

## 投機実行の動的なスケジューリング技法

3H-7

宋丕尤 山内斉 前田敦司 曾和将容

電気通信大学 情報システム研究科 情報ネットワーク専攻 分散処理研究室

### 1. はじめに

現在の情報処理は複雑多岐に渡る範囲に及び、様々な分野で高性能な計算機が必要とされている。これまでに、計算機の高性能化のために同時に実行可能な命令列を動的に発見し、並列実行を行うことによって性能を向上させる並列コンピュータが提案されている。

しかし、プログラム中の分岐命令の存在により、同時に実行可能な命令列は制限されることがこれまでの研究から判明している [2]。この問題に対して、分岐命令の実行が完了する前に、分岐が実行された場合と、そうでなかった場合の両方の命令列を実行し、後に分岐方向が確定した時点で間違った方向の結果を捨てるという投機的実行法が提案されている。

投機的な実行により、並列コンピュータをさらに効率的に利用することは可能であるが、そのためにハードウェア量は増加し、複雑になるという問題点がある。そこで本論文では、コンパイル時に並列性を抽出し、コンパイラと協力した並列コンピュータ上で投機的実行を行なう方法を提案する。

### 2. 提案の概要

#### 2.1 提案の特徴

本論文で提案する投機的実行スケジューリングには3つの特徴がある。第1に静的に並列性を抽出すること、第2に実行時に動的な命令のスケジューリングを行うこと、そして第3に、分岐条件が判明する前に、不必要となる可能性のある分岐命令を投機的に実行することである。本提案は、並列コンピュータの各命令の実行時間は同じと仮定している。

#### 2.2 提案の方針

プログラムの実行時間をできるだけ短縮するため

に、アイドルプロセッサがあれば、データ従属性によって実行できる命令をコントロール従属性と関係なく実行します。そしてコントロール従属性によって修正してから、また、データ従属性によって実行する。

#### ①DFG,CFG とプライオリティリストの作成

・図2, 3に示しているように図1のプログラム中のデータ依存関係によりDFG(Data Flow Graph)を作成する、ノードは命令で、エッジはデータ依存関係を表す。プログラムフロー(コントロール依存関係)からCFG(Control Flow Graph)を作成する、ノードは命令で、エッジはプログラムフローを表す。

```

1: c=a+5;
2: e=c+d;
3: if e>0
  then
4: {f=e+1;
5: k=c-1;
6: l=f+k;
7: i=l+3;}
  else
8: {f=e+2;
9: k=c-3;}
10: g=e+e;
11: h=k+3;
    
```

図1:元のプログラム

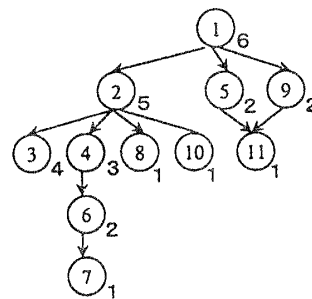


図2:DFG

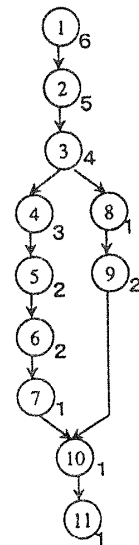


図3:CFG

優先順位	1	2	3	4	5	6	7	8	9	10	11
タスク番号	1	2	3	4	5	6	9	7	8	10	11

図4:プライオリティリスト

Dynamic Scheduling of Program by Speculate Execution

Song Peiyou, Yamauchi Hitoshi, Maeda Atusi, Sowa Masahiro

University of Electro-Communications

- ・DFGとCFGから各ノードのCCP長(Control-Critical Path)を求める。
  - a. 出口ノード(後継ノードがないノード)のCCP長は1とする。
  - b. 分岐条件命令ではない命令ノードのCCP長はDFGの中の後継ノードの最も長いCCP長+1とする。
  - c. 分岐条件ノードのCCP長はCFGの中に後継ノードの最も長いCCP長+1とする。
- ・図4に示しているように命令CCP長の長いほど、優先順位が高いようにプライオリティリスト(priority list)を作る。

### ②動的スケジューリング

・並列コンピュータの持つ複数のプロセッサのうち一つのプロセッサをSP(Scheduling Processor)とする, 他はEP(Executing Processor)とする。

・SPはプライオリティリストを参照しながらでデータとコントロール従属性に従って実行可能な命令を探し, 実行できる命令にマークをつける。条件命令の実行結果により, 実行必要のない命令をプライオリティリストから取り除き, プライオリティリストを修正する。

・EPはプライオリティリストから, 優先順位に従って実行できる命令を取り出し, 実行する。

### ③分岐の場合:

プログラムの実行時間をできるだけ短縮するため, 分岐を超えて実行可能になった命令が存在し, たつ, アイドルプロセッサがある場合に, その命令を優先順位に従って投機的にアイドルプロセッサに割り当てる[1]。このとき, 確定していない2つの命令ストリームから交互的に実行するようにアイドルプロセッサに割り当てる。投機実行の命令の結果は一時的なバッファに書く。分岐命令が確定すると, 一時的なバッファの正しい方の結果をメモリに書く, 誤りであった分岐先命令をプライオリティリストから取り除く。

### ④ループの場合:

1つのプロセッサはあるループの1つのイタレーションを実行する, データ依存関係がない場合, N回目のイタレーションはNクロック目に実行を開始する。あるプロ

セッサが1つのイタレーションを終えたらアイドルプロセッサなり, 次の実行可能なイタレーションが割り当てられる。

### 3. 終わりに

本研究では, 非数値計算のプログラムを多数のプロセッサで効率的に実行するため, 投機実行の動的なスケジューリング方法を提案した。プログラムを並列実行する時, 命令データ従属性によって実行し, そしてコントロール従属性によって修正してから, また, データ従属性によって実行する。

今後の課題としては, たとえばコントロールフロー予測, 一時的なメモリの設計法などがある。

### 参考文献:

- 1) 葛西 信行, “並列度に依存しないタスクスケジューリング技法”, 平成9年度電気通信大学大学院 修士論文
- 2) Simonjit Dutta and Manoj Franklin, “Control Flow Prediction Schemes for Wide-Issue Superscalar Processors” *IEEE Trans. Parallel and Distributed Systems*, vol.10, no.4, pp.346-359, Apr. 1999.