

投機的実行に関する最新技術動向

斎藤史子¹ 山名早人^{2,1}¹ 早稲田大学理工学部 ² アドバンスト並列化コンパイラ研究体

命令レベルの投機的実行について、制御依存を緩和する分岐予測とデータ依存を緩和するデータ予測に分類してまとめる。本稿では、1996-2001年のHPCA, 1996-2000年のISCA, MICRO, ASPLOSから分岐予測に関する論文36本、データ予測に関する論文27本を調査すると共に、他の論文誌から19本をとりあげまとめた。全体の傾向として、1998年までは、分岐予測とデータ予測に関する研究は分岐予測が全体の半分以上を占めていたが、1997年からデータ予測に関する研究も盛んになってきている。

The Latest Technical Trends in Speculative Execution

Fumiko Saito¹ Hayato Yamana^{2,1}¹ School of Science and Engineering, Waseda Univ. ² Advanced Parallelizing Compiler Project

Instruction level speculative execution schemes are classified into the branch prediction which alleviates control dependence, and the data prediction which alleviates data dependence. In this paper, we summarize 36 papers on the branch prediction and 27 papers on the data prediction in HPCA from 1996 to 2001, ISCA, MICRO, and ASPLOS from 1996 to 2000. As the general trends, until 1998, more than half of the researches on speculative execution are related to the branch prediction. However, since 1997, researches on data prediction have increased.

1 はじめに

投機的実行とは、実際に問題(分岐、データ)が解決するのに先駆けて、問題の結果を予測し、実行を滞りなく進行させることによりプログラムを高速に実行させる手法である。本稿では、分岐予測とデータ予測に着目してサーベイを行った。分岐予測は、制御依存を緩和する投機的実行であり、1972年頃から研究が行われている。また、分岐予測は、データ依存によってその性能が半分に制限される[39]ため、データ依存を緩和する投機的実行として1996年頃からデータ予測が提案されてきている。

以下、2は分岐予測について、3ではデータ予測についてまとめる。一般に分岐予測は制御情報に基づいて予測を行い、データ予測はデータ情報に基づいて予測を行う。4では、データ情報に基づいた分岐予測、制御情報に基づいたデータ予測をまとめる。5は、全体のまとめである。

2 分岐予測

分岐予測は、2分岐を予測する分岐方向予測と多分岐の分岐先も予測する分岐先アドレス予測に分類される。

2.1 分岐方向予測

現在、最も多く考案されている分岐方向予測の基礎となる動的な分岐予測器は、2レベル適応型分岐予測器(2-level adaptive branch predictor) [2]と呼ばれるものである。これは、2階層の表で構成され、1階層目の表(BHR, BHT:branch history register or table)¹には、分岐履歴²が格納され、2階層目の表(PHT:pattern history table)¹は、1階層目の表に格納された分岐履歴でインデックス付けされた飽和カウンタを複数持つ。この飽和カウンタに分岐の状態遷移を保持し、次の分岐方向を予測する。また、2レベル適応型予測器は、予測対象となる分岐以前に実行された有限個の分岐を参照する方法(global)と、分岐をグループ毎に参

照する方法(per-set)³と、分岐のアドレス毎に参照する方法(per-address)をそれぞれBHTとPHTに組み合わせで適用でき、9種類⁴に分類される[2]。特に、分岐履歴とPC(program counter)をXORしたものをPHTのインデックスとして用いた予測器をgshare [1]と呼び、多くの研究で用いられている。

(1) 2レベル適応型予測器の可能性

データ圧縮技法の適用 2レベル適応型分岐予測器は、データ圧縮で用いられる予測器(PPM: Prediction by Partial Matching) [83, 84]を単純化したものであると考え、データ圧縮技術を応用した分岐予測手法が提案されている[3, 12]。PPM⁵は、globalな履歴を持ち、長さの異なる履歴をエントリとする複数の予測器を階層的に組み合わせ、予測可能な最長の履歴を持つ予測器の予測結果を最終的な予測とする。PPMの予測ミス率は、PAgよりもわずかに低く、大容量のPHTで予測するよりも小容量のPHTを組み合わせの方が効率が良いことが分かる。また、データ圧縮技法の一種であるcontext tree weighting(CTW)⁶も条件分岐方向予測に適用でき[12]、PAgより予測ミス率がわずかに低い。

³グループは、分岐のオペコード、コンパイラによって割り当てられた分岐クラス、又は、分岐アドレスなどで決定される。

⁴GAg:global BHR+global PHT, GAS:global BHR+per-set PHT, GAp:global BHR+per-address PHT, PAg:per-address BHT+global PHT, PAs:per-address BHT+per-set PHT, PAp:per-address BHT+per-address PHT, SAg:per-set BHT+global PHT, SAs:per-set BHT+per-set PHT, SAp:per-set BHT+per-address PHT

⁵正確には、0-th order から m-th order の Markov predictor を階層的に組み合わせた予測器であり、予測可能な最も高い order の Markov predictor の予測結果を最終的な予測とする。j-th order Markov predictor は、最新 j 個の先行する bit 毎に次に続く bit の出現回数を数えておき、最も出現頻度が高いものを予測とする。

⁶2分木のノード毎に対応した global な履歴の出現確率で重み付けをすることで、次の分岐を予測

¹register は単一の履歴、table は複数の履歴を表す。

²分岐履歴は、一般には、分岐する場合(taken)には1、分岐しない場合(not-taken)には0である。

予測器の改良 globalな分岐履歴を採用した予測器では、PHTの競合が問題となる [7]。アドレスの一部のみを用いて PHT へのインデックスが行われるので、残りのアドレスをタグとして保持することによって、競合を除去する手法がある。しかし、頻繁に PHT のエントリの入替が行われ、却って予測正解率が低下する [20]。そこで、異なる偏向を持つ分岐による破壊的競合 (destructive interference) の影響を緩和するために、分岐の偏向を考慮して PHT の 2-bit 飽和カウンタの動作を見直した agree predictor [10] と複数の PHT を用いる手法 [8, 11, 9] が提案されている。

まず、agree predictor [10] について述べる。agree predictor の飽和カウンタは、分岐の偏向方向と分岐方向が一致した時はインクリメントし、不一致時はデクリメントする。これにより、分岐の偏向方向に一致する方向に分岐するか否かを予測する。agree predictor は、SPECint95 の gcc で gshare よりも最大 7% 高い予測正解率に達する。

複数の PHT を用いる手法には、PHT へのインデックス法を工夫した skewed predictor [9] と分岐方向に偏向した 2 つの PHT を選択器で選ぶ bi-mode predictor [8, 11] がある。skewed predictor は、skewed associate cache [86] のインデックス法にならって、BHR と複数の PHT 間をインデックス付け、PHT の予測結果の多数決をとることで最終的な予測を決定する。この手法は、gshare よりも小さいハードウェア量で同等の予測正解率に達する。また、bi-mode predictor は、gshare よりもわずかに予測正解率が改善するが、1 つの分岐が 2 つの PHT の両方にエントリを持つ点が冗長である。そのため、1 つの分岐がどちらかの PHT に対してしかエントリを持たない YAGS predictor (Yet Another Global Scheme) が提案された [11]。YAGS は、kernel trace を含んだ SPEC95 でも、以上に挙げたどの予測器よりも高い予測正解率に達する。さらに、PHT の選択法の改良も提案されているが、その効果は小さい [19, 21, 22]。

また、一定数の分岐毎に予測ミス数の変化に応じて 1 ずつ増減させることによって、最適な履歴長を動的に調整する手法として DHLF (dynamic history-length fitting) がある [13]。この手法は、最適な分岐履歴長で予測できるが、履歴長が変化すると PHT が初期化され、予測器の学習時間が長くなるという欠点を持つ。しかし、SPECint95 における gshare の予測正解率とほぼ同等の性能に達する。さらに、分岐方向の代わりに、global な分岐先アドレスを分岐履歴とし、分岐履歴長を静的に調整する手法がある [15]。この手法は、プロファイルによって、静的分岐⁸毎に分岐履歴長を決定する。SPECint95 の gcc において、この手法では、4KB の gshare よりも 4.5% 低い予測ミス率に達する。global な分岐履歴の代わりに、予測の対象とする分岐に至るまでのトレースを履歴として用いる試みもなされたが、その効果は少ない [18]。

他に、PHT のエントリに飽和カウンタの代わりにパーセプトロンを用いるという提案もある [17]。これは、4KB の容量では、gshare よりも 10.1% 予測正解率が優れている。

さらに、分岐方向の遷移確率 (branch transition rate) を予測の参考にすると、2 レベル適応型予測器で予測し難い分岐の約 9% が予測可能になる見込みがあることが示唆されている [14]。

confidence counter 分岐予測ミス時のペナルティを少なくするために、分岐の成功、失敗で遷移する confidence counter⁹ で分岐予測の信頼性を求め、信頼性が低い時は予測を行わずに両方のパスを同時実行するなど、ペナルティの少ない処理を行う手法が提案されている [6]。[6] では、動的分岐の 20% に影響を及ぼす一連の静的分岐に対する

⁷ 分岐予測において context switch を考慮する手法としては、一定数の命令間隔毎に履歴を一新するというモデルが主流であった [4] が、このモデルでは、実際に kernel trace を含んだ状況を正しく表現できない [5]。

⁸ ソースプログラム上の分岐

⁹ 主に飽和カウンタを使用

予測ミスが全体の予測ミスの約 63% を占めていることを示し、静的分岐毎に confidence を割り当てるよりも動的に confidence を割り当てることの重要性を示唆する。

分岐予測の遅延 分岐予測の遅延を考慮した実験もなされている [16]。[16] では、大きさの異なる 2 つの PHT を用意し、予測遅延により大容量の PHT に基づいた予測が利用できない時のみ、小容量の PHT に基づいた予測を採用する。この手法はハイブリッド予測器¹⁰と予測正解率においてはほぼ同等であるが、動作性能においては IPC が 1.2 程度で性能が良い。

(2) プロファイル

動的予測器では、予測器のハードウェア量が有限であるために、競合が起きる。そのため、プロファイルで予測できる分岐は静的に予測し、動的予測器で予測を行う分岐を減らすことによって、競合を削減する手法が提案されている [26]。また、per-address の履歴で問題となる BHT の競合を削減する [7] ために、プロファイルを用いて、互いに競合する分岐を working set と呼ばれる集合に集め、working set 内の分岐間の競合回数を求め、競合回数がある閾値以上の時は、同じ BHT にマッピングしない [24]。この手法では、動作性能を低下することなく、BHT のサイズを 60-80% まで減少できる。さらに、分岐が実行前に解決する場合には予測する必要がないことに着目して、プロファイルによって分岐方向を設定し、設定された値をそのまま予測に用いる手法が提案されている [23]。

他に、静的にプロファイルを用いる代わりに、動的プロファイルを利用する際のオーバーヘッドを防ぐために単純化したプロファイル¹¹をとる backward taken branch の予測手法 (NET¹² Hot Path Prediction) が提案された [25]。この手法は、静的なプロファイルを用いた時と匹敵した予測正解率に達する。

(3) multiple branch prediction

分岐毎に予測するのではなく、複数の分岐を block 単位で一度に予測する手法が提案されている [27, 28]。分岐方向を 16 命令からなる block 単位で予測する技法 [27] と tree 状に配置した動的予測器で分岐履歴に基づいて連続する基本ブロックを予測する手法 [28] が提案されている。[27] では、block の先頭命令の PC とその block 内に含まれる分岐の分岐結果を trace id として、block 毎に履歴をとり、次の block を予測する。block を単位とした分岐予測ミス率¹³は、16-bit の gshare に対して最大で 26% 低下する。

また、4 つの連続する基本ブロックを予測する手法として、Tree-based Multiple Branch Predictor (TMP) が提案されている [28]。[28] では、SPRCint95 において、32KB の TMP では、1 番目の block は 96%、2 番目の block は 93%、3 番目の block は 87%、4 番目の block は 82% の予測正解率である。

2.2 分岐先アドレス予測

分岐先アドレス予測は、分岐先のアドレス値を予測する手法であり、2 分岐と多分岐の両方を対象とする手法である。ハードウェア構成としては、分岐先アドレスバッファ (BTB: branch target buffer) と呼ばれる分岐先アドレスを PC でインデックス付けられたバッファに格納し、格納された分岐先アドレスに基づいて予測する構成と、分岐方向予測と同様に 2 階層の表を用いて予測する構成に分類される。

¹⁰ 2 つの予測器を組み合わせると分岐毎に予測正解率が高いと推定される予測を最終的な予測結果とする予測器

¹¹ path 毎にプロファイルをとる手法を動的に実装するとオーバーヘッドが大きいため、backward taken branch の戻り先についてプロファイルする手法に単純化する。

¹² Next Executing Tail

¹³ trace 内の分岐で予測ミスした分岐がある時、予測ミスとして計算

(1) 分岐先アドレスバッファ

分岐距離¹⁴のほとんどが 15bit 以下であることを利用し、分岐先アドレスを下位ビットと残りの上位ビットの 2 階層の表に分け、下位部分に格納できるアドレス bit よりも分岐先のアドレスが大きいときのみ、上位部分のアドレスが格納されている表を用いて、BTB の容量を削減する手法が提案されている [30, 31, 32]。この手法によって、分岐先アドレス部のハードウェア量を約 49%削減できる。

また、dynamic linked library (DLL) によって BTB miss が頻繁になるため、従来の BTB とは別に特別な BTB (DLL BTB) を用意することが提案されている [29]。DDL BTB は 1990 年に提案された Victim BTB [85] よりも最大で 14%性能が向上している。

(2) 2 階層予測器

分岐方向予測と同様に、2 階層の表を用いた予測器が提案されている [15, 33, 34, 36]。分岐先アドレス予測では、2 種類の分岐履歴情報 (pattern history¹⁵, path history¹⁶) があり、2 階層目の表を Target Cache と呼ぶ。また、分岐方向予測と同様に、分岐先アドレス予測でも、PPM の適用が考えられる [36]。BTB や間接分岐の分岐先アドレスのみでインデクス付けられた Target Cache [33] よりも、SPEC95 において PPM は平均予測ミス率が 9.47%で最も優れている。

また、Driesen らは、ハイブリッド予測器について検証した [34, 35]。2 階層の予測器のハイブリッド構成は単独の 2 階層の予測器に比べて、それほど性能が向上しない [34]。そのため、長い履歴の予測器が予測を失敗する時のみ、短い履歴の予測器をフィルターとして用いる cascaded predictor が提案された [35]。SPECint95 において、1K エントリの予測器に対して、BTB をフィルターとして用いた場合、全体の予測ミス率は 9.8%から 8.7%まで低下する。

さらに、分岐方向予測と同様に、global な分岐先アドレス履歴を持つ path-based branch predictor が間接分岐¹⁷に対して提案されている [15]。[15] では、プロファイルによって、静的分岐⁸毎に分岐履歴長を決定する。SPECint95 の gcc において、512B の予測器におけるこの手法の予測ミス率は、pattern based predictor [33] よりも 16.5%低い。

(3) two-block ahead branch predictor

現在の基本ブロックから次の基本ブロックに続く基本ブロックを予測する手法が提案されている [37]。[37] では、BTB のヒット率は従来のものと比べて、SPECint92 において、gshare と比べても 0.3%も予測正解率は低下せず、この手法は、投機的実行の開始タイミングを早めることができる点において大変興味深い。

(4) return address stack

投機実行する時に、return-address-stack の先頭のポインタやデータ値を保持し、分岐予測ミス時に復帰させるための回復機構を設けることで、return-address-stack の性能を改善する手法が提案されている [38]。SPECint95 では、この手法による予測正解率はほぼ 100%に近い

3 データ予測

データ依存関係は、出力依存関係、逆依存関係、フロー依存関係に分類できる。出力依存関係、逆依存関係はリネーミングで解決できる。残るフロー依存関係を解決するためにデータ予測が提案された。データ予測は、データ値を予測するデータ値予測、データアドレスを予測するデータア

ドレス予測、特にロード命令とストア命令の依存関係を予測するメモリ依存予測に分類される。

3.1 データ値予測

主なデータ値予測器は、命令毎に、最後に用いられた値を予測とする last-value prediction、値の差分を使って予測する stride value prediction、値の履歴によって予測を行う context based predictor に分類される [74]。主な context based predictor として fcm (Finite Context Method Predictors) がある。fcm は、先行する有限のデータ値パターンに続くデータ値毎の回数を数え、その中で最も出現回数が多いデータ値を予測とする。一般には、複数個の fcm を組み合わせ、一致するパターンの中で最長の値パターンに対応する fcm の予測結果を最終的な予測とする。

(1) レジスタ値の局所性

レジスタ値の局所性を利用した速度向上手法として、過去の実行と等しい入力があった場合には過去の実行をそのまま利用するレジスタ値再利用とデータ値を予測して投機的に処理を進めるデータ値予測が挙げられる。SPECint95 では、命令の 80-90%が冗長でそのうち 84-97%がレジスタ値の再利用が可能である [41]。このため、レジスタ値の再利用を活用して value buffer を不要にした register value prediction (RVP) が提案された [42]。この手法は、SPEC95 に対して平均 12%の性能向上を示す。

一方、データ値予測では、Lipasti らは、予測に信頼性があるレジスタ生成命令に対象を絞り、データ値の履歴の中に正しい予測結果となるものがあれば、必ず正しい予測を行う理想的な予測器を仮定した [40]。その結果、Power PC 620 において平均 5%の速度向上が見られる。

(2) データ値予測器の改良

新しいデータ値予測器としては、値自身の代わりに差分を履歴とする differential FCM (DFCM) が提案されている [43]。DFCM では、同じ差分を持つ stride pattern のエントリの冗長性が排除でき、従来の FCM よりも予測正解率がほぼ 33%優れている。

また、データ値の局所性やデータ幅を考慮することで、データ値予測器の容量を削減することも提案されている [44, 45]。まず、佐藤は SPECint95 では 50%のデータが 16 bit 以下であること [87] に着目し、データ幅が大きいデータ値予測器とデータ幅が小さいデータ値予測器を用意し、命令毎の演算結果のデータ幅の変動が小さいことから、最初の命令の演算結果が収まるように大小のデータ値予測器を選択する 2 モード値予測器を提案した [45]。その結果、わずかに予測正解率は低下したが、40%のハードウェア量が削減された。さらに、佐藤らは SPECint95 においてレジスタに書き込まれる値のうち平均で 15%が 0 であることに着目し、0 だけを予測するハードウェア規模が 4 倍の last-value に匹敵する性能の 0 値予測器を提案した [44]。

(3) ハイブリッド予測器

last-value は、stride value prediction において stride を 0 とした場合と同値であり、予測可能な対象に重なりがあるが、stride と context-based predictor は予測できる範囲が異なる。そのため、データ値予測で最もよく提案されるハイブリッド予測器は、stride と context-based predictor を用いたものである [46, 47, 48]。context based predictor として two-level predictor¹⁸を組み合わせたハイブリッド予測器は、SPECint92 においてレジスタ値を生成する命令のうちの約 50-82%の値を正確に予測でき、予測ミスは 5-18%である。

¹⁴ 分岐命令アドレスと分岐先アドレスの差分

¹⁵ global な条件分岐の分岐方向の履歴

¹⁶ 分岐の目的アドレスの履歴

¹⁷ 複数の分岐先をもつ分岐

¹⁸ [47] における two-level value predictor は、最近実行された命令で用いられた特有な 4 つの値を保持し、履歴をとる。2 階層目の表では、固有の値毎に出現回数を数え、その中で最も出現回数が多いものを予測とし、出現回数が閾値よりも小さい場合は予測は行わない。

(4) データ値予測ミスの回復

データ値予測は、予測ミスしたデータ値に依存する命令すべてをやり直さなければならず、予測ミス時のペナルティが大きい。そのため、予測ミス時のペナルティを小さくするための手法が提案されている [49, 50]。佐藤は、命令スケジューリングの自由度が低下するのを防ぐために、命令スケジューリングのための小さなスケジューリングウィンドウと命令発行のためのバッファを分割することを提案した [50]。この手法では、1 サイクル遅延の RUU(register update unit) と似たような動作性能に達することを示した。

また、Nakra らは、予測ミス時のペナルティを緩和するために、予測されたデータ値に依存した命令を処理するハードウェアを別に用意し、並列に実行する手法を提案した [49]。

(5) ロード値予測

ロード値を予測するために、Lipasti らは、ロード命令値の局所性に着目し、ロード命令を対象にして last-value を適用することを提案した [51]。この手法は、Power PC 620 において平均 3% の速度向上が見られる。

また、ロード命令とストア命令の関係に着目した手法がある。つまり、ロード命令は、ストア命令によってメモリに格納されたデータ値を使用する命令なので、互いに同じアドレスにアクセスするロード命令とストア命令を関連付ければ、予測できる可能性がある。佐藤は、同じアドレスにアクセスするロード命令とストア命令を関連付け、ストア命令によってメモリに格納された値を保持する機構を設けることにより、ロードされるデータ値を予測する機構を提案した [53, 54]。その結果、SPECint95 において、[50] の命令再発行機構を応用することによって、データ値の予測正解率の平均は 79.0% で、IPC は平均 3.4% 向上した。さらに、近年、ストア命令で格納されるデータ値にも、局所性があることが示された [52]。

(6) プロファイル

データ値予測は、予測ミスペナルティが大きいので、一般にすべての命令に適用されるわけではなく、予測可能と推察される命令を選んで予測が行われている。

Gabbay らは、データ値予測を適用する命令をプロファイルで選別した場合、プロファイルに用いる入力と実際の実行に用いる入力異なっても、似たような予測正解率であることを示した [55]。また、プログラムに予測器の正誤を確かめる分岐を挿入して、分岐予測器を利用することで、データ値予測の信頼性を測る法が提案されている [59]。

また、プロファイルで critical path 上にあると判断される命令のみデータ値予測を適用することで、総予測命令数を最大で半分削減し、予測命令を削減しない場合とほぼ同等の速度向上を得られた [58, 61]。さらに、critical path 上の命令を動的に予測する critical path prediction [60] が提案されている。

Fu らは、ハードウェアで実現されたデータ値予測器を利用して、ソフトウェアで静的にスケジューリングを行う value speculation scheduling (VSS) を提案した [56]。ロード命令を対象にして、予測器の大きさに制限が設けられていない場合、8CPU のマシン構成で SPECint95 の vortex に対して約 17% 速度向上を見せた。さらに、ソフトウェアで stride value predictor を実現した Software-Only Value Speculation Scheduling (SVSS) も提案されている [57] が、ソフトウェアで実現された予測器のオーバーヘッドが大きく、8% しか速度向上しない。

(7) 命令フェッチ幅の影響

命令フェッチ幅の与える影響についても研究されている [62, 63]。[62] では、同時に異なる命令から異なる bank に値予測が要求された場合、どちらの命令に対して予測するのかを選び、複数の同一命令から要求された場合には、

予測器に対応した処理¹⁹を行う。[63] では、命令フェッチステージでは予測を行わず、解読する時に予測を行う機構、decoupled value prediction を提案した。このことで不要な値予測が 20% 削減²⁰された。

3.2 データアドレス予測

データアドレス予測もデータ値予測と類似した last-value, stride, context-based predictor を用い、主にロード命令のアドレスを対象として予測が行われる。

(1) データアドレス予測の影響

データ依存を除去するために、ロード命令のアドレスを予測する投機的な load-speculation とデータ依存のある複数の命令をひとつにまとめる非投機的な data collapsing を比較する。data collapsing の対象となる命令は全体の 29-47% であり、stride value predictor で load-speculation を同時に適用しても性能向上の大部分が data collapsing に依存しているが、よりよい予測精度の load-speculation と共に適用することで data collapsing が効率的に利用できることも示されている [64]。

(2) データアドレス予測ミスの回復

データアドレス予測ミスをした場合でも、投機実行された命令とデータ依存にある命令だけを選択して再実行することでミスペナルティを軽減する機構が提案されている [50]。佐藤は [65] で、命令間のデータ依存関係を命令ウィンドウ中に保持し、予測ミスを犯したロード命令とデータ依存関係にある命令を命令ウィンドウ中で再発行している。その結果、命令レベル並列度は最大で 26.5% 向上した。

(3) データアドレス予測器の改良

データアドレス予測器でも、per-address な履歴に基づいたロード命令を対象とした予測手法が提案されている [66]。[66] では、過去の base address を履歴として、次の base address²¹ を予測する。また、confidence counter で予測実行の判断が行われ、分岐予測ミスによるペナルティを避けるため、データアドレス予測ミスをした分岐履歴パターンを保持し、その分岐履歴パターンと一致する時は予測しない。この予測器による平均速度向上は 10.2% である。

また、佐藤は、投機的に実行されたロード命令の結果をデータ依存の関係にある命令に伝播することによっても、動作速度が向上することを示す [67]。

3.3 データ値予測とデータアドレス予測の融合

佐藤は、ロード命令の読み出すデータを対象として、データ値予測とデータアドレス予測をハイブリッド構成した予測器を提案した [68]。この手法では、データ値予測が不可能な時に限り、データアドレス予測を適用する。その結果、予測可能な割合は平均 68.1% 改善し、動作性能は、平均 5.2% 向上した。

3.4 メモリ依存予測

Moshovos らは、データ依存に基づいたロード、ストア命令のスケジューリングを提案した [69]。[69] では、ロード命令とストア命令間の依存の有無を予測し、依存が存在すると予測された場合には、同期をとってロード、ストア命令を実行することを提案している。さらに、ロード命令の遅延は先行するストア命令に依存するため、一定ではない。そのため、ロード命令の遅延を動的に予測することによるロード命令の最適なスケジューリングの可能性が示唆されている [72]。

¹⁹last-value ではすべての要求に同じ値を返し、stride では命令の流れに応じて stride を加えた値を返す。

²⁰この時、基本とするアーキテクチャの速度向上は 18.2% である。

²¹offset は、LB(load buffer) に記録

また、ロード命令が依存しているストア命令の組を store set とした 2 階層の予測器が提案されている [71]。この手法と完全にメモリ依存を予測できる場合の動作性能の差はほとんどない。また、データ依存関係の有無の予測を利用して、メモリ内の操作を最適化する手法も提案されている [70]。

4 制御情報とデータ情報の融合

4.1 データ情報を用いた分岐予測

(1) 動的予測

分岐予測ミスのほぼ半分以上がその分岐のソースオペランドが予測可能な時に起きることが示され、分岐のソースオペランド値を予測することによって分岐予測正解率を高めうるということが示唆された [74]。しかし、データ値予測を直接用いても、それほど予測正解率の改善が期待できない [76, 77]。そのため、Heilらは、予測し難い分岐を分岐のソース・オペランド値で特徴づけることによって、その情報をハイブリッドな分岐方向予測器の選択器の代わりに用いる手法を提案した [75]。この手法によって、1-2%ではあるが、SPECint95 における平均予測正解率が単独の gshare, bi-mode よりも改善した。

さらに、複数の backward branch で終了判定が行われるループを対象に、繰返判定に影響を及ぼす命令を data collapsing して、繰返の判定条件を stride value predictor を用いて k 回スキップすることで繰返判定を早く解決しようという手法が提案されている [73]。この手法は、SPECint95 において、平均予測ミス遅延を 60%まで減少する。

(2) 静的予測

一定サイクル前の命令が用いるレジスタのデータ値をインデクスとする配列を用意し、データ値のプロファイルによって予測する compiler synthesized dynamic branch prediction が提案されている [78]。[78] では、分岐時は分岐以前のレジスタのデータに対応する配列の要素をインクリメントし、非分岐時はデクリメントする。次に、配列の要素が正数になる場合と負数になる場合をうまく分割するように閾値を設定する。予測は、命令実行時のデータと閾値を比較することで行われる。この手法は、SPEC92 において、2 レベル適応型予測器と同等の予測正解率に達する。

4.2 制御情報を用いたデータ予測

制御情報を用いてデータ値を予測しようという試みもなされている [79, 80, 81, 82]。

Nakra らは、最近実行された global な分岐履歴と PC で VPT にインデクス付けする予測器を提案した [80]。この手法を従来のデータ値予測器に適用すると、7-9%予測正解率が改善する。

また、予測する命令までの制御流の変化に対しても考慮するように stride value predictor を拡張した 2 レベル・ストライド値予測機構が提案されている [81, 82]。この 2 レベル・ストライド値予測機構は、値が一定間隔で変化している時と値が初期化されるときで異なった制御の流れをとると推察し、制御流によって予測器の動作を変える。この手法によって、stride value predictor に比較しヒット率²²は、4.8%改善し、予測ミス率は 0.6%低下する。他に、trace 毎に予測の対象となるレジスタ値の差分を持つ increment predictor が提案されている [79]。

5 おわりに

本稿では、命令レベルの投機的敵実行を分岐予測とデータ予測に分類して、1996-2001 年の HPCA, 1996-2000 年の ISCA, MICRO, ASPLOS を対象にサーベイを行った。分岐予測、データ予測に関する論文は、1998 年に 17 本も発表され、最も盛んに研究されたが、それ以降は本数は減ったもののほぼコンスタントに研究されている。

²² 値予測器が正しく値を予測した回数を実行命令数で割った値

参考文献

- [1] S.McFarling: "Combining branch predictors", Technical Report TN-36, Digital Western Research Laboratory, June 1993
- [2] T.-Y.Yeh, Y.N.Patt: "A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History", 20th ISCA, pp.257-266, 1993
- [3] I.K.Chen, J.T.Coffey, and T.N.Mudge: "Analysis of Branch Prediction via Data Compression", 7th ASPLOS, pp.128-137, 1996
- [4] M.Evers, P.-Y.Chang, and Y.N.Patt: "Using Hybrid Branch Predictors to Improve Branch Prediction Accuracy in the Presence of Context Switches", 23rd ISCA, pp.3-11, 1996
- [5] N.Gloy, C.Young, et al.: "An Analysis of Dynamic Branch Prediction Schemes on System Workloads", 23rd ISCA, pp.12-21, 1996
- [6] E.Jacobsen, E.Rotenberg, and J.E.Smith: "Assigning Confidence to Conditional Branch Predictions", MICRO-29, pp.142-152, 1996
- [7] S.Sechrest, C.-C.Lee, and T.Mudge: "Correlation and Aliasing in Dynamic Branch Predictors", 23rd ISCA, pp.22-32, 1996
- [8] C.-C.Lee, I.K.Chen, and T.N.Mudge: "The Bi-Mode Branch Predictor", MICRO-30, pp.4-13, 1997
- [9] P.Michaud, A.Seznez, and R.Uhlig: "Trading Conflict and Capacity Aliasing in Conditional Branch Predictor", 24th ISCA, pp.292-303, 1997
- [10] E.Sprangle, R.S.Chappell, et al.: "The Agree Predictor: A Mechanism for Reducing Negative Branch History Interference", 24th ISCA, pp.284-291, 1997
- [11] A.N.Eden, T.Mudge: "The YAGS Branch Prediction Scheme", MICRO-31, pp.69-77, 1998
- [12] E.Federovsky, M.Feder, and S.Weiss: "Branch Prediction Based on Universal Data Compression Algorithms", 25th ISCA, pp.62-72, 1998
- [13] T.Juan, S.Sanjeevan, and J.J.Navarro: "Dynamic History-Length Fitting: A third level of adaptivity for branch prediction", 25th ISCA, pp.155-166, 1998
- [14] M.Haug, P.Salle, and M.Farrens: "Branch Transition Rate: A New Metric for Improved Branch Classification Analysis", 6th HPCA, pp.241-250, 1998
- [15] J.Stark, M.Evers, and Y.N.Patt: "Variable Length Path Branch Prediction", 8th ASPLOS, pp.170-179, 1998
- [16] D.A.Jimenez, S.W.Keckler, and C.Lin: "The Impact of Delay on the Design of Branch Predictors", MICRO-33, pp.67-76, 2000
- [17] D.A.Jimenez, C.Lin: "Dynamic Branch Prediction with Perceptrons", 7th HPCA, pp.197-206, 2001
- [18] 児島, 弘中: "複数分岐での投機的実行の有効性", ARC-123-10, pp.55-60 (1997.3)
- [19] 野口, 森, et al.: "分離型パターン履歴表による分岐予測機構の競合耐性の改善", JSPP'98, pp.7-14 (1998.6)
- [20] 佐藤: "2 レベル適応型分岐予測機構のパターン履歴表におけるタグの影響", 信学論 D-1, J81-D-1.6, pp.728-737 (1998.6)
- [21] 野口, 森, et al.: "分岐方向の偏りを利用し破壊的競合を低減する分岐予測方式", 情処論, pp.2119-2131 (1999.5)
- [22] 布施, 安藤, 島田: "sgshare 分岐予測機構における選択機構が予測性能に与える影響の評価", ARC-141-10, pp.53-64 (2001.1)
- [23] D.I.August, D.A.Connor, et al.: "Architectural Support for Compiler-Synthesized Dynamic Branch Prediction Strategies: Rationale and Initial Results", 3rd HPCA, pp.84-93, 1997
- [24] S.P.Kim, G.S.Tyson: "Analyzing the Working Set Characteristics of Branch Execution", MICRO-31, pp.49-58, 1998
- [25] E.Duesterwald, V.Bala: "Software Profiling for Hot Path Prediction: Less is More", 9th ASPLOS, pp.202-211, 2000
- [26] H.Patil, J.Emmer: "Combining Static and Dynamic Branch Prediction to Reduce Destructive Aliasing", 6th HPCA, pp.251-262, 2000
- [27] Q.Jacobson, E.Rotenberg, and J.E.Smith: "Path-Based Next Trace Prediction", MICRO-30, pp.14-23, 1997

- [28] R.Rakvic, B.Black, and J.P.Shen: "Completion Time Multiple Branch Prediction for Enhanced Trace Cache Performance", 27th ISCA, pp.47-58, 2000
- [29] S.Vlaovic, E.S.Davidson, and G.S.Tyson: "Improving BTB Performance in the Presence of DLLs", MICRO-33, pp.77-86, 2000
- [30] 小林, 山田, et al.: "2 レベル表方式による分岐先バッファ", 情処論, 41.5, pp.1351-1359 (2000.5)
- [31] 山田, 小林, et al.: "分岐先アドレスの性質を利用した 2 レベル表による分岐先バッファの容量削減", ARC-131-9, pp.59-64 (1998.9)
- [32] 山田, 小林, et al.: "2 レベル表構成の導入による分岐先バッファの容量削減", JSPP'99, pp.103-110 (1999.3)
- [33] P.-Y.Chang, E.Hao, and Y.N.Patt: "Target Prediction for Indirect Jumps", 24th ISCA, pp.274-283, 1997
- [34] K.Driesen, U.Holzle, "Accurate Indirect Branch Prediction", 25th ISCA, pp.167-178, 1998
- [35] K.Driesen, U.Holzle: "The Cascaded Predictor: Economical and Adaptive Branch Target Prediction", MICRO-31, pp.249-258, 1998
- [36] J.Kalamatianos, D.R.Kaeli: "Predicting Indirect Branch via Data Compression", MICRO-31, pp.272-281, 1998
- [37] A.Seznec, S.Jourdan, et al.: "Multiple-Block Ahead Branch Predictors", 7th ASPLOS, pp.116-127, 1996
- [38] K.Skadron, P.S.Ahuja, et al.: "Improving Prediction for Procedure Returns with Return-Address-Stack Repair Mechanisms", MICRO-31, pp.259-271, 1998
- [39] E.Rotenberg, Q.Jacpbson, and J.E.Smith: "A Study of Control Independence in Superscalar Processors", 5th HPCA, pp.115-124, 1999
- [40] M.H.Lipasti, J.P.Shen: "Exceeding the Dataflow Limit via Value Prediction", MICRO-29, pp.226-237, 1996
- [41] A.Sodani, G.S.Sohi: "Understanding the Differences Between Value Prediction and Instruction Reuse", MICRO-31, pp.205-215, 1998
- [42] D.M.Tullsen, J.S.Seng: "Storageless Value Prediction Using Prior Register Values", 26th ISCA, pp.270-279, 1999
- [43] B.Goeman, H.Vandierendock, and K.D.Bosschere: "Differential FCM: Increasing Value Prediction Accuracy by Improving Table Usage Efficiency", 7th HPCA, pp.207-216, 2001
- [44] 佐藤, 有田: "複雑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", SLDM-98-4, pp.17-22 (2000.5)
- [45] 佐藤, 有田: "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", JSPP'00, pp.245-252 (2000.5)
- [46] Y.Sazeides, J.E.Smith: "The Predictability of Data Values", MICRO-30, pp.248-258, 1997
- [47] K.Wang, M.Franklin: "Highly Accurate Data Value Prediction using Hybrid Predictors", MICRO-30, pp.281-290, 1997
- [48] Y.Sazeides, J.E.Smith: "Limits of Data Value Predictability", International Journal of Parallel Programming, Vol. 27, No.4, pp.229-256, 1999
- [49] T.Nakra, R.Gupta, and M.L.Soffa: "Value Prediction in VLIW Machines", 26th ISCA, pp.258-269, 1999
- [50] 佐藤: "データ投機実行のための命令再発行機構と命令スケジューリング機構の分割", ARC-133-4, pp.25-30 (1999.5)
- [51] M.H.Lipasti, C.B.Wilkerson, and J.P.Shen: "Value Locality and Load Value Prediction", 7th ASPLOS, pp.138-147, 1996
- [52] K.M.Lepak, M.H.Lipasti: "On the Value Locality of Store Instructions", 27th ISCA, pp.182-191, 2000
- [53] 佐藤: "アドレス名前替えによるロード命令の投機的実行", JSPP'98, pp.15-22 (1998.6)
- [54] 佐藤: "2 ホップアドレス名前替えを用いたロード命令の投機的実行", 情処論, pp.2109-2118 (1999.5)
- [55] F.Gabbay, A.Mendelson: "Can Program Profiling Support Value Prediction?", MICRO-30, pp.270-280, 1997
- [56] C.-Y.Fu, M.D.Jennings, et al.: "Value Speculation Scheduling for High Performance Processors", 8th ASPLOS, pp.262-271, 1998
- [57] C.-Y.Fu, M.D.Jennings, S.Y.Larin, T.M.Conte: "Software-Only Value Speculation Scheduling", Technical Report, 1998
- [58] B.Calder, G.Reinman, and D.M.Tullsen: "Selective Value Prediction", 26th ISCA, pp.64-74, 1999
- [59] E.Larson, T.Austin: "Compiler Controlled Value Prediction using Branch Predictor Based Confidence", MICRO-33, pp.327-336, 2000
- [60] E.Tune, D.Liang, et al.: "Dynamic Prediction of Critical Path Prediction", 7th HPCA, pp.185-196, 2001
- [61] 飯塚, 小沢, et al.: "プロファイルを用いた値予測命令削減手法", HPC-82-22, pp.125-136 (2000.8)
- [62] F.Gabbay, A.Mendelson: "The Effect of Instruction Fetch Bandwidth on Value Prediction", 25th ISCA, pp.272-281, 1998
- [63] S.-J.Lee, Y.Wang, and P.-C.Yew: "Decoupled Value Prediction on Trace Processors", 6th HPCA, pp.231-240, 2000
- [64] Y.Sazeides, S.Vassiliadis, and J.E.Smith: "The Performance Potential of Data Dependence Speculation & Collapsing", MICRO-29, pp.238-247, 1996
- [65] 佐藤: "命令再発行機構によるデータアドレス予測に基づく投機実行の効果改善", 情処論, 40.5, pp.2093-2108 (1999.5)
- [66] M.Bekerman, S.Jourdan, and G.Kishenboim: "Correlated Load-Address Predictors", 26th ISCA, pp.54-63, 1999
- [67] 佐藤: "ロード・ストアアドレスの早期生成によるデータ依存解消手法", ARC-125-1, pp.1-6 (1997.8)
- [68] 佐藤: "データ値予測とアドレス予測を組み合わせたデータ投機実行", JSPP'99, pp.111-118 (1999.6)
- [69] A.Moshovos, S.E.Breach, et al.: "Dynamic Speculation and Synchronization of Data Dependences", 27th ISCA, pp.181-193, 1997
- [70] A.Moshovos, G.S.Sohi: "Streaming Inter-operation Memory Communication via Data Dependence Prediction", MICRO-30, pp.235-245, 1997
- [71] G.Z.Chrysos, J.S.Emery: "Memory Dependence Prediction using Store Sets", 25th ISCA, pp.142-153, 1998
- [72] A.Yoaz, M.Erez, et al.: "Speculation Techniques for Improving Load Related Instruction Scheduling", 26th ISCA, pp.42-53, 1999
- [73] A.Farcy, O.Temam, et al.: "Dataflow Analysis of Branch Mispredictions and Its Application to Early Resolution of Branch Outcomes", MICRO-31, pp.59-68, 1998
- [74] Y.Sazeides, J.E.Smith: "Modeling Program Predictability", 25th ISCA, pp.73-84, 1998
- [75] T.H.Heil, Z.Smith, and J.E.Smith: "Improving Branch Predictors by Correlating on Data Values", MICRO-32, pp.28-37, 1999
- [76] 戸田, 布施, et al.: "値予測を用いた分岐予測", JSPP'00, pp.237-244 (2000.5)
- [77] 中村, 片山, et al.: "値予測を用いた分岐予測機構の計算機性能に与える影響", ARC-141-11, pp.59-64 (2001.1)
- [78] S.Mahlke, B.Natarajan: "Compiler Synthesized Dynamic Branch Prediction", MICRO-29, pp.153-164, 1996
- [79] P.Marcuello, J.Tubella, and A.Gonzalez: "Value Prediction for Speculative Multithreaded Architecture", MICRO-32, pp.230-236, 1998
- [80] T.Nakra, R.Gupta, and M.L.Soffa: "Global Context-Based Value Prediction", 5th HPCA, pp.4-12, 1999
- [81] 吉瀬, 坂井, 田中: "マルチレベル・ストライド値予測機構による命令レベル並列性の向上", JSPP'98, pp.119-126 (1998.6)
- [82] 吉瀬, 坂井, 田中: "2 レベル・ストライド値予測機構の可能性検討", 情処論, 41.5, pp.1340-1350 (2000.5)
- [83] J.G.Cleary, I.H.Witten: "Data compression using adaptive coding and partial string matching", IEEE Trans. on Comp., 32.4, pp.396-402 (1984.4)
- [84] A.Moffat: "Implementing the PPM data compression scheme", IEEE Trans. on Com., 38.11, pp.1917-1921 (1990.11)
- [85] N.Jouppi: "Improving Direct-Mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers", 17th ISCA, pp.364-373, 1990
- [86] A.Seznec: "A case for two-way skewed-associative caches", 20th ISCA, pp.169-178, 1993
- [87] D.Brooks, M.Martonosi: "Dynamically exploiting narrow width operands to improve processor power and performance", 5th HPCA, pp.13-22, 1999