

6Y-6

関数型言語FLの並列処理方式

樋谷一 高橋義造
徳島大学

1. はじめに

近年、計算機の発達およびソフトウェアへの負担の増大に伴って、並列計算機および並列処理言語に注目が集まっている。関数型言語はプログラムの参照透明性により関数の並列実行が可能で、プログラムに内在する並列性を抽出し易い。そこで、並列処理を意識せずに書かれたプログラムを効率よく並列処理することを目標として、当研究室で開発した二進木並列計算機Coral68K^[1]上で、ハンダーソンが[2]において用いている純関数型言語(以下FLと呼ぶ)を拡張した関数型言語FLの並列処理系の設計、試作を進めている。

2. 関数型言語FL

関数型言語FLは、ハンダーソンのPFLに様々なinfix演算子、prefix演算子、および集合演算機能を付け加えた言語で、同様なプログラミング・スタイルとしてLandinのISWIM^[3]などがある。FLによるプログラム例を図1に示す。

```
qsort(x) =
  if x == nil then nil
  else {qsort([blb<-y;b<=a]) ++ [a]
        ++ qsort([blb<-y;b>a]) where a:y = x}
  図1 Quicksort program
```

'++'はappend関数、':'はcons関数のinfix演算子である。

3. 並列処理方式

現在、試作中のプロトタイプFLにおいては、関数の評価方式として、先行評価(eager evaluation)を用い、引数の並列評価のみを行う。しかし全ての引数を並列処理すると、並列処理の粒度があまりにも小さくなり、オーバーヘッドが大きくなるので、ユーザ定義関数についてのみ引数の並列評価を行う。FL第2版においては怠け評価(lazy evaluation)による、関数本体の並列実行を実現する予定である。

3.1 FLの処理方式

FL処理系では、次のようにホスト計算機上で仮想機械コードにコンパイルし、二進木並列計算機Coral68K上で仮想機械をエミュレートする。

```
ソースコード  ---> 仮想機械コード  ---> 実行
                  コンパイル                エミュレート
```

仮想機械としては、Landinの考案したSECD機械^[3]に並列処理機能を付け加えたものである。

3.2 データ構造

リスト構造を用いると処理系も簡潔になり、実行効率も良いと思われるが、プロセッサ間の通信においてはオーバーヘッドが大きい。ゆえに本処理系においては、通信のオーバーヘッドを緩和するために列(sequence)がデータ構造として採用されている。

3.3 SECD機械

Landinがその原型を考案したSECD機械の名前は、それを

構成する次の4つの主要レジスタの名前に由来している。
S (stack) 式の値を計算するときの中間結果を入れる。
E (environment) 計算の途中で各変数に束縛される値を入れる。
C (control list) 実行されるべき機械語プログラムを入れておく。
D (dump) 新しい関数呼び出しが起こったときに、他のレジスタの内容をしまっておくスタックの一種。
FLではその他に、どちらの方向のプロセッサからプロセスを受け取ったかを記憶しておくレジスタなどが必要となる。

3.4 SECD機械命令

基本命令21個、遅延評価を実行する命令3個、計24個の命令がある。機械語プログラムはS式で表される。

3.5 並列プロセスの発生

機械語命令AP(apply function)に出くわすと、周りのプロセッサが暇かどうか調べ、手が空いているプロセッサがあれば、そのプロセッサに現時点における環境および実行しようとしている関数へのポインタを渡すことにより、並列プロセスを発生する。プロセスを委ねたプロセッサはそのプロセスから返ってくる値を格納するSレジスタの場所にマークを付け、自分は次の実行可能なステップに進んでいく。一方、プロセスをもらったプロセッサはもらった方向をレジスタに記憶しておき、プロセスが終了したらもらった方向のプロセッサに割り込みをかけ値を返す。

3.6 プロセス・スケジューリング

並列実行可能なプロセスに出くわすと、top, left, right三方向のいずれかにプロセスの実行を委ねる。プロセス割当の優先順位によって、プログラムの実行時間およびプロセッサ間の通信回数が異なってくる。ゆえにプロセス割当の優先順位は調整できるようにしなければならない。

3.7 シミュレーション

フィボナッチ数を求める机上シミュレーションを行った。関数一つ当たりの処理をt時間とすると、フィボナッチ数6の場合、プロセッサ1台で実行すると38t時間かかり、プロセッサ7台使い、プロセス割当の優先順位をtop, left, own, rightとした場合12t時間、通信回数は10回、優先順位をtop, left, right, ownにした場合は11t時間、通信回数は14回であった。プロセッサ利用率は約50%弱であり、並列処理を意識せずに書かれたものとしてはまずまずの効率である。

4. おわりに

本報告においては、関数型言語FLの並列処理方式の概要を述べた。現在プロトタイプFLをインプリメント中である。

参考文献

[1]遠藤俊雄, 松尾賢二, 白方新洋, 樋谷一, 高橋義造 : 二進木マシンCoral68Kのシステム構成と性能評価, 計算機アーキテクチャ研究会66-5(1987-7)
[2]P. ハンダーソン : 関数型プログラミング, 日本コンピュータ協会(1985)
[3]P. J. Landin: The next 700 programming languages. CACM, 9, 157-164(1966)