

頻繁な値の局所性を考慮したデータ値予測機構のハードウェア量削減

佐藤 寿倫 有田 五次郎

九州工業大学 情報工学部 知能情報工学科

{tsato,arita}@ai.kyutech.ac.jp

<http://www.mickey.ai.kyutech.ac.jp/~tsato/cosmos/>

あらまし 近年値予測を用いたデータ依存の投機的実行が注目されているが、値予測のためのハードウェア量が問題となっている。本稿では、値予測機構のハードウェア量を削減することを検討している。具体的には、頻繁な値の局所性に着目し、予測値を0だけに限定している。SPECint95 ベンチマークではレジスタに書き込まれる値の平均で15%が0であるので、予測値を制限しても有意義なパフォーマンスが得られると予想される。シミュレーションの結果、提案する予測器は、ハードウェア規模が約4倍の最終値型予測器と匹敵することが確認された。

キーワード 命令レベル並列処理, データ投機実行, 値予測, 頻繁な値の局所性, ハードウェア実装効率

Reducing Hardware Budget of Data Value Predictors by Exploiting Frequent Value Locality

TOSHINORI SATO ITSUJIRO ARITA

KYUSHU INSTITUTE OF TECHNOLOGY

{tsato,arita}@ai.kyutech.ac.jp

<http://www.mickey.ai.kyutech.ac.jp/~tsato/cosmos/>

Abstract Recently, the practice of speculation in resolving data dependences has been studied as a means of extracting more instruction level parallelism (ILP). An outcome of an instruction is predicted by value predictors. The instruction and its dependent instructions can be executed simultaneously, thereby exploiting ILP aggressively. One of the serious hurdles for realizing data speculation is huge hardware budget of the predictors. In this paper, we propose a technique reducing the budget significantly by exploiting frequent value locality. Based on the proposal, we evaluate a value predictor, named *zero-value predictor*, which generates only value 0. Simulation results show that the zero-value predictor has better performance than the last-value predictor which is twice as large in the hardware budget as the zero-value predictor. Therefore, the zero-value predictor is one of the cost-effective and practical value predictors which can be implemented in real microprocessors.

key words instruction level parallelism, data speculation, value prediction, frequent value locality, hardware efficiency

1. はじめに

現在主流のマイクロプロセッサは性能向上を命令レベルの並列性 (ILP: instruction level parallelism) の抽出に頼っている。しかしながら、ILP は命令間の依存関係によって制限される。それらは、制御依存、資源競合、データ依存の3つに分類できる。制御依存と資源競合を取り除く方法には既に多くの研究があるが、これまでデータ依存は ILP を制限する深刻なボトルネックと見做されてきた。近年、より多くの ILP を抽出するためにデータ依存を投機的に解消する方式が注目されている。ある命令の結果は値予測機構を用いて予測され、この命令と依存関係にある命令を同時に実行できる。したがって、ILP を積極的に抽出できるわけである。

ここで注意しなければならないのは、間違った投機によるペナルティを被ること無く値予測を有効に利用するためには、高い予測精度が必要になるということである。ハイブリッド値予測機構¹⁴⁾やコンテキスト型値予測機構¹²⁾は非常に高い予測精度を達成できるが、ハードウェアコストも非常に大きくなってしまふ。つまり、値予測機構の膨大なハードウェア量がデータ投機実行の現実性を妨げているわけである。ハードウェアコストを軽減するためには、以下の方策が考えられる。

- (1) シンプルな回路
- (2) エントリ数の削減
- (3) タグアレイサイズの削減
- (4) データアレイサイズの削減

容易にわかるようにシンプルな回路はハードウェア量が小さい。例えば、ダイレクトマップ方式のキャッシュは連想度の大きなキャッシュよりもずいぶんハードウェア規模が小さくなる。値予測表のエントリ数を削減すればそのハードウェア量が小さくなるのは明白である。タグアレイサイズはタグのビット幅を制限することで削減できる。値予測ではキャッシュとは異なり、必ずしも正しい値を供給する必要は無い。間違っただけのための訂正機能が備わっているからである。データアレイサイズの削減は、値予測表に保持されるデータ幅を制限することで達成できる。Brooks ら¹⁾によれば SPECint95 ベンチマークに現われるデータの半数以上が 16 ビット以下であるので、下位ビットだけを保持することが可能である。近年、これらを考慮した値予測機構の削減方法が検討されている^{2)~4),6),10),11),13),16),17)}。本稿ではデータアレイサイズを削減する新しい方式を提案する。

2. 関連研究

値予測に基づいたデータ投機実行^{7),9)}は、命令間の依存関係を緩和し命令レベルの並列性を抽出する新たなパラダイムとして注目されている。これまでに様々な予測機構が提案されているが、比較的予測精度の高い 2 レベル型予測機構¹⁴⁾、ハイブリッド型予測機構¹⁴⁾、そしてコンテキスト型予測機構¹²⁾はハードウェア量が非常に大きくなるという問題がある。

Morancho ら¹⁰⁾、Rychlik ら¹¹⁾、Del Pino ら⁴⁾や Calder ら³⁾は値予測機構のハードウェア量を削減することを検討している。Morancho ら¹⁰⁾、Rychlik ら¹¹⁾と Del Pino ら⁴⁾の方式では、演算結果の予測容易性に基づいて命令を分類し、予測容易な命令はハードウェア量の小さな予測機構を使用し、予測困難な命令だけ上述のハードウェア量の大きな予測機構を使用する。つまり、ハイブリッド値予測機構のハードウェア量を削減に着目している。しかし、構成要素となる各値予測機構のハードウェア量は削減できてはいない。Calder ら³⁾は予測可能性に着目して、値予測表に保持される命令を選択している。データ投機実行に有効な命令だけを選択することで値予測表の利用効率を改善し、その結果エントリ数の削減を達成している。一方、Fu ら⁶⁾と Tullsen ら¹³⁾は、コンパイラの支援によって予測機構のハードウェアを完全に取り除くことを検討している。

分岐予測器においてタグのビット幅を制限することは、Fagin⁵⁾らによって検討されている。Fagin らはタグ部のビット幅と予測精度との間のトレードオフを調査し、タグのビット幅を 2 ビットに制限しても予測精度はほとんど低下しないと報告している。われわれはこの方式を値予測に適用し、最終値型予測器ではタグは全く必要無いこと、ハイブリッド型予測器でもタグは 2 ビットあれば十分であることを確認している¹⁷⁾。

最近、我々は値予測表に保持されるデータ幅を制限することで、

値予測機構のハードウェア規模を削減することを検討した¹⁶⁾。Brooks ら¹⁾によれば、SPECint95 ベンチマークで現われるデータの半数以上が 16 ビット以下である。つまり、値予測表の上位ビットはほとんど利用されていない。このことから、我々はデータの下位ビットのみを値予測表に保持することを検討し、ハードウェア量を最大 45.1%削減することに成功している¹⁶⁾。Burtscher ら²⁾は、上位ビットを複数のデータのデータで共有することで、同様の効果を得る方式を提案している。

3. 頻繁な値の局所性を利用した値予測器

頻繁な値の局所性 (frequent value locality)¹⁵⁾は新しい局所性であり、従来の値の局所性⁹⁾とは以下の点で異なる。すなわち、後者があるひとつの命令において定義されるのに対して、前者は複数の命令間で観測される。本節では、頻繁な値の局所性がどのように観測されるかを示し、これに基づいたハードウェア削減方式を提案する。評価環境の詳細は 16), 17) を参照されたい。

3.1 頻繁な値の局所性

Zhang ら¹⁵⁾によると、プログラム実行中のある時点においては、メモリアクセスで参照されるデータの大部分が、少数の値の集合に集中していることが観測されている。SPECint95 ベンチマークにおいては、中でも 0 と 1 が上位を占める場合が多い。われわれは彼らの研究を更に進め、レジスタに値を書き込む全ての命令について頻繁な値の局所性を調査した。シミュレーション結果を図 1 にまとめる。各棒グラフは二つの部分から成っており、下部 (黒) は実行結果が 0 であった命令の割合であり、上部 (灰) は実行結果が 1 であった命令の割合である。平均して 15% の命令が 0 を生成していることがわかる。これはメモリアクセスの場合の局所性と比較するとかなり小さな値であるが、それでも考慮するには十分な割合である。以下では、この 0 値の局所性を利用して、データ値予測器のハードウェア量削減を検討する。

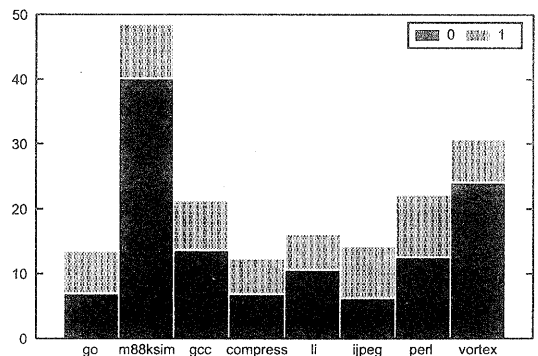


図 1 (%) 頻繁な値の局所性

3.2 0 値予測器

0 が最もよく生成される値であるのならば、0 だけを予測する値予測器であっても意味のあるパフォーマンスが得られる可能性がある。われわれはそのような、0 だけを予測する 0 値予測器 (zero-value predictor) を提案する。0 値予測器は最終値型値予測器の値予測表 (VHT: Value History Table) からデータフィールドを取り除くことで実現できる。つまり、信頼性フィールドとタグから構成される。したがって、ハードウェア規模が大幅に削減できる。図 1 を見ると 1 も大きな局所性を持つことが分かるが、0 と 1 とを予測する 0/1 値予測器は将来の検討課題とする。なぜなら、そのような拡張はハードウェア規模を拡大するとともに、値の入れ換えなどの複雑な問題が生じるからである。

☆ 4.1.2 で確認されるように、タグは必ずしも必要無い。

4. 評価

本節でシミュレーション結果を紹介する。まず、予測可能性(predictability)を評価する。予測可能性とは、予測対象となる全ての命令の中で、予測の成否に関係なく予測可能だった命令の割合を示している。また、予測対象命令の中で、正しく予測できた命令だけの割合を予測率(prediction coverage)と定義する。一方、予測精度(prediction accuracy)は、予測可能だった命令の中で正しく予測された命令の割合であるが、以下の式により求めることができる。

$$(\text{Prediction accuracy}) = \frac{(\text{Prediction coverage})}{(\text{Predictability})}$$

これらの評価に続いて、データ投機実行を用いた際のプロセッサ性能を評価する。

4.1 予測可能性

予測可能性と予測精度とはトレードオフの関係にある。予測可能性を上げるために全ての命令を投機の対象とすると、予測が困難な命令も含まれてしまうために、予測精度は低下してしまう。したがって、予測が容易な命令だけを注意して選択する必要がある。0値予測器においては、0しか予測しないために、元々予測可能性が小さい。つまり、最適なトレードオフポイントを見つけ出すことが非常に困難になる。そこで、まず信頼性の機構がどのように予測可能性に影響を及ぼすかを調査する。信頼性の機構を決定した後、予測器の容量と予測可能性の関係を調べる。予測可能性の評価には命令レベルのシミュレータを用いていることに注意されたい。

4.1.1 信頼性機構

信頼性機構には、2ビットと3ビットの飽和型カウンタを評価する。予測を行うか否かを決定する閾値には、それぞれ1~3、1~7を対象にする。図2に調査結果をまとめる。ここでVHTの容量には制限がないことを確認されたい。各プログラムの結果で、左端のグラフは予測可能性の限界、すなわち3.1節で調べた0値の局所性を持つ命令の割合、を表している。残りのグラフが、閾値を変化させた場合の予測可能性と予測率を表している。各シミュレーション結果は、信頼性機構のカウンタのビット長と閾値の組をラベルとして示されている。例えば3b:5のラベルは、閾値5の3ビットカウンタを示している。各棒グラフは二つの部分に分けられており、下部(黒)が正しく予測された命令の割合であり、上部(灰)が間違っただけの命令の割合である。すなわち、下部が予測率を、両者の和が予測可能性を表している。

まず、2ビットと3ビットのカウンタでは結果に大差が無いことがわかる。したがって、ハードウェア規模の観点から2ビットカウンタを選択する。つづいて、閾値が1の場合では予測失敗の割合が大きく、一方閾値が3の場合では予測可能性が制約を受けることがわかる。これらから、容量2の2ビットカウンタを用いることにする。

4.1.2 容量

つづいて、0値予測器のエントリ数が予測可能性に与える影響を調べる。図3にシミュレーション結果をまとめる。各プログラムで、左端の棒グラフは容量が無制限、すなわち図2で得られた結果である。残りのグラフは二つのグループから成り、それぞれ左から順に、1K、2K、4K、8K、16K、そして32K-エントリのVHTの場合である。図2と同様に各グラフは二つの部分に分けられる。左のグループはタグを持つ0値予測器の結果であり、右のグループはタグを持たない場合の結果である。0値予測器は最終値型予測器の特殊な実装と考えることが出来るので、タグは必要無いと予想できる。また、タグが必要無ければ、ハードウェア規模をさらに削減できることになる。タグビット幅を制限すると命令間の競合(aliasing)が増加する。すなわち、複数の命令が同一のエントリを共有することになる。一般的に競合には建設的(constructive)、破壊的(destructive)、そして無害(neutral)の場合があるが、0値予測器は0しか予測しないため破壊的な競合は殆ど無いと予想される。更に建設的競合を増やすためには、bi-mode分岐予測器⁸⁾にならって、VHTを高い信頼性を持つ命令のためのVHTと低い信頼性を持つ命令のためのVHTとに分割することが考えられるが、この検討は将来の課題とする。

まず、予想されたようにタグを取り除くと予測可能性が若干向上することが確認できる。また、VHTのエントリは8Kあればほぼ無限大の場合と同じパフォーマンスが得られることが観測される。これらの結果から以下では、8K-エントリのタグを持たない0値予測器を用いることにする。このハードウェア規模は2Kバイト相当であり、256-エントリのタグを持つ最終値型値予測器あるいは512-エントリのタグを持たない最終値型値予測器のハードウェア量とほぼ同等である。更に付け加えると、現在のマイクロプロセッサが持つオンチップキャッシュと比較するとその2%程度の規模にすぎない。

4.2 プロセッサ性能

本節で、8K-エントリのタグを持たない0値予測器が、プロセッサ性能にどの程度貢献できるかを調べる。評価には1サイクル当たりの完了命令数(IPC:committed instructions per cycle)を用いた。カウントされる命令にはnop命令を含んでいない。プロセッサの性能向上は、増加したIPCをデータ予測を行なわない基本モデルのIPCで割った増加率で示す。続いて、0値予測器を最終値型予測器と比較する。プロセッサ性能の評価には、サイクルレベルのタイミングシミュレータを用いていることに注意されたい。値予測に失敗した場合には、正しい実行を保証するために、間違っただけの予測値を用いて投機的に実行された命令を、正しいオペランドで再実行する必要がある。このための回復機構には、投機に失敗した命令とデータ依存の関係にある命令だけを選択的に再実行する方法を選択した。再実行する時には、予測値と正しい値とを比較するために必要な1サイクルの比較ステージがペナルティとなる。

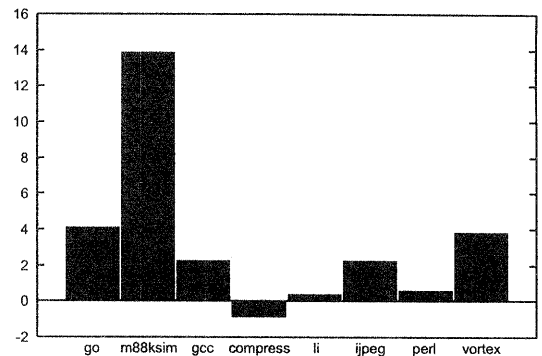


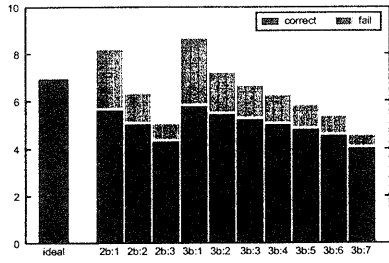
図4 (%)性能向上率

図4にプロセッサ性能向上率を示す。129.compressを除いて性能向上が確認できる。特に124.m88ksimでは13.9%もの向上が達成できた。平均では3.4%の改善である。プロセッサ性能向上率の傾向は、図1で調査した0値の局所性の傾向と類似していることが分かる。0値予測器は0だけしか予測できないにも関わらずプロセッサ性能に貢献していることとあわせると、0値の局所性はILPを抽出するために有効であると言える。

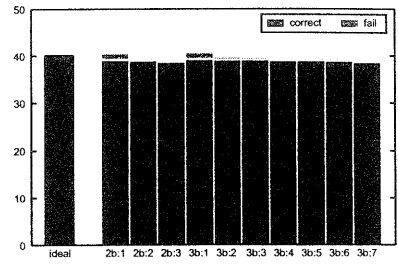
0値予測器は0しか予測しないので、予測値(常に0)と正しい値を比較するための回路は簡単に出来る。そのため再実行時に被る1サイクルのペナルティを取り除くことも可能である。このような仮定の下で性能向上率を調べた結果が図5である。平均して4.5%の改善となる。

つづいて、0値予測器を最終値型値予測器と比較する。最終値型値予測器は現状で最もシンプルな機構なので、ハードウェアコストの点で比較することに意味がある。表1にまとめられた3つのケースを比較対象に選んだ。表には各予測器のハードウェア規模も合わせてまとめている。どちらの予測器もタグを持たないことに注意されたい。表から、8K-エントリの0値予測器は512-エントリの最終値型値予測器とハードウェア規模がほぼ同じとなることがわかる。

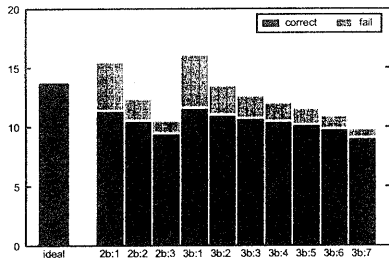
各場合のプロセッサ性能向上率を図6にまとめる。各プログラムに対して左から順に、8K-エントリの0値予測器、512、1K、2K-エントリの最終値型値予測器の結果である。投機の失敗時には、最終値型値予測器は1サイクルのペナルティを被るが、0



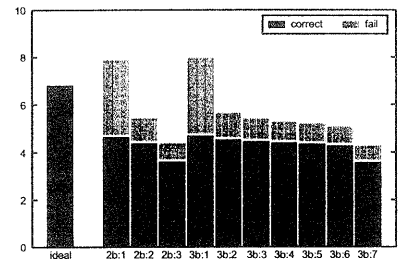
(i) 099.go



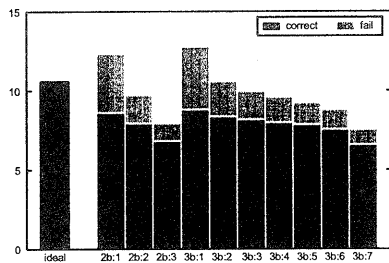
(ii) 124.m88ksim



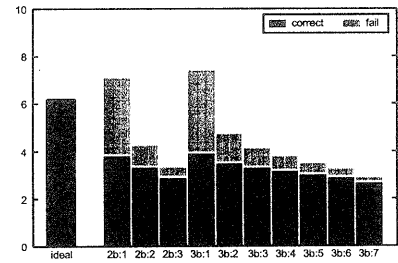
(iii) 126.gcc



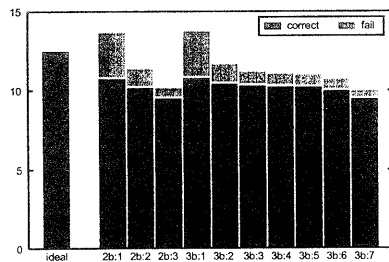
(iv) 129.compress



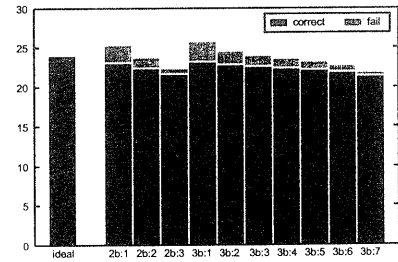
(v) 130.li



(vi) 132.jpeg

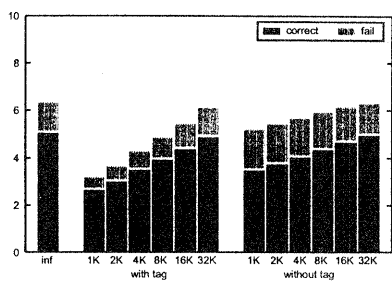


(vii) 134.perl

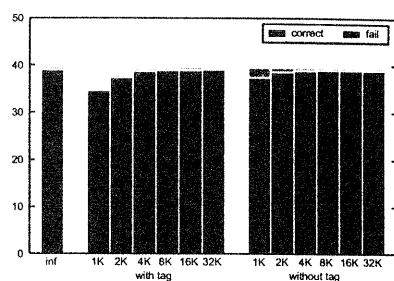


(viii) 147.vortex

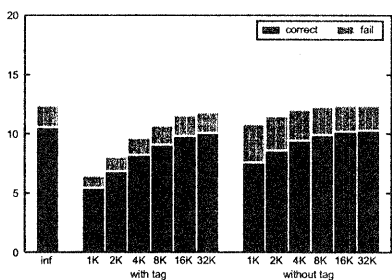
図 2 信頼性と予測可能性との関係



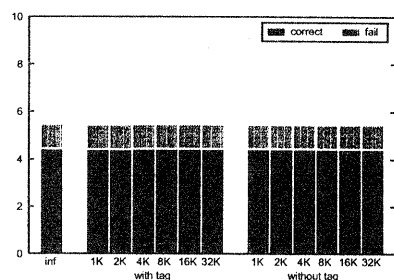
(i) 099.go



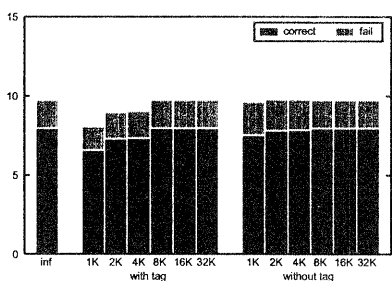
(ii) 124.m8ksim



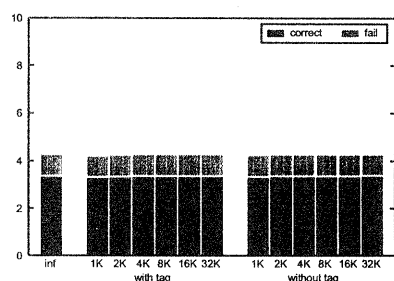
(iii) 126.gcc



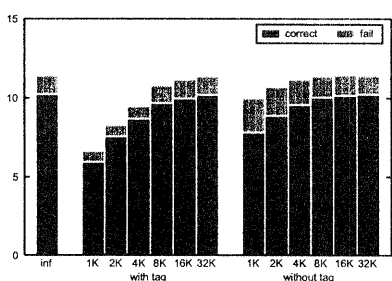
(iv) 129.compress



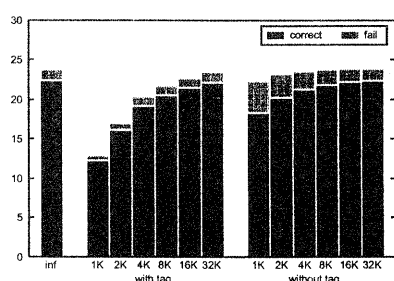
(v) 130.li



(vi) 132.jpeg



(vii) 134.perl



(viii) 147.vortex

図3 エントリ数と予測可能性との関係

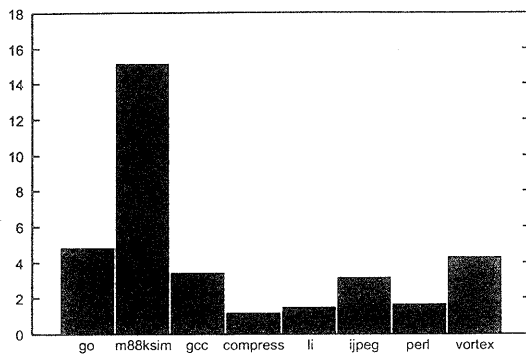


図 5 (%) 性能向上率 (比較ペナルティ無し)

表 1 予測器のハードウェア規模

predictor type	capacity (entries)	cost (bytes)
zero-value	8K	2K
last-value	512	2K
	1K	4K
	2K	8K

値予測器にはそのペナルティが無いことに注意されたい。容易に分かるように、0 値予測器はハードウェアコストが倍の 1K-エントリの最終値型よりも性能向上率が高い。さらに多くのプログラムの場合で、2K-エントリの最終値型と匹敵した貢献度を示している。特に 124.m88ksim で優位性が際だっているが、これは最終値型値予測器の場合にエントリ数が小さいためにデータの入れ換えが頻発し、0 値の局所性を有効利用できていないためと考えられる。以上から、0 値予測器のコストパフォーマンスの高さが確認できた。

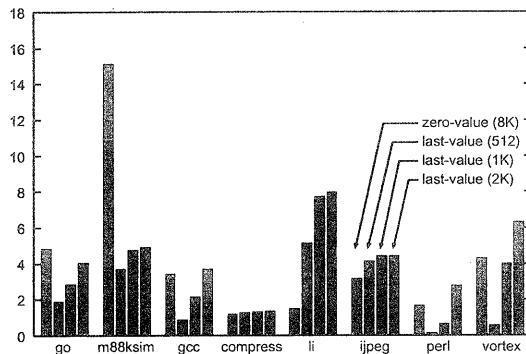


図 6 (%) 最終値型との比較

5. ま と め

本稿では値予測器のハードウェア規模を削減するために、0 値の局所性を利用した 0 値予測器を提案し、詳細なシミュレーションにより評価した。その結果、0 値予測器はハードウェアコストが倍の 1K-エントリの最終値型よりも性能向上率が高く、2K-エントリの最終値型と匹敵した貢献度を示す場合も多いことが確認できた。したがって、0 値予測器はコストパフォーマンスが高いと言える。

データ投機実行の今後の展開としては、フォールトトレラントへの応用を検討している。データ値予測機構には、予測値と実際の値を比較するための比較器と、演算を再実行するための命令再発行機構が備えられているので、フォールトトレラントへの拡張を低コストで実現できる。

本稿は九州工業大学で現在進行中である高性能低消費電力マ

イクロプロセッサ COSMOS の検討における一つの成果です。Web ページからは追加情報が入手可能です。

謝 辞

本研究の一部は、科学研究費補助金 奨励研究 (A) 課題番号 12780273、福岡県産業・科学技術振興財団 テーマ探索・シーズ発掘事業の援助によるものです。

参 考 文 献

- 1) D.Brooks, M.Martonosi, "Dynamically exploiting narrow width operands to improve processor power and performance", *5th International Symposium on High Performance Computer Architecture*, 1999.
- 2) M.Burtscher, B.G.Zorn, "Hybridizing and coalescing load value predictor", *Technical Report CU-CS-903-00*, Department of Computer Science, University of Colorado at Boulder, 2000.
- 3) B.Calder, G.Reinman, D.M.Tullsen, "Selective value prediction", *26th International Symposium on Computer Architecture*, 1999.
- 4) S.Del Pino, L.Pinuel, R.A.Moreno, F.Tirado, "Value prediction as a cost-effective solution to improve embedded processor performance", *3rd International Meeting on Vector and Parallel Processing*, 2000.
- 5) B.Fagin, "Partial resolution in branch target buffers", *IEEE Transactions on Computers*, vol.46, no.10, 1997.
- 6) C-y. Fu, M.D. Jennings, S.Y. Larin, T.M. Conte, "Software-only value speculation scheduling", *Technical Report*, Department of Electrical and Computer Engineering, North Carolina State University, 1998.
- 7) F.Gabbay, "Speculative execution based on value prediction", *Technical Report #1080*, Department of Electrical Engineering, Technion, 1996.
- 8) C-C.Lee, I-C.K.Chen, T.N.Mudge, "The bi-mode branch predictor", *30th International Symposium on Microarchitecture*, 1997.
- 9) M.H.Lipasti, C.B.Wilkerson, J.P.Shen, "Value locality and load value prediction", *International Conference on Architectural Support for Programming Languages and Operation Systems VII*, 1996.
- 10) E.Moranco, J.M.Llaberia, A.Olive, "Split last-address predictor", *International Conference on Parallel Architectures and Compilation Techniques*, 1998.
- 11) B. Rychlik, J.W. Faistl, B.P. Krug, A.Y. Kurland, J.J. Sung, M.N.Velev, J.P.Shen, "Efficient and accurate value prediction using dynamic classification", *Technical Report CMuART-98-01*, Department of Electrical Computer Engineering, Carnegie Mellon University, 1998.
- 12) Y.Sazeides, J.E.Smith, "Implementations of context based value predictors", *Technical Report TR-ECE-97-8*, Department of Electrical Computer Engineering, University of Wisconsin-Madison, 1997.
- 13) D.M.Tullsen, J.S.Seng, "Strageless value prediction using prior register values", *26th International Symposium on Computer Architecture*, 1999.
- 14) K.Wang, M.Franklin, "Highly accurate data value prediction using hybrid predictors", *30th International Symposium on Microarchitecture*, 1997.
- 15) Y.Zhang, J.Yang, R.Gupta, "Frequent value locality and value-centric data cache design", *International Conference on Architectural Support for Programming Languages and Operation Systems IX*, 2000.
- 16) 佐藤寿倫, 有田五次郎: データ幅を考慮したデータ値予測機構のハードウェア量削減, 並列処理シンポジウム JSPP2000, 2000.
- 17) 佐藤寿倫, 有田五次郎: タグビット幅を考慮したデータ値予測機構のハードウェア量削減, 信学技報 CPSY2000-3, 2000.