

## 値予測を行う制御フロー並列実行に関する基礎検討

3H-9

安里 彰<sup>\*</sup> 鈴木 和宏<sup>\*</sup> 池田 吉朗<sup>\*\*</sup> 木村 康則<sup>\*</sup><sup>\*</sup> 新情報富士通研<sup>\*\*</sup> (株)富士通研究所

### 1 研究の背景

SPECベンチマークをはじめとするプログラムには理論的に非常に大きな並列性が内在しているという研究報告がある[1]。こうした並列性を最大限に引き出すためには、プログラムをある程度の粒度を持ったブロックに分割し並列実行する、いわゆる制御フロー並列実行を最適に行なう必要がある。しかし、そのための具体的な方法に関して、いまだに決定的な方式は確立されていない。

一方、投機実行の一分野である値予測に関する研究が近年活発化している[2][3]。値予測は真依存によるストールを回避できるため、大きなブレイクスルーとなる可能性があると考えられる。しかしながら、これまで提案されている予測方式では的中率が不十分なことや、対象としているブロックが細粒度であるため予測的中によって得するサイクル数が小さいことなどから、大きな性能向上を得るには至っていない。

我々はこのような背景の下、制御フロー並列と値予測を組み合わせることにより、更なる性能向上が図れると考え、研究に着手した。

### 2 我々のアプローチの予備評価

我々のアプローチの効果を調べるために予備実験として、値予測が100%的中する仮定の下で、プログラムを基本ブロック単位に並列実行させるシミュレーションを行った。基本ブロックのPEへの割り当ては、下記の制約を満たし、かつ空いているPEがある限り、真依存を無視して、プログラム順に基本ブロックを次々に実行させるようにした。

制約1 同一アドレスへのストアは順序を変えない。  
制約2 同一アドレスへのロード値予測は高々1個。

制約3 投機的ブロック実行をしたPEには、それが投機的でなくなるまで、新たに割り当てない。

制約3で「投機的でなくなる」とは、本来それよりも以前に実行すべきブロックが全て終了することを意味する。この制約は値予測が外れた時のリカバリ処理を考慮したものである。

評価は、方式A(制約1のみ適用)、方式B(制約1,2を適用)、方式C(制約1,3を適用)の3通りについて、我々が開発したSPARCベースのSMPシミュレータMetatool[4]で行った。個々のPEは4並列スーパスカラとし、キャッシュは100%ヒットとした。ベンチマークはSPEC92からgccとeqntottを用いた。

1PEでの逐次実行に対して、複数PEで投機的に並列実行した場合の性能向上を図1に示す。

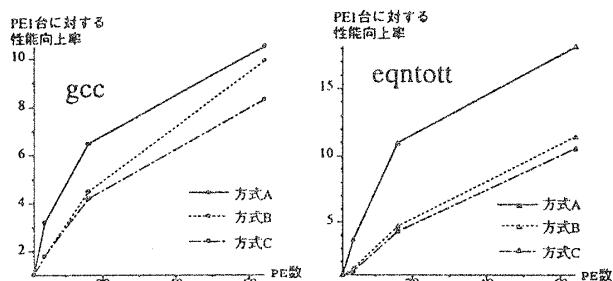


図1: 各方式の性能向上率

方式AではPE64台で10倍以上の性能向上が得られた。また、制約2,3による性能低下は10%~40%となった。ただし、今回の実験ではブロックの割り当てを実行順としたため、十分に並列度を抽出しきれていないことが考えられる。よりアグレッシブな割り当て手法を用いて、再評価を試みる予定である。

### 3 おわりに

制御フロー並列に値予測を取り込むアプローチを紹介し、予備評価として値予測が100%的中した場合の性能向上率を示した。今後はマイクロアーキテクチャとコンパイラの両面から、より具体的な手法の検討を進めていきたい。

### 参考文献

- [1] M.S.Lam et. al. "Limits of Control Flow on Parallelism", ISCA'92, pp.46-57, 1992.
- [2] H.Lipasti et. al. "Exceeding the dataflow limit via value prediction", MICRO'97, pp.227-237, 1997.
- [3] 佐藤寿倫. "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.
- [4] 鈴木他, "トレースベースSMPシミュレータ Metatool の開発", 第58回情処全大, 1999.