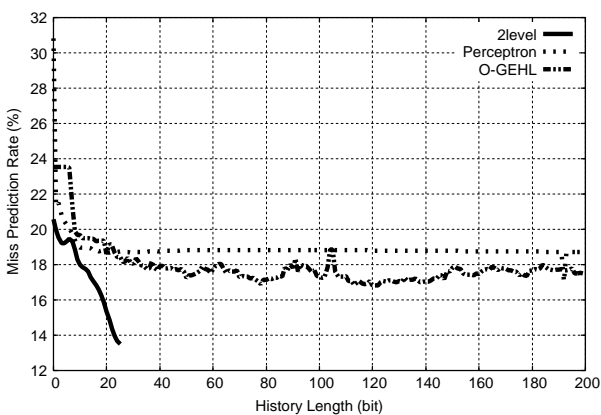


図 4: 予測失敗率

ビット長にかかわらず予測失敗率の変化が少なくなる。プログラムの処理においてパスは同じものが複数回続く傾向があり、特に #1 パスが続くことが多い。よって履歴レジスタに 1 を格納する可能性が高いため、パーセプトロンテーブルの学習により全ての重みが正に飽和することが多い。履歴長が小さい場合、出現するパスが #1 パス以外に変わりしばらくそのパスが続くと、履歴レジスタとパーセプトロンテーブル内の重みとの内積が早い段階で負となり予測先を変える。履歴長が長い場合、内積が負になるのが遅く、予測先の変化が遅れる。履歴長が一定以上になると、内積が負になる前に #1 パスが再び出現し、結果内積は正を維持する。これにより多くの処理において #1 パスを予測し続けるため、予測失敗率の変化がなくなると考えられる。

O-GEHL パス予測器は履歴ビット数を増やすと予測精度の向上が見られるが、履歴長 60 ビットを越えると予測失敗率は 17% から 19% の間となる。各履歴アドレスやループアドレスをもとにしたハッシュをインデックスとするため、特定の履歴レジスタの内容に対応する重みの値が、処理に対して最適に学習していると考えられる。しかし、処理の中で 1 つのループは複数回続くため、(非ループパスの)パス履歴レジスタは 1 か 0 のどちらかのみで満たされることが多い。このためパス履歴レジスタが有効に機能せず、予測精度の低下につながっていると考えられる。

### 4 おわりに

本稿では、ループパス予測器の予測成功率の向上を目指し、3 つのループパス予測器の評価を行った。ハードウェア資源量を考慮せず予測精度に関して評価を行った結果、2 レベルパス予測器にて履歴ビット数 26 ビットのときに最小の予測失敗率 13.5% となった。

今後は、ハードウェア資源量を考慮に入れた予測性能の比較や、予測にかかるレイテンシについて評価を行う予定である。

#### 謝辞

本研究は、一部日本学術振興会科学研究費補助金(基盤研究(C)21500050, 同(C)21500049)の援助による。

#### 参考文献

- [1] 横田 隆史ほか: “2 パス限定投機方式の提案”, 情報処理学会論文誌: コンピューティングシステム, Vol.46, No.SIG 16 (ACS-12), pp.1-13, 2005.
- [2] Daniel A. Jiménez and Calvin Lin: “Dynamic Branch Prediction with Perceptrons”, Proc. 7th International Symposium on High-Performance Computer Architecture, pp.197-206, 2001.
- [3] André Sez nec: “Genesis of the O-GEHL Branch Predictor”, The Journal of Instruction-Level Parallelism Vol.7, 2005.
- [4] “The 1st JILP Championship Branch Prediction Competition”, <http://www.jilp.org/cbp/>, 2004.