

# 58 JAVA による PRAM コンパイラの作成

情報論理工学研究室：板東俊助 共同研究者：池田直樹 神谷道利

## 1. 序論

並列アルゴリズムの設計およびその計算量の解析は多くの場合 PRAM(Parallel Random Access Machine) 上で行われるが、それを実現するには非常に困難である。そこで本研究では、並列アルゴリズムの設計およびその計算量の解析の容易化を支援するために PRAM シミュレータの一部である PRAM コンパイラを設計する。

## 2. 研究内容

PRAM シミュレータは、PRAM コンパイラ、高級言語、アセンブラ、VSM ( Virtual Stack Machine ) これらの4つからなるが、本研究では、この PRAM シミュレータの一部である PRAM コンパイラを作成する。

本研究では、高級言語として k05 言語、アセンブラとして VSM アセンブラを用いる。本研究では、k05 言語に以下の PRAM 上での並列処理を行う命令“ parallel 文”および特殊記号“ \$p ”を加えた。parallel 文の文法は、「parallel(式 ,式 )文」という構成であり、プロセッサ番号式 から式 のプロセッサを用いて後ろに続く文の並列処理を行う命令である。また“ \$p ”を“ parallel 文”実行中のプロセッサ番号を表す記号である。

また、本研究では VSM アセンブラに以下の3つの命令、プロセッサを並列に処理する開始命令である“ PARA ”、並列処理の終了とプロセッサ間の同期を表示する命令“ SYNC ”、プロセッサ番号をスタックに入れる命令“ PUSHP ”を加えた。

さらに本研究では、拡張した高級言語 k05 言語をアセンブリコードに変換するコンパイラを作成した。本研究で作成したコンパイラで変換された VSM アセン

ブラを PVSM ( Parallel Virtual Stack Machine ) で実行することにより、PRAM アルゴリズムの実行をシミュレートでき、PRAM 上で実行させた際の実行時間を計測できる。これにより逐次状態と並列状態の PRAM アルゴリズムの計算量を実験的に評価できる。

## 3. 結果・考察

以下に本研究での結果を示す。

図1は0番から15番までのプロセッサ番号を出力する、PRAM 用並列言語プログラムである。本研究で作成した PRAM コンパイラを用いることにより、図1のプログラムは図2に示す VSM アセンブラに変換される。図2の VSM アセンブラを PVSM で実行することにより、図3の実行結果が得られ、図1のプログラムの実行にかかるステップ数が7であると計測される。

```
main() {  
  parallel (0,15) {  
    write ($p);  
  }  
}  
  
PUSHI 0  
PUSHI 15  
PARA  
PUSHP  
OUTPUT  
SYNC  
HALT
```

図1 PRAM 用並列言語 図2 VSM アセンブラ

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
Execution time : 7
```

図3 PVSM 実行結果

## 4. 結論

本研究で PRAM コンパイラを作成したことにより、PRAM アルゴリズムをシミュレートできる。およびその計算量の解析の容易化を支援することができる。また、このコンパイラを用いて PRAM アルゴリズムの実行にかかる時間を計測できる。