2パス限定投機システムPALSの評価環境 – システムシミュレータ –

十鳥 弘泰† 福田 明宏† 大津 金光† 横田 隆史† 馬場 敬信†
†宇都宮大学大学院工学研究科情報システム科学専攻

1 はじめに

プログラムのループ中には，条件分岐によって複数の実行経路 (以下，パス) が存在する．次に実行されるパスを予測し，マルチコアプロセッサ上で並列に実行することによりプログラム実行の高速化を期待できる．

我々は，多くのループにおいて高々2本のパスが支配的な割合を占めることを明らかにし，これら2本のパスを投機的に並列実行することで高速化を達成する2パス限定投機方式を提案した [1]．さらに，同方式をハードウェア上で実現するためのマルチコアプロセッサシステムである，2パス限定投機システム PALS を開発している [2]．

本稿では，同方式および PALS の性能評価を行うため，PALS の動作をクロックレベルで正確に模擬するシステムシミュレータについて述べる．

2 2パス限定投機方式

一般に，ループ中には複数の条件分岐が存在する．実行される可能性のあるパスは条件分岐の数に応じて指数関数的に増大するため，全てのパスを対象とした予測を行うのは難しい．しかし，実際に実行されるパスには偏りがあり，多くの場合においてごく少数のパスが支配的である．これらのパスを，実行頻度の高い順に#1パス，#2パスと呼ぶ．

従来研究 [1] では，SPEC CPU95 ベンチマークのいくつかのプログラムのループ中におけるパスの実行割合を調査し，#1パスと#2パスの実行が全体の約8割以上を占めていることを明らかにすると共に，上位2つのパスを予測の対象とし，投機的に実行することでループの高速化の達成を期待できることを示した．

2パス限定投機方式では，事前にプログラムの実行挙動を解析しておき，その情報を基に#1パスおよび#2パスを決定し，それぞれのパスに沿ってプログラムコードの抽出を行う．そして，これらのコードに対して命令のスケジューリングや不必要な命令の除去等を行い，投機実行に特化したコード (以下，投機スレッドコード) を作成する．投機スレッドコードでは，元のプログラム中の条件分岐命令を assert 命令に置き換える．assert 命令により，実行中のスレッドがパスに沿った分岐を行ったか否かを判定することができる．投機が失敗した場合，スレッド実行前の状態へ回復処理を行った後，実行していないもう一方のパスを実行する．さらにそのパスでも投機が失敗した場合，元のループの逐次コードを実行する．

3 2パス限定投機システム PALS

広範なプログラムに対する2パス限定投機方式の詳細な評価を行うためには，現実的なハードウェア構成を想定したアーキテクチャが必要となる．PALS (PAth Limited Speculation) は同方式を実現するマルチコアプロセッサアーキテクチャである．

3.1 PALS のハードウェア構成

図1に PALS のハードウェア構成を示す．マルチスレッド制御機構 (Thread Management Unit: TMU) は，内部にパス予測を行う path predictor を持ち，全てのスレッド実行機構 (Thread Unit: TU) に対してスレッド制御を行う．TU は従来のプロセッサに相当する機構であり，TMU から受け取ったパス予測結果を基に，該当するパスの投機スレッドコードを実行する．

TMU と TU は双方向に通信を行う．通信の内容は，スレッドの制御や，投機実行の成否といった通知であり，データの通信は行わないため多くの帯域を必要としない．また，隣接する TU 間は双方向通信を行うリング構造となる．TMU がリングに沿って TU に順次スレッドを割り当てることにより，マルチスレッド実行を実現する．TU 間で実現するレジスタ間通信では，後続スレッド方向にのみデータが送られ，先頭スレッド方向にデータを送ることは無い．そのため，後続スレッド方向への通信に必要な帯域に比べ，先頭スレッド方向への通信は多くの帯域を必要としない．

Memory Buffer (MB) および Load Shelter (LS) は，投機的なメモリアクセスを適切に処理するための機構である．PALS では，MB と LS を併せてメモリアクセス機構と呼ぶ．TU と MB は1対1の関係で接続され，TU は全てのメモリアクセスを MB に対して行う．また，隣接する MB 間は TU と同様のリング構造となる．LS は MB の記憶領域が不足した際に使用する補助的な記憶装置であり，全ての MB とバス接続される．

3.2 PALS におけるプログラム実行

PALS では，投機の対象であるループ以外の逐次処理については，従来のプロセッサと同様にシングルスレッド実行を行う．投機スレッドコードの作成時に PALS 独自の命令である start2path 命令をループの開始位置に挿入し，シングルスレッド実行時に本命令を実行すると，PALS はマルチスレッド実行モードとなり2パス限定投機方式に基づいて投機実行を開始する．その後，ループの終端に挿入される stop2path 命令を実行するとシングルスレッド実行モードに戻る．PALS は，これらの実行モードを切り替えながら実行する．

An Evaluation Environment for Two-Path Limited Speculation System PALS – System Simulator –
†Hiroyoshi Jutori, Akihiro Fukuda, Kanemitsu Ootsu, Takashi Yokota and Takanobu Baba
Department of Information Systems Science, Graduate School of Engineering, Utsunomiya University (†)

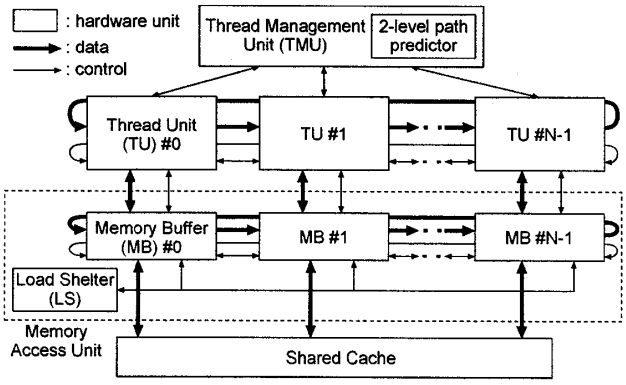

図 1: PALS のハードウェア構成

4 システムシミュレータ

4.1 概要

PALS が 2 パス限定投機方式を正確に実現できているかどうかの検証や，実際のプログラムに対してどの程度の高速化を達成できるかを評価するためには，シミュレーションソフトウェアを実装することが現実的である．本節では，PALS および 2 パス限定投機方式の評価を行うための環境として，PALS の動作をクロックレベルで模擬するシミュレーションソフトウェアを構築する．以降，アーキテクチャの PALS と区別するため，PALS シミュレータを *pals* と表記する．

PALS において，TU は分岐予測や out-of-order 実行といった汎用プロセッサとしての機能を持つ．汎用プロセッサのシミュレーションソフトウェアは既に多く公表されているため，この部分を新たに開発するメリットは少ない．一方で，TU は汎用プロセッサにはない投機的なスレッドを制御するための機能を併せ持つ．このため，TU を実現するにあたっては，ベースとするシミュレーションソフトウェアの構造を熟知した上で必要な機能を追加する必要がある．また，TMU，MB および LS は，PALS 独自の全く新しいハードウェア機構となっているため，これらの機構は新規に実装を行う必要がある．このため，*pals* のベースとするシミュレーションソフトウェアとしては，高性能なマルチプロセッサシミュレーションが可能であることと，拡張が容易であることが求められる．

これらのことから，*pals* の開発のベースとして，ISIS-SimpleScalar[3] を用いる．ISIS-SimpleScalar では，プロセッサやキャッシュをユニットと呼ばれる独立した単位で扱うことができ，各ユニットに同一のポートクラスを持たせ，ポート上でやり取りを行うパケットを定義することでユニット間での通信が実現できる．

4.2 シミュレータの実現方法

TU における独自の機能を実現するには，ISIS-SimpleScalar における既存のプロセッサユニット（以下，既存プロセッサ）に変更を加える必要がある．そこで，TU では既存プロセッサを構成するクラスを継承することにより，既存プロセッサへの変更を極力抑えつつ独自の機能を実装した．

TMU，MB および LS は，PALS 独自のハードウェア機構であるため新規に実装し，既存プロセッサと共通のポートクラスを持たせることで，変更箇所を最小限に抑えながら既存プロセッサとの通信を実現する．図 2 にユニット間接続の概念を示す．TMU から TU への送信，また，TU から TMU への送信を行うポートクラスをそれぞれ用意し，各ユニットクラスに持たせる．そして通信内容を示すクラスを定義することで，各ユニット間で様々なデータを通信することができる．

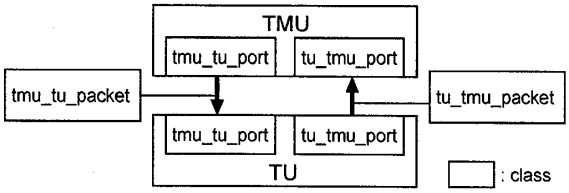

図 2: ユニット間接続の概念

既存プロセッサ内部に対する大きな変更点としては，プロセッサのメモリアクセスに関する処理がある．TU では全てのメモリアクセスを MB に対して行う必要があるが，ISIS-SimpleScalar では各プロセッサが独自に利用するデータは全てプロセッサ内部のローカルメモリに格納される．このため，既存プロセッサにおけるローカルメモリへのアクセスを行うか否かを判定する条件分岐の結果が必ず偽となるよう変更し，この分岐先において MB へのポートにメモリアクセスを発行することで，投機的なメモリアクセスを実現した．

5 おわりに

本稿では，ループ内の実行頻度上位 2 本のパスに着目した投機的マルチスレッド実行によりプログラムを高速化する，2 パス限定投機システム PALS およびそのシミュレータ *pals* について述べた．

現在，*pals* の動作検証を行っており，小規模なプログラムを並列に実行できることを確認した．今後は，本システム上で SPEC ベンチマークプログラムを始めとした様々なプログラムを実行した場合に，どの程度の速度向上を達成できるかを検証し，また，どのようなハードウェア構成が最も性能を発揮することができるかについて検討する予定である．

謝辞

本研究は，一部日本学術振興会科学研究費補助金 (基盤研究 (C)20500047，同 (C)21500049，同 (C)21500050) および宇都宮大学重点推進研究プロジェクトの援助による．

参考文献

[1] 横田隆史ほか: “2 パス限定投機方式の提案”, 情報処理学会論文誌:コンピューティングシステム, Vol.46, No.SIG 16(ACS-12), pp.1-13, 2005.

[2] 十鳥弘泰ほか: “2 パス限定投機方式を実現するマルチコアプロセッサ PALS の提案”, 信学技報, Vol.109, No.319(CPSY2009-46), pp.19-24, 2009.

[3] 薬袋俊也ほか: “ISIS-SimpleScalar の実装”, 情報処理学会研究報告, 2004-Arc-160, pp.29-34, 2004.