

1. 序論

並列アルゴリズムの設計およびその計算量の評価は多くの場合 PRAM(Parallel Random Access Machine) 上で行われる。PRAM は共有メモリ型並列計算モデルであり、個々の演算による実行時間の違いや通信や同期のコストを無視した単純なモデルであるため、アルゴリズムの設計および評価を行い易いためである。しかし、大規模なプロセッサでのメモリの共有化や、通信や同期の高速化には様々な問題があるため、PRAM 自体の実現は困難である。そこで本研究では、PRAM シミュレータの一部である VPSM (Virtual Parallel Stack Machine) の設計を行う。

2. 研究内容

PRAM アルゴリズムの実行をシミュレートする PRAM シミュレータは以下の 4 要素から成る。

(1) PRAM 用並列言語 (2) 並列アセンブラ (3) PRAM コンパイラ (4) VPSM (Virtual Parallel Stack Machine)

PRAM 用言語は、C や JAVA のような人間にとって記述し易い高級言語であり、PRAM 上での並列処理を記述できる言語である。並列アセンブラは並列処理を記述できる低級言語である。PRAM コンパイラは PRAM 用並列言語を並列アセンブラに変換するコンパイラであり、VPSM は並列アセンブラの実行をシミュレートするインタプリタである。

本研究では、JAVA 言語を用いて VPSM の設計を行った。本研究で設計した VPSM は、以下の 3 要素から成る。

(1) 変数の値を保存するデータセグメント (Data Segment) (2) 並列アセンブラの命令を保存する命令セグメント (Instruction Segment) (3) それぞれ 1 個のデータスタックとプログラムカウンタを持つスタックマシン (Stack Machine)

各スタックマシンは、PRAM の各プロセッサの動作をシミュレートする。各スタックマシンは、プログラムカウンタが指す番地にある命令セグメントを読み込む。読み込んだ命令に応じてデータセグメントへのデータの読み書き、スタックへのデータの出し入れを行い、次の命令へ移る。VPSM は、逐次状態と並列状態の 2 つの状態を持つ。逐次状態では、1 台のスタックマシンのみが動作を行い、並列状態では PRAM 用並列言語プログラム中で指定された複数のスタックマシンが動作を行う。

並列アセンブラは、通常のアセンブラ命令に加えて並列処理の開始を表す命令 PARA、並列処理の終了とプロセッサ間の同期を表す命令 SYNC、プロセッサ番号をスタック

に入れる命令 PUSHHP を持つ。初期状態においては VPSM は逐次状態にある。命令 PARA を読み込むと並列状態に以降し、複数のスタックマシンが起動される。並列状態において命令 SYNC を読み込むと、全てのスタックマシンが SYNC に到達するまで処理を中断し、その後逐次状態に移行する。

```
PUSHI 1
PUSHI 10
PARA
PUSHHP
OUTPUT
SYNC
HALT
```

図 1 並列アセンブラプログラムの例

```
1 2 3 4 5 6 7 8 9 10
```

図 2 VPSM の出力結果

3. 結果・考察

図 1 は、10 台のプロセッサがそれぞれそのプロセッサ番号を出力する並列アセンブラプログラムである図 1 のプログラムを VPSM で実行すると図 2 の出力結果が得られる。

4. 結論

本研究では、VPSM の JAVA 言語を用いて設計を行った。本研究で設計した VPSM は、並列アセンブラの実行をシミュレートすることができる。また、PRAM 用並列言語を PRAM 上で実行させたときの実行時間を出力する機能を持ち、PRAM アルゴリズムの計算量を実験的に評価することができる。

参考文献

- 1) 加藤暢, "平成 17 年度第 5 セメスター 情報・システムプロジェクト I 指導書," 近畿大学理工学部情報学科, (2005)
- 2) 辻野嘉宏, "情報工学入門選書 10 コンパイラ," 昭晃堂, (1996)
- 3) 疋田輝雄, 石畑清, "コンパイラの理論と実現," 共立出版, (1988)