様式 C-19

科学研究費補助金研究成果報告書

平成　22 年　5 月　1 日現在

研究種目：　基盤研究(C)
研究期間：　2007　～　2009
課題番号：　19500041
研究課題名（和文）　高性能コンピュータの方式に関する研究
研究課題名（英文）　Study on architectures of high-performance computers

研究代表者
安藤　秀樹（Hideki Ando）
名古屋大学・大学院工学研究科・教授
研究者番号：40293667

研究成果の概要（和文）：本研究では、低速な主記憶による性能劣化を抑えるデータのプリフェッチ方式を考案した。プリフェッチとは、プロセッサが必要とする以前に、主記憶からデータを事前にキャッシュに取り込んでおくものである。プリフェッチは従来より研究されてきたが、配列アクセスのような予測可能な規則的パターンにしか対応できなかった。これに対して本研究では、従来不可能とされてきた不規則なパターンを含む種々のパターンに対応できる汎用的なプリフェッチ方式を考案した。

研究成果の概要（英文）：This study proposed a data prefetch scheme that suppresses performance degradation due to slow main memory. Here, prefetching moves data from main memory to a cache before a processor requires them. Although many studies have been carried out, they are successful only for regular patterns like array accesses, which are highly predictable. On the other hand, this study proposed a general prefetch scheme, which allows to successfully handle various data access patters including irregular patterns.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2007 年度	900,000	270,000	1,170,000
2008 年度	1,000,000	300,000	1,300,000
2009 年度	1,100,000	330,000	1,430,000
年度			
年度			
総　計	3,000,000	900,000	3,900,000

研究分野：総合領域
科研費の分科・細目：情報学　・　計算機システム・ネットワーク
キーワード：データ・プリフェッチ

１．研究開始当初の背景

プロセッサのクロック速度は、年率約 60%という非常に高い速度で向上しているが、一方で、主記憶をなす DRAM のアクセス速度は年率 5%程度でしか向上しない。この結果、現在では、プロセッサと主記憶の間には、100 倍以上もの速度差がある。この傾向は今後も続き、速度差は指数的に（年率 50%）増大していく。このギャップを埋めるため、主記憶とプロセッサの間に数階層のキャッシュを配置することが一般的である。キャッシュと

は、最近アクセスしたデータを保存しておく、主記憶よりは小容量であるが高速なメモリである。最近アクセスされたデータほど再びアクセスされるという傾向を利用している。アクセスするデータがキャッシュに見つかれば（ヒットという）、短い時間内にデータを供給できる。しかし見つからない場合（ミスという）、主記憶にまでアクセスしなくてはならず、性能は大きく低下する。前述したように、現在ではプロセッサと DRAM の速度ギャップが非常に大きいので、わずかなミスでも性能低下は著しく生じる。ミスを極力さけるため、現在のコンピュータではキャッシュの容量を年々大きくしており、現在ではプロセッサ・チップの大半を占めるに至っている。これは高コスト化の最大の要因となっている。それにも関わらず、キャッシュ・ミスにより実際の性能はピーク性能の数分の 1 しかない応用も数多く存在する。コスト制約が厳しい電子機器では、キャッシュは小容量にせざるをえなく、ミスは頻繁に起こるので、プロセッサ／DRAM 間速度ギャップはより深刻な問題である。さらに、ソフトウェアは年々複雑化しており、それに応じて扱うデータ量も増加している。また、メディア処理など大量のデータを扱う応用が増加している。これによってキャッシュの容量不足がより深刻に起こり、ミスの確率が上昇し、本問題をより悪化させている。

２．研究の目的

本研究では、低速な主記憶による性能劣化を最小限に抑えるデータのプリフェッチ方式を検討する。プリフェッチとは、プロセッサが必要とする以前に、主記憶からデータを事前にキャッシュに取り込んでおくものである。後述するように、プリフェッチは従来より研究されてきたが、現在のプロセッサが搭載しているプリフェッチ機構は、配列アクセスのような予測可能な規則的パターンにしか対応できない。このため効果は限定的である。これに対して本研究では、従来不可能とされてきた不規則なパターンを含む種々のパターンに対応できる汎用的なプリフェッチ手法を研究する。これにより、コンピュータをはじめプロセッサ搭載機器におけるあらゆる応用において、プロセッサのクロック速度向上に応じた高い性能を得ることが可能となる。また、キャッシュが小容量ですむので、コスト性能比の良いコンピュータや電子機器が実現できる。

３．研究の方法

本研究の目的は、不規則パターンにも対応する汎用な方式を見いだすことにあるが、不規則パターンは予測できないので、従来とは全く別のアプローチが必要である。本研究では、次のようなアプローチでプリフェッチを行う。すなわち、プロセッサが実際に計算を行う命令流に先んじる命令流を生成する。これを仮に、すなわち、マシン状態を変更することなく実行し、メモリ・アクセスを事前に行いプリフェッチを実現する。このアプローチは従来のように予測に依存することなく、実際の実行に則したものであるから、不規則パターンに対応できる。

４．研究成果

(1) 基本方式の提案

実際に計算を行う命令流に先んじる実行を先行実行と呼ぶ。本研究では、2 段階物理レジスタ開放方式（TSD; two-step physical register deallocation）と呼ぶ先行実行方式を提案した。次のような仕組みのものである。

まず最初に、リネーム・ステージで命令のデスティネーション・レジスタに現在割り当てられている物理レジスタを仮に開放する。従来と異なりこの時点で物理レジスタが開放されるので、リネーム時の物理レジスタ不足によるパイプライン・ストールは完全に除去される。リネーム後、命令は発行キューに挿入される。割り当てられた物理レジスタが真に開放され書き込み可能となると、それを命令に通知する仕組みを発行キューに用意する。そのような通知が行われるまで命令の実際の実行は行えない。しかし、その待ち合わせ時間の間に、ソース・オペランドが利用可能になったら、マシン状態を更新することなく先行的に実行する。この実行は次の依存命令を実行可能とし、実行は連鎖的に生じる。この先行実行において、ロードが実行されキャッシュ・ミスが生じると、メモリからのデータがフェッチされ、これは実効的にプリフェッチとなる。つまり、有限の物理レジスタの下での実際の実行のスループットと、無限の物理レジスタの下での実行のスループットの違いを利用して先行実行を実現し、それによりプリフェッチを実現するものである。

図 1 に性能評価結果を示す。同図は、各ベンチマーク（横軸）における IPC (instructions per cycle)を示している。IPC とは 1 サイクルに実行された平均命令数であり、性能を表す指標である。各ベンチマークにつき 4 つの棒グラフがある。左から順に以下の 4 つのモデルの IPC である。

・プリフェッチも先行実行も行わない基準となる通常のプロセッサ

・基準プロセッサに、規則的なストライド・メモリ・アクセスに対応するストライド・プ

リフェッチャを導入したもの
・TSD による先行実行を行うもの
・ストライド・プリフェッチャと TSD を組み合わせたもの
　図からわかるように、TSD は平均で基準プロセッサに対し 26%の性能向上を達成している。ストライド・プリフェッチャを備えたプロセッサに比べると性能は落ちるが、それを TSD に組みこめば、18%高い性能を達成することができる。以上より、TSD は不規則なメモリ・アクセス・パターンにも対応し、高い性能を達成できることが確認できた。

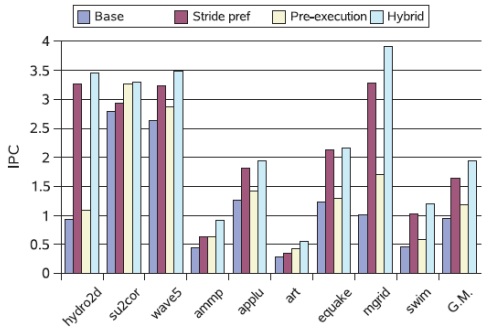

図 1：性能

(2) 電力効率の向上

前述したように、TSD は大きな性能向上を達成することができるが、電力効率が悪いという問題があった。具体的には、TSD はできるだけ多くの命令を先行実行するように働くが、先行実行の中には性能向上に寄与しないものも多くあった。性能向上に寄与しない先行実行は電力を消費するのみであり、できるだけ削減することが望まれる。そこで、本研究では、次のような方式を提案した。
　まず、頻繁に L2 キャッシュ・ミスを生じるロードを見つける。このようなロードを delinquent load（DL）と呼ぶ。過去の研究から、少数のロードが L2 ミスのほとんどを占めていることが知られている。DL は履歴ベースで見つける。DL が見つかったら、DL に先行する命令をバッファに格納し、データフロー解析を行う。これにより DL に直接あるいは間接的に依存する命令を特定する。これらの命令を先行実行の候補とする。
　次に、候補となった命令列を、性能への寄与を下げずにできるだけ短くする操作を行う。具体的には、命令列の先頭の少数の命令を先行実行しないようにし、性能への寄与を測定する。その結果、寄与がほとんどないことがわかれば、これらの命令は先行実行の必要がないと判断する。これを繰り返すことにより、先行実行命令列長を最短にする。
　図 2 に全動的命令に対し先行実行された動的命令の割合（先行実行率）を示す。各ベンチマーク・プログラムについて 2 本の棒グラフがある。左は、電力効率改善を行わない従来の TSD の場合であり、右は本提案方式を導入した TSD の場合である。グラフからわかるように、先行実行される命令は大幅に減少し、平均で 76%減少させることができた。このときの性能低下率はわずか 2%である。この結果、実行コアの消費エネルギーを 7%減少させることができることを確認した。

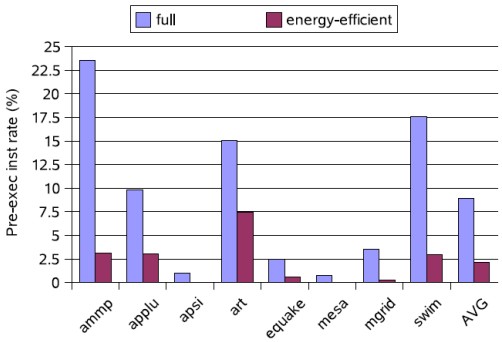

図 2：先行実行率

(3) レジスタ・ファイル縮小

TSD による性能向上をさらに大きなものにするには、より多くの命令を先行実行し、しかも実際の実行より大きく先行する必要がある。解析の結果、これを妨げている要因として、2 つのことがわかった。1 つは、TSD における先行実行命令間のデータ受け渡しにレジスタが存在せずバイパス論理で受け渡しているが、タイミング上これに失敗する点である。もう 1 つは、連続して L2 キャッシュ・ミスを生じる場合、後のミスほどプリフェッチによるペナルティ隠蔽効果が低い点である。これらはいずれもデータ依存が問題である。そこで、解決のために値予測を導入することを提案した。
　図 3 はその効果の測定結果である。縦軸は、物理レジスタ数を変化させたときの性能（IPC）を示している。折れ線グラフが 3 つある。最も下の線は、基準プロセッサの場合であり、中ほどの線は、従来の TSD の場合であり、上の線は、値予測を導入した場合の TSD の場合である。同図からわかるように、値予測を導入した TSD は、従来の TSD と比較して、より少ない物理レジスタ数で同一の性能を達成している。たとえば、物理レジスタが 96 個の時を基準とすると、同一の性能を従来の TSD では 64 個で達成し、値予測を導入した TSD ではさらに少なく 48 個で達成することができた。すなわち、値予測を導入することにより、従来の TSD の場合と比較して、レジスタ・ファイル・サイズを 25%縮小できることを確認した。

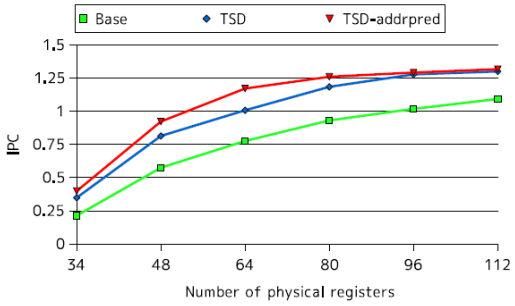

図 3：レジスタ数に対する性能

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 12 件）

1. K. Hyodo, K. Iwamoto, and H. Ando, ''Energy-Efficient Pre-Execution Techniques in Two-Step Physical Register Deallocation,'' IEICE Transactions on Information and Systems, Vol.E92-D, No.11, pp.2186-2195, November 2009.査読有

2. A. Yamamoto, Y. Tanaka, H. Ando, and T. Shimada, ''Two-Step Physical Register Deallocation for Data Prefetching and Address Pre-Calculation,'' IPSJ Transactions on Advanced Computing Systems, Vol. 1, No. 2, pp.34-46, August 2008.査読有

〔学会発表〕（計 12 件）

1. 甲良祐也、安藤秀樹、''命令発行キューの遅延時間評価,'' 2010 年先進的計算基盤システムシンポジウム SACSIS 2010, 2010 年 5 月. (採録決定、2010/5/27, 奈良市)

2. 岩原佑磨、安藤秀樹、''リオーダ・バッファのハードウェア量と消費エネルギーの削減,'' 2010 年先進的計算基盤システムシンポジウム SACSIS 2010, 2010 年 5 月. (採録決定、2010/5/27, 奈良市)

3. Y. Tanaka and H. Ando, ''Reducing Register File Size through Instruction Pre-Execution Enhanced by Value Prediction,'' In Proceedings of the 27th IEEE International Conference on Computer Design, pp.238-245, October 2009. (2009/10/04, Olympic Valley, CA, USA)

4. 加藤伸幸、安藤秀樹、''命令発行キューの深いパイプライン化のための投機発行,'' 2009 年先進的計算基盤システムシンポジウム SACSIS 2009, pp.319-326, 2009 年 5 月. (2009/5/29, 広島市)

5. 田中雄介、安藤秀樹、''値予測を用いた命令先行実行によるレジスタ・ファイルの縮小,'' 2009 年先進的計算基盤システムシンポジウム SACSIS 2009, pp.335-343, 2009 年 5 月. (2009/5/29, 広島市)

6. 岩本健吾, 安藤秀樹, ''物理レジスタ 2 段階解放方式の低消費電力化手法の評価,'' 情報処理学会研究報告, 2009-ARC-183, 2009 年 4 月.(2009/4/24, 沖縄市)

7. 田中雄介, 安藤秀樹, ''値予測を用いた物理レジスタ 2 段階開放による命令先行実行方式の性能向上,'' 情報処理学会研究報告, 2008-ARC-180, pp.3-8, 2008 年 10 月. (2008/10/20, 二日市市)

8. 加藤伸幸, 安藤秀樹, ''命令発行キューの深いパイプライン化,'' 情報処理学会研究報告, 2008-ARC-179, pp. 37-42, 2008 年 8 月. (2008/8/6, 佐賀市)

9. 兵藤一永、安藤秀樹、''選択的先行実行による物理レジスタ 2 段階解放方式の低消費電力化,'' 2008 年先進的計算基盤システムシンポジウム SACSIS 2008, pp.237-244, 2008 年 6 月. (2008/6/11, 旭川市)

10. A. Yamamoto, Y. Tanaka, H. Ando, and T. Shimada, ''Data Prefetching and Address Pre-Calculation through Instruction Pre-Execution with Two-Step Physical Register Deallocation,'' In Proceedings of the Eighth Workshop on Memory Performance: Dealing with Applications, Systems and Architectures, pp.41-48, September 2007. (2007/9/14, Brasov, Romania)

11. 兵藤一永、安藤秀樹、''物理レジスタ 2 段階解放による命令先行実行方式の低消費電力化,'' 情報処理学会研究報告, 2007-ARC-174, pp. 169-174, 2007 年 8 月. (2007/8/14, 旭川市)

12. 田中雄介、安藤秀樹、''物理レジスタ 2 段階解放による命令先行実行方式の評

価,'' 情報処理学会研究報告, 2007-ARC-174, pp. 163-168, 2007 年 8 月. (2007/8/14, 旭川市)

〔その他〕
ホームページ等

http://www.ando.nuee.nagoya-u.ac.jp/

６．研究組織
(1)研究代表者
安藤　秀樹 (Hideki Ando)
名古屋大学・大学院工学研究科・教授
研究者番号：40293667

(2)研究分担者
(　　)

研究者番号：

(3)連携研究者
(　　)

研究者番号：