

投機的マルチスレッドシステムにおけるスレッド動作の可視化

小室 貴弘[†] 十鳥 弘泰^{††} 大川 猛^{††} 大津 金光^{††} 横田 隆史^{††} 馬場 敬信^{††}

[†]宇都宮大学工学部情報工学科 ^{††}宇都宮大学大学院工学研究科

1 はじめに

我々は、マルチスレッド処理による高速化について、ループ中の実行経路(以下、パス)のうち、上位2つのパスに限定して投機マルチスレッド処理を行うアーキテクチャ、2パス限定投機システム PALSを開発している [1]。PALSシステムなどの投機的マルチスレッドシステムの開発では、投機的なスレッドの実行状況、投機の成功失敗、スレッド間通信の状況を把握することが、システムの動作検証や性能向上を図る上で極めて重要である。しかし、現在の PALSシステムでは、スレッド動作を確認するためにはトレースログを確認するしかなく現実的ではない。そこで、今後のシステム開発に向けて投機的マルチスレッドシステムのスレッド動作を明確化することを目的とし、これを解決する手段として、スレッド動作の可視化ツールを構築する。

2 2パス限定投機方式

2パス限定投機方式とは、プログラム中のループ構造に注目し、ループの実行頻度が最も高い上位2本のパスに限定して投機的にマルチスレッド処理を実行する手法である。この方式では、上位2つのパスそれぞれに特化した投機スレッドコードと、ループ構造をそのまま保持している非投機スレッドコードの計3種類を用意する。プログラムの実行の際には、どちらの投機スレッドコードを実行するのかを予測し、対応するコードを投機実行する [2]。

3 2パス限定投機システム PALS

PALSシステムではパスの予測を Thread Management Unit(以下、TMU)が行い、各 Thread Unit(以下、TU)は TMU のパス予測に基づいて投機コードを実行する [1]。

PALSシステムのスレッド開始の処理は、全ての TU を管理する TMU によるパスの予測に基づいて行われる。TU は TMU のパス予測の結果から、パスの投機コードを実行する。

また、TMU によるパス予測が失敗する場合がある。PALSシステムではパス予測の失敗が検知されると、スレッドの回復処理を行い、予測失敗したスレッドを破棄する。このときプログラムの整合性を保証するために、後続スレッドも同様に破棄する。スレッドを破

棄してプログラムの回復処理を行った後、もう一方のパスの投機コードを実行する。後続スレッドは、新たにパス予測を行い、順次スレッドを実行する。

また、スレッド間通信の動作には、あらかじめ指定されたレジスタに関して、先行するスレッドからレジスタ値が送られてくるまでスレッドが待機状態に入る機能がある。これによってデータの依存違反を発生させずに通信が可能となる。

メモリ依存違反が起きた場合もスレッドの再実行を行う必要がある。これは、先行するスレッドで値を変更しようとしているデータが既に後続スレッドで使用していることによる依存違反である。これが検知されると、後続のスレッドは実行を破棄して回復処理に入る。回復処理を行った後、それまで実行していたパスを再実行する。

4 スレッド動作の可視化

PALSシステムでのスレッド動作の可視化について検討する。

PALSシステムはプログラムの高速化を目指していることから、パスの予測ミスや依存違反によるサイクルの損失の大きさや、スレッド間通信による他スレッドの動作への影響等の情報を明確化する必要がある。これらの情報が分かることで、システム開発における高速化の余地の検討や、どのような条件でサイクル損失があるのかを視覚的に理解できる等のメリットがあると考えられる。

動作内容は単純な図形と文字で描画出力することで実現することを検討した。図形描画することにより特定の動作でかかるサイクル数の大きさや、通信による別のスレッドへの影響などの、トレースログ上の文字列では分かりにくい情報が明確化できる。このことから、可視化はツールとして実現し、その上で図形描画させる形で可視化を実現する。

また、PALSシステムを評価するものとして、システムの動作を模擬するシミュレータ palsがある。可視化はこのシミュレータ pals によるプログラムの実行によって出力されるトレースログの情報を利用する。手法としては、トレースログの情報から各 TU 毎にサイクル単位の動作情報を得て、TU 毎の動作データをひとまとめにしたものから可視化を行う。この一連の動作を可視化ツールに実装する。この流れを簡略化したものを図1にて表す。なお、TU[0] から TU[N-1] はそれぞれ個別の TU で、N は TU 数である。

これらのことより、可視化の要件は以下の通りである。

- スレッドの生成・終了のタイミングを表示、かかっ

Visualization of Path Limited Speculation System

[†]Takahiro Komuro, ^{††}Hiroyoshi Jutori,

^{††}Takeshi Ohkawa, ^{††}Kanemitsu Ootsu,

^{††}Takashi Yokota and ^{††}Takanobu Baba

Department of Information Science, Faculty of Engineering,
Utsunomiya University ([†])

Graduate School of Engineering, Utsunomiya University
(^{††})

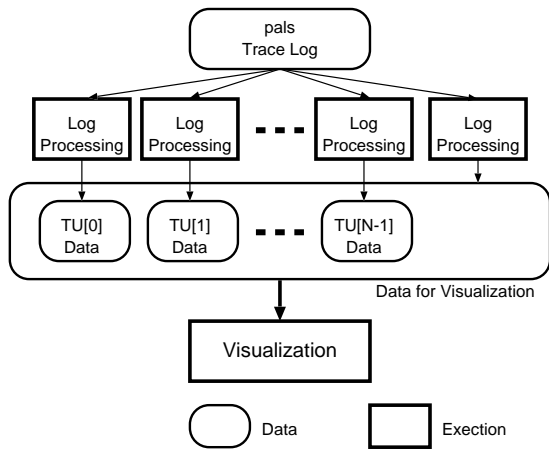


図 1: トレースログ処理から可視化まで流れ

たサイクル数の大きさを表す

- スレッド間通信の内容、及びスレッドへの影響を明確化
- パス予測の失敗などの処理によるサイクル損失などの大きさを図形表示できるようにする

なお、可視化ツールは GUI による実装が容易であること、Linux をはじめとする複数の環境で使用できる等のメリットがあることから、Java 言語で実装した。

5 可視化の例

テスト用のプログラムをシミュレータ pals 上で実行し、取得したトレースログを用いて可視化ツールを使用した。図 2 に TU 台数を 4 台として、可視化ツールを使用したときの実行結果を示す。図の左から順に、TU[0], TU[1], TU[2], TU[3] (これらも、それぞれ個別の TU) の動作状況を表す。縦軸が時間軸を表しており、単位はサイクル数である。向きは下方向である。

スレッドの動作内容に関してブロック状に出力し、色分けをしている。赤色が #1 パスを、青色が #2 パスを実行していることを示しており、これらの情報からスレッド動作の内容を知ることができる。黄色のブロックはパスの予測ミスなどによる回復処理中であることを表している。これらの動作内容を表すブロックの大きさから、どの動作でかかっているサイクルの大きさが分かり、サイクル数の大きさの比較も可能である。また、マウスオーバーすることで正確なサイクル数が表示されるようにしている。

スレッド破棄によるスレッド内のサイクル損失は×印で表しており、これの正確な損失分のサイクル数も、マウスオーバーで同様に表示される。

スレッド間の通信は矢印で表し、矢印の色から通信内容を知ることができる。なお、黒色の矢印はレジスタ値の通信を指しているが、送信しているレジスタ番号を \$記号と組み合わせる表記している。現在実装している通信内容の可視化は、スレッド終了の通知 (水

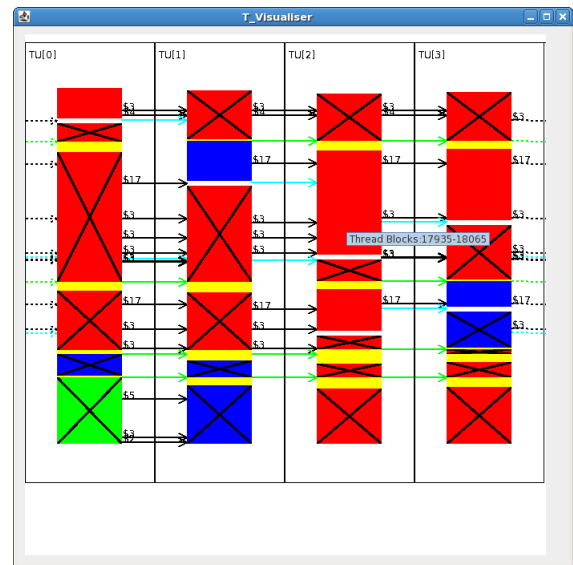


図 2: 可視化の例 (TU4 台)

色)、パス予測ミスの発生通知 (緑色)、メモリ依存違反通知 (橙色) の計 4 種類である。この通信内容と、スレッド動作の状態から、他スレッドへの影響が確認できる。例えば、TU[1] の最初のスレッド動作では、#1 パスを実行中にスレッド動作の破棄が発生し、以降のスレッドに対してパス予測ミスの発生通知を送っている。その通信内容を受けて、後方スレッドが同様にスレッドを破棄していることが確認できる。

6 おわりに

本稿ではマルチスレッドシステムを対象に、高速化に向けたシステム開発の面からスレッド動作を可視化する手法として、可視化ツールの実装を検討した。ツールの画面上で、スレッド動作の内容やスレッド間の通信による動作の影響等が視覚的に表示され、把握しやすくできるようにした。

今後の課題としては、システム開発に関して必要な情報を追加し、それらの情報のツールへの対応や、スレッド動作の可視化以外にパイプラインの可視化等が挙げられる。

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)21500050, 同 (C)21500049) の援助による。

参考文献

- 十島弘泰ほか: “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.
- 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.