

木庭 優治[†], 中村 友洋, 吉瀬 謙二, 辻 秀典, 安島 雄一郎, 田中 英彦

{kiniwa, hiro, kis, hide, aji, tanaka}@mtl.t.u-tokyo.ac.jp

[†]東京大学工学部, 東京大学大学院 工学系研究科 *

1 はじめに

プロセッサの性能向上に伴って相対的に増大しているメモリアクセスのオーバーヘッドを低減する研究は盛んに行なわれており、データプリフェッチはその一つである。ハードウェアによるデータプリフェッチでは近い将来参照されるデータの動的な予測が非常に重要である。本稿では動的なロードアドレス予測に関して新たなアルゴリズムを提案し評価を行なう。

2 ロードアドレス予測

2.1 データプリフェッチ

データプリフェッチはプログラムの実行において必要となるデータをあらかじめ高速のメモリに移動させておくことによりメモリアクセスのレイテンシを削減しようとするものであり、大きくソフトウェアによるものとハードウェアによるものの2つに分けられる [1]。

このうちハードウェア・プリフェッチでは専用のハードウェアがロード命令の実行に先だてて参照アドレスを予測し実際のロード命令の実行前にデータが到着するようプリフェッチを行なう。従ってハードウェア・プリフェッチではロード命令の参照アドレスと実行タイミングの予測の正確さが重要となる。本稿ではそのうちのロードアドレスの予測に焦点を当てる。

2.2 既存のロードアドレス予測法

もっとも単純な予測法は OBL(One Block Look-ahead) と呼ばれる方法で、あるメモリブロックへの参照がキャッシュミスを起こした場合その次のブロックへの参照が近い将来おこると予測するものである。

ロードアドレス予測法でもっとも一般的な方法は各ロード命令の参照アドレスに対し一定の変位 (ストライド) を期待する方法である [1]。本稿ではこの方法をストライド法と呼ぶ。実際には各ロード命令の前の参照アドレス、ストライドなどを保持するテーブル (キャッ

シュとして実現される) を用意し、前回のアドレスにストライドの値を加えたものを予測アドレスとする。

その他の予測法としてはキャッシュミスを起こすアドレスの遷移がマルコフ過程になると仮定する Markov Predictor がある [3]。

3 新しい予測法の提案

ロード命令の参照アドレスの推移を調べると図 1 のようにストライドアクセスの間に一定の周期で異なるアドレス変位がおこるパターンが多く見られる。

これに着目し、ストライド法を拡張した複合ストライド法と呼ぶ予測法を提案する。通常のストライドアクセス時の変位を内側のストライド、周期的に起こる変位を外側のストライドと呼ぶことにす

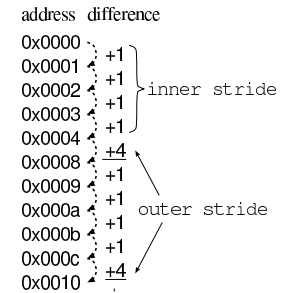


図 1: ロードアドレスのパターン

複合ストライド法ではストライド法と同様に各ロード命令の予測情報を保持するテーブル (図 2) をハードウェアキャッシュとして用意する。テーブルに保持する情報はロード命令の命令アドレス (tag)、前回の参照アドレス (addr)、内側のストライド (delta1)、外側のストライド (delta2)、外側のストライドが起こる間隔 (interval)、内側のストライドが何回行なわれたか (count)、delta1, delta2 の更新を決める状態ビット (state) である。予測の際はロード命令の命令アドレスをタグ情報としてこの表を引き、 $count = interval$ の時は予測アドレス = $addr + delta2$ 、 $count \neq interval$ の時は予測アドレス = $addr + delta1$ とする。

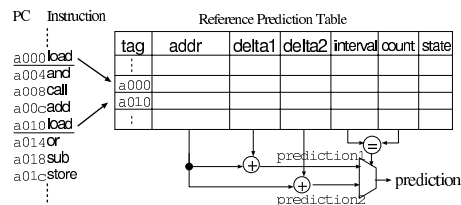


図 2: 予測テーブル

*"Prediction of Load Address using Complex-Stride Method"

Yuji Kiniwa, Tomohiro Nakamura, Kenji Kise, Hidenori Tsuji, Yuichiro Ajima, Hidehiko Tanaka
University of Tokyo, School of Engineering, Graduate School of Engineering,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

また、テーブルの更新のアルゴリズムは図3のようになる(実際のロードアドレスを Ar とする)。

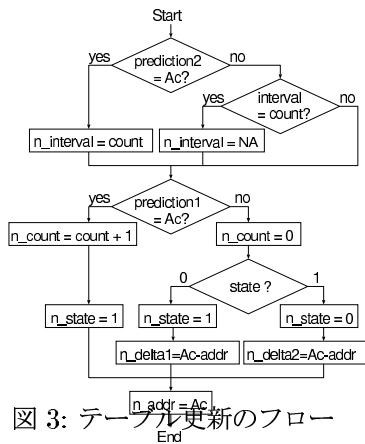


図 3: テーブル更新のフロー

4 評価

図4はテーブルのエントリ数を無限とした場合のストライド法と複合ストライド法の予測の成功率を示したものである。ただしストライド法は2回連続で間違えたときのみ delta の値を変更する [2] の方法を用いた場合である。複合ストライド法はストライド法に比べて

平均 4.52%、espresso では 10%以上の予測成功率の向上を得ている。

次にテーブルのハードウェア量について考察する。図5~7は複合ストライド予測が成功した時の delta1, delta2, interval の値の累積分布をとったグラフである。delta1, delta2 は 18 ビット程度、interval は 7 ビット程度でほぼ 100%となっている。delta1, delta2 20 ビット、interval, count : 8 ビット程度とすれば、複合ストライド法のテーブルのハード量は同じエントリ数のストライド法のものに比べ4~5割増となる。

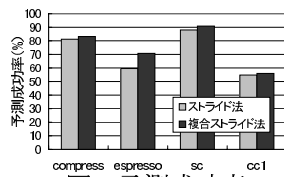


図 4: 予測成功率

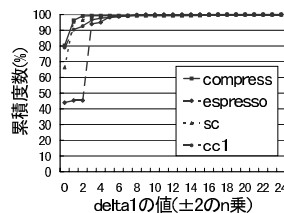


図 5: delta1 の値の分布

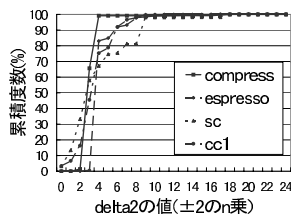


図 6: delta2 の値の分布

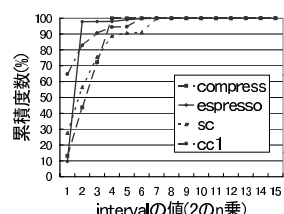


図 7: interval の値の分布

上の結果をもとに delta1, delta2 : 20 ビット、interval, count : 8 ビットとして、エントリの数を変化させてストライド法・複合ストライド法の予測成功率を調べた。図8はその結果を横軸ハード量、縦軸予測成功率で表したものである。

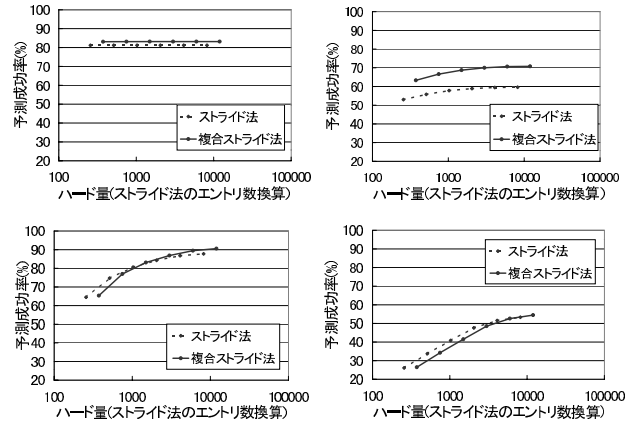


図 8: ハードウェア量と予測成功率

エントリ数が少ない場合、複合ストライド法を用いることによるハードウェア量の増加にみあう予測成功率の向上が見られず、ストライド法でエントリを増やした方が効率がよい場合がある。これはエントリ数がロード命令の数に比べて不足しているため、エントリ数を増やすことによる予測率向上の効果が大きいためである。エントリ数が十分ある場合は複合ストライド法の効果が大きくなる。

5 おわりに

複合ストライド法というアドレス予測法を提案しハードウェア量の評価などを行なった。その結果、予測テーブルのエントリがプログラム中のロード命令に対し十分ある場合にはストライド法に比べて高い予測性能を実現できることを確認した。

参考文献

- [1] Tien-Fu Chen, Jean-Loup Baer: "A Performance Study of Software and Hardware Data Prefetching Schemes", the 21st Annual International Symposium on Computer Architecture, pp.223-232, 1994
- [2] 勝野昭, 木村康則: "ロード・アドレス予測方法の検討", ARC117-9, 1996
- [3] D. Joseph, D. Grunwald: "Prefetching using Markov Predictors", the 24th International Symposium on Computer Architecture, pp.252-263, 1997