

1. 序論

並列アルゴリズムの計算量の評価を行うには、PRAM(Parallel Random Access Machine)が必要とされる。しかし、実際に PRAM を実現し計算量の評価を行うことは困難とされる。そこで本研究では、PRAM 上での計算量の評価を実験的に行うためのツールとして、PRAM シミュレータの一部である PRAM コンパイラの作成を行う。

2. 研究内容

PRAM シミュレータは PRAM アルゴリズムを使用するための高級言語とアセンブラ、高級言語をアセンブラに変換するコンパイラ、そしてそのアセンブラを実行する PVSM (Parallel Virtual Stack Machine) からなる。本研究は、高級言語に K05 言語、アセンブラに VSM アセンブラを用いる。それらを用いて PRAM 用の並列プログラムを記述するために、K05 言語に並列命令 `parallel` 文と特殊記号 `$p` を加え、また、VSM アセンブラの命令セットに `PARA`・`PUSHP`・`SYNC` を加えた。さらに、これらを用いて、拡張 K05 言語から拡張並列アセンブラを生成するコンパイラを作成する。

`parallel` 文の文法は以下の通りである

```
parallel(式 ,式 )文
```

式 ・ は並列計算をする際に使用するプロセッサ番号であり、式 ~ のプロセッサを用いて以降に続く文を並列に実行する。さらに、特殊記号 `$p` はその記号を用いた場所における実行時のプロセッサ番号を表す。

アセンブラの命令セットに加えた `PARA`・`SYNC`・`PUSHP` については以下の意味を表す。`PARA` とはそれ以降での並列計算の開始を表す。`SYNC` は各プロセッサが `SYNC` を読むまで動作を停止し、全てがその命令を読むと再び動作を開始する、すなわち各プロセッサで同期をとる意味を表す。`PUSHP` は、各プロセッサがプロセッサ番号をスタックトップにつむという意味を表す。

3. 結果・考察

図 1 は拡張 K05 言語を用いて記述したプログラム例である。これは、1 から 15 までのプロセッサを使用し各プロセッサにプロセッサ番号を表示させるプログラムである。図 1 のプログラムを本研究で作成したコンパイラでコンパイルすることにより図 2 に示す VSM アセンブラが得られる。さらに図 3 は、図 2 のアセンブラを PVSM で実行させたときの出力結果の一部であり、各プロセッサの出力と実行にかかったステップ数が出力されている。

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

図 1 : 並列プログラム

図 2 : 並列アセンブラ

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

図 3 : PVSM の実行結果

4. 結論

本研究では PRAM での実験的な評価を行うために、並列コンパイラと PRAM の動作を記述できる高級言語を作成した。本研究で作成した高級言語は PRAM アルゴリズムの動作を記述できる。また本研究で作成したコンパイラを用いることで PRAM アルゴリズムの計算量の実験的な評価を行うことができる。

5. 参考文献

- 1)平成 17 年度 情報・コンピュータシステム プロジェクト 指導書