

## 動的なスケジューリングを行うマルチスレッドプロセッサの提案

3H-6

齋藤誠一郎 前田敦司 曾和将容

電気通信大学 情報システム学研究科 情報ネットワーク学専攻

## 1, はじめに

現在最も普及しているスーパースカラアーキテクチャは、従来のプログラムとの互換性を維持したまま命令レベルの並列性を抽出でき、高速化することができる技術である。しかしながらこの方法で取り出せる命令レベルの並列性には限界があり、このままでは設計コスト・ハードウェア量に見合うだけの性能向上を望めなくなってしまう。その一方、複数のプロセッサを利用して並列に処理を行うという方法もあるが、従来の単体のプロセッサと互換性が取れない場合もあり、ソフトウェア資産の無駄になってしまう。

本稿では、オンチップマルチプロセッサを前提として、要素プロセッサのアーキテクチャを選ばず、従来のプログラムとの互換性を維持したまま並列実行が行えるアーキテクチャを提案する。

## 2, 特徴

本稿のアーキテクチャは、ひとつのプロセッサはプログラムの実行中にループを探索し、そのループ情報から各プロセッサにループのイタレーションを割り当てて並列に実行させる。ループの発見、プロセッサへの割り当てと各イタレーション間の依存性の解決はハードウェアで行われコンパイラの支援を必要としない。プロセッサはリング状に接続され、片方向へのみ通信を行う。これによりプロセッサ間の通信のための無駄なオーバーヘッドを必要としない。同じようにループを動的に発見するマルチプロセッサアーキテクチャとして[1]があるが、スレッド生成タイミング、レジスタの継承、依存関係の解消の方法が異なる。

## 3, ループの検出・イタレーションの重複投機実行

並列に実行されるループは、プログラムの実行時に動的に発見される。ループは後方へのブランチあるいはジャンプをもって、制御命令によってジャンプした先のアドレスからその制御命令のアドレスまでをループの本体としている。ループの情報はプロセッサ間を順々に伝わっていき、情報を得た各プロセッサは投機的にイタレーションを実行する。各イタレーションは重複して実行される。

イタレーションを投機的に実行しているプロセッサは、演算結果を自プロセッサ内のみにはしか反映させない。メモリ等へ書き出すことのできるのは、投機実行を行ってないプロセッサだけでそれは1つだけである。それをここで確定状態と呼ぶ。ループ実行時に、確定状態のプロセッサがループの条件式確定後、次のイタレーションが有効であるとわかったなら、それを投機的に実行しているプロセッサを確定状態にする。無効であるとわかったなら、そのプロセッサを停止させ、演算結果をすべて破棄する。

#### 4, レジスタ・データメモリの継承

ループの情報を得たプロセッサは、イタレーションを実行するのに必要なデータを前もって揃えておく必要がある。このデータはループの情報を送ってきたプロセッサのデータに最も近い。他の並列アーキテクチャ[2][3]ではイタレーション実行時にすべてのレジスタを継承する方法や、生成されるデータを任意に転送する方法などを用いているが、これらの方法はコンパイラによる支援がある程度必要であるため従来のソフトウェアとの互換性がとれない。本稿ではこのレジスタの継承を、書き込みの発生したレジスタの値を常に他のプロセッサに転送する方法を用いている。これによりプロセッサがループの情報を得た時点で、情報を送ってきたプロセッサと同じレジスタ情報をオーバーヘッド無しで持つことができる。

並列化していないプログラムは、動的な並列化によって本来は解決されているべき依存関係ができてしまう可能性がある。投機的に実行しているプロセッサはレジスタの情報が足りないまま見切り発進をしてしまうため、ループの情報を送ってきたプロセッサからあとから書き込みの起きたレジスタの情報が送られてくる場合がある。その時投機実行をしているプロセッサがそのレジスタをすでに読み込んでいたなら間違った実行を行ったということになるので、レジスタを読み込んだ位置まで命令を巻き戻す必要がある。そのために、イタレーション実行時にレジスタの書き込み・読み込み情報を記録する。送られてきたレジスタの情報とその記録をもとに命令を巻き戻す。

また本稿のアーキテクチャはロード命令を投機的に実行するためデータメモリの継承も行う。ストア命令の情報は投機実行を行っている他のプロセッサへ伝えられ、場合によっては命令を巻き戻して再実行する。

#### 5, おわりに

本稿ではループを動的に発見し、各イタレーションを投機的に重複して実行するマルチプロセッサアーキテクチャを提案した。このアーキテクチャは、従来のプログラムとの互換性を維持したまま複数のプロセッサによる並列実行が可能である。今後は、提案したアーキテクチャのシミュレータを作成し性能の評価を行う予定である。

#### 参考文献

- [1] 玉造潤史, 松本尚, 平木敬: Loop を並列実行するアーキテクチャ, 情報処理学会研究報告, 96-ARC-119, pp.61-66, (1996年8月)
- [2] 鳥居淳, 近藤真己, 本村真人, 池野晃久, 小長谷明彦, 西直樹: オンチップ制御並列プロセッサ MUSCAT の提案, 情報処理学会論文誌, vol39, no6, pp1622-1631, (1998年6月)
- [3] 小林良太郎, 岩田充晃, 安藤秀樹, 島田俊夫: 非数値計算プログラムのスレッド間命令レベル並列を利用するプロセッサ・アーキテクチャ SKY, JSPP '98, pp87-94, (1998年6月)