C-005

学習テーブルを用いた値予測の効率化
Efficiency improvement of Value Prediction with Learning-Table

下村 佳生 † 小林 良太郎 †
Yoshio Shimomura Ryotaro Kobayashi

1. はじめに

近年，コンピュータの性能向上のために，周波数を増加させるのではなく，並列性を上げることによって処理能力を向上させることが主流となっている．並列性の向上にはスレッド単位から処理単位まで様々な点で議論されているが，最終的には命令レベルでの依存関係が問題となってくる．その中の一つとして真の依存と呼ばれる状態がある，これは前の実行結果がないと次の命令が実行できない状態で，この状態のものは通常，2 つの命令を並列に実行することができない．

この問題を解決する技術として値予測という手法が使われている．これは過去の結果を基に実行結果を予測するもので，それにより依存関係にある命令を投機的に実行することが可能となる．予測に成功すれば実行速度の向上が見込めるが，失敗した場合には先行して行った命令は無駄になり，電力等のペナルティを受ける．そのため，予測はなるべく精度の良いものを使用したいが，精度の良いものは機構が複雑になり，チップ面積などのハードウェアコストや実行時の電力コストが増加してしまう．

本稿では従来の値予測機構に学習用の単純な機構を取り入れることで，単純ながら高い予測精度が実現できることを示す．

2. 値予測

値予測は過去の値が再び使われる可能性が高いという値の局所性を基に行われる [1]．値予測の対象となりうる，値を生成する命令の結果を値履歴表 (Value Histery Table:VHT) に保存し，次に同じ命令が実行された時に VHT を用いて予測を行う．VHT にはキャッシュと同じように，タグとその命令の実行結果が保存される．また，予測の種類によっては実行結果以外に，前回の実行結果との差分であるストライドと状態を付与したものや，実行結果を複数保存し，出現パターンを調べることで高い予測精度が得られるように工夫されたものもある [2]．もちろん保存する情報が多いほど精度は向上するが，同じエントリ数でも VHT のサイズが増え，機構自体も複雑になるためハードウェアコストが増加する．

以下に主な予測方法を述べる．

- **最終値予測**
  前回の実行結果を予測値として用いる．仕組みが単純なのでハードウェアコストは小さいが，予測精度は良くない．
- **ストライド値予測**
  ループ変数などのようにある一定の値ずつ増加する変数の予測に対応したもの．一回前の実行結果と今回の実行結果の差分をストライドとして保存しておき，ストライドが一致すると予測を開始する．ストライドが検出された場合にのみ予測を行うので予測精度が高くなるが，予測可能な状態になるまでに同じ命令を複数実行する必要がある．なお，最終値予測で予測できる値はストライド値予測でも予測可能である．
- **コンテキスト・ベース値予測**
  過去に使われた値を複数保存しておき，値の出現パターンを元に予測を行う．特定の出現パターンを持つものについては完璧に予測を行うことができ，高い予測精度を持つ．その反面，出現パターンを補える十分な大きさの VHT が必要になることに加えて，機構自体が複雑であり，ハードウェアコストも非常に大きくなる．
- **ハイブリット値予測**
  上記の予測手法を組み合わせたもの．値の局所性のあるものはコンテキスト・ベース予測，ストライドが検出されたものについてはストライド予測といった具合に場合により予測方法を変更する．

3. 提案手法

プログラム中で生成される値には局所性があり，過去と同じ値が生成されやすいことや，単純増加する変数が存在することを考慮すれば，値の予測が可能なことは先に述べた通りである．しかしそれとは別に，実行ごとにアトランダムな値を返す予測困難な命令も一定数存在する．このような命令の結果を VHT に登録することは電力的に無駄があるばかりでなく，同じエントリ箇所にある予測の容易な命令の結果を VHT 上から追い出す可能性すらある．

そこで我々は，予測の困難な命令を VHT に登録しないために学習テーブル (Learning-VHT:L-VHT) の使用を提案する．この機構は VHT に値を登録する前に試

† 豊橋技術科学大学大学院工学研究科
Graduate School of Engineering, Toyohashi University of Technology

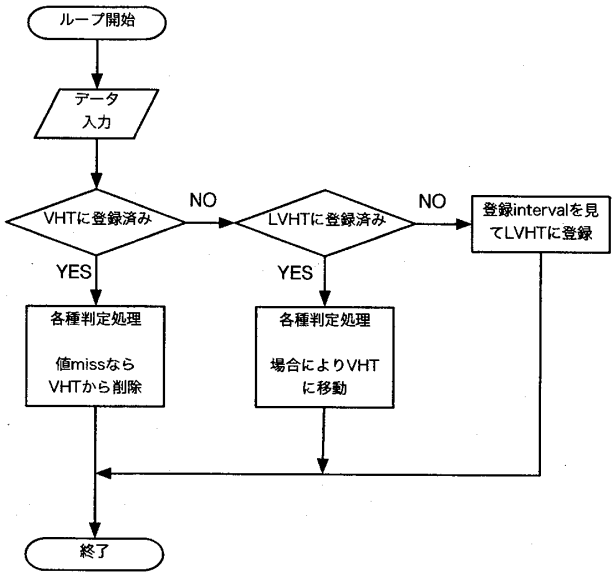

図 1　L-VHT 併用時の動作

験的に予測するためのもので，L-VHT 上で予測が成功した場合に初めて VHT 上に値が登録される．これにより，予測の困難な命令が VHT に登録されることを防ぐことができ，結果として VHT の電力削減と予測ヒット率の向上という 2 つの効果が期待できる．

ここで L-VHT 自体の大きさが問題となってくるが，プログラムは普通，短いブロックを繰り返し実行するのが普通であり，L-VHT のサイズは 32 エントリ程度の大きさでも十分であると考えられる．

本機構を用いたフローチャートを図 1 に示す．VHT にエントリがない命令の場合，通常なら VHT に命令の結果を登録するが，提案手法では L-VHT に登録を行う．この時，プログラムのループの長さが L-VHT のエントリ数よりも長ければ L-VHT のエントリが枯渇してしまい，一度も予測できなくなる可能性があるため，L-VHT には一定のインターバルを設ける．L-VHT にすでに登録されている場合は最終値予測やストライド値予測など，通常通り予測を行う．L-VHT 上で予測が成功した場合，エントリを VHT に移す．

VHT 上にエントリされていた場合は通常通り予測を行う．ただし，予測が外れた場合は VHT 上からエントリを削除する．これは予測困難な命令が偶々 VHT に登録され，全体のヒット率を下げないための配慮である．

4.　評価

ベンチマークプログラムとして SPECint2000 を用い，SimpleScalar Tool Set[3] を利用して bzip2, gcc, gzip, mcf, parser, perl, vortex, vpr について予測率およびヒット率を評価した．予測率は値予測の対象となる命令のうち，予測を行った命令の割合であり，ヒット率は予測を行った命令のうち，予測が成功した命令の割合である．表 1 に，VHT が 4K エントリ，L-VHT が 32 エントリの場合の評価結果をベンチマーク平均で示す．

表 1　計測結果

予測方式	予測率	ヒット率
最終値予測	94.20%	43.02%
最終値予測 (L-VHT)	19.09%	89.79%
ストライド予測	40.89%	90.86%
ストライド予測 (L-VHT)	19.05%	92.22%

表より，L-VHT を使用した場合は予測率が低下し，ヒット率が上昇していることが分かる．これは提案手法の導入により，実行するたびに生成する値の異なる，予測の困難な命令が VHT に登録される頻度を削減できたからであると考えられる．

5.　まとめ

L-VHT という単純な機構で予測率の低い命令を VHT 上から除去できたと考えられる．これにより，VHT 上に比較的ヒット率の高いもののみが残る結果となり，予測率こそ低下したが，全体のヒット率を上昇させることができるようになった．成果として，一番単純な機構である最終値予測でストライド予測と同程度のヒット率を得ることができた．

電力効率の評価は今後の課題であるが，L-VHT によって VHT へのアクセス頻度が削減できることから，従来の VHT よりも電力効率が改善することが期待される．

謝辞

本研究の一部は，文部科学省科学研究費補助金若手研究 (B) 課題番号 21700057，柏森情報科学振興財団研究助成，及び，中山隼雄科学技術文化財団研究助成の支援により行った．

参考文献

[1] M. Lipasti and J. P. Shen, “Exceeding the Dataflow Limit via Value Prediction,” In *Proc. Seventh ASPLOS*, pp.138-147, October 1996.

[2] K. Wang and M. Franklin, “Highly Accurate Data Value Prediction using Hybrid Predictors,” In *Proc. 30th MICRO*, pp.281-290, December 1997.

[3] T. Austin, E. Larson, and D. Ernst, “SimpleScalar: An Infrastructure for Computer System Modeling,” *IEEE Computer*, Vol.35, No.2, pp.59-67, February 2002.