

ストリーム通信に基づく並行プログラミング言語NET/C*

2J-1

中林 嘉徳 岩田 竜一 青柳 洋一 上原 稔 森 秀樹†

東洋大学工学部情報工学科‡

1 始めに

我々は、現在WSI上で八近傍格子型結合されたセルを持ち、効率の良い耐故障計算を行うアーキテクチャ、SOFT(Stream Oriented Fault Tolerant)の研究を行っている[1]。SOFTでは、目的の処理を三重化、もしくは、信頼度付きの一致によって得られるデータのパイプライン処理によって計算を実行するが、この時の処理は、WSI上に展開されたパイプラインのネットワークによって表されることが出来る。

そこで我々は、このようなネットワークモデルを簡潔に記述する並行言語として、NET/Cの開発を行った[2]。NET/Cの主な特徴は、

- アクターとストリームによって、ネットワークモデルを形作ることによる問題解決。
- ネットワークのノードである各アクターの並行動作による処理の高速化、効率化。
- C言語ライクの文法による、広く一般に理解されやすいプログラム記述。
- SOFTの性能を最大限引き出すことが出来る言語仕様。

といった点が上げられる。

2 NET/Cのモデル

NET/Cのプログラムの記述は、C言語のそれと極めて良く似ている。しかし、実行型のモデルは、アクターとストリームによって形成されたネットワークモデルであり、これはデータフローや関数型言語のリスト処理と同様の性質を持ちうる。

- ストリームは二つのアクター同士を結ぶ通信路である。これは一方向にのみ働き、その入出力の順序を保証する。

*A Stream Based Concurrent Programming Language NET/C

†Yoshinori NAKABAYASHI, Ryuichi IWATA, Youichi AOYAGI, Minoru UEHARA, Hideki MORI

‡Department of Information and Computer Sciences, Toyo University

- アクターは、複合アクターとプリミティブアクターとに分類できる。複合アクターは、ストリームの生成や結合、より細かな処理を行うアクターの生成を行うアクターであり、プリミティブアクターは1個以上のストリームを持ち、データの処理や、データの流れを制御する。これらのアクターは、プログラム実行時に生成され、それぞれ並行に動作する。ネットワークモデルは、このプリミティブアクターとストリームによって実装される。

3 NET/Cの言語仕様

NET/Cの言語仕様は、修得を容易にするため、C言語を模して設計されたため、その文法は非常に良く似ているが、実行概念が根本的に異なっているため、NET/C独自の記述もいくつかある。以下、NET/Cの仕様を列挙し、それらを説明する。

3.1 定数

NET/Cでは、C言語と同様の定数を持つ。3、3.5、'a'、"abc"はそれぞれ整数、実数、文字定数、文字列を表す。各定数はそれ自身を無限に生成するアクターを表すストリームの出力端である。その他、特殊な定数として、nil、EOF、sink、cin、coutなどがある。

3.2 型付きストリーム

NET/Cでは、すべての変数はあるストリームの入力端か出力端を表す。変数は宣言時に、そのデータの型と、ストリームの入力端(out)か出力端(in)のいずれか一方の指定を行う。また、特殊なデータ型として、すべての型と適合するany型がある。データ型の記述は省略可能で、その場合、自動的にany型と判定される。

変数への操作は代入と連結がある。代入は同じ型のストリームに別名をつける操作である。代入に際して型変換を行うときには、型変換も一つのアクターとして自動挿入される。また、連結はin側の変数とout側の変数とを結ぶことでストリームを生成し、その両端

にそれぞれの変数名をつける操作である。当然、ストリームの両端は同じ型を持っていなければならない。連結は、ストリーム連結子 \leftarrow で表され、out から in へ向かって記述される。

```
例)  int in x=3;
      any in i  $\leftarrow$  any out o;
      struct str {
        double in a;
        any    in b;
      } S;
```

3.3 アクター

NET/Cのアクターは、C言語における関数呼び出しと同様の構文で記述される。アクターは生成されると同時に実行を開始する。このあいだの処理は同期して行われる。すなわち、新しくアクターを生成した側は生成された側が明示的に実行権を放棄するまで待たされる。アクターが実行権を放棄するのは、1) それが終了したとき、2) return を実行したとき、3) reply を実行したときのいずれかである。アクターは終了と同時に消滅する。アクターは return を実行するとそのパラメータを生成者に返し、終了する。reply はパラメータを生成者に返すが、実行を継続する。したがって、reply はプロセスの fork を意味している。

3.4 プリミティブアクター

プリミティブアクターとは、それ以上細分化する事の出来ないアクターのことである。すべてのアクターは、プリミティブアクターとストリームによって実現されている。プリミティブアクターには主に、四則演算の様にデータを変化させる物と、データの数や流れを制御する物とに大別できる。

3.5 複製子の自動挿入

NET/Cでは、C言語との互換性を重視するため、同名の in 型変数が参照されたときには複製子(duplicator)の自動挿入を行う。複製子はストリームの内容を複製して別のストリームへ流すアクターである。

```
例)  int in square(int in x)
      {
        return x*x;
      }
```

もし、複製子の挿入が行われなかったら、例のような場合、NET/Cでは入力側のストリームからデータを二つ消費して、それらの積を出力する。すなわち、この時返される値は、 $x_n * x_{n+1}$ となる。

4 プログラム例

下に、NET/Cによって記述されたアクターの例を示す。

例1: バッファ

```
any in buffer(any in s_in)
{
  in i  $\leftarrow$  out o; //ストリーム生成
  reply i;          //reply 実行
  for(;;)
    pipe(o,s_in);
}
```

例2: tail -n line

```
void tail_minus(any in n, any in line)
{
  in len = length(line);
  return tail_plus(len-n,FIFO(line));
  // lineからのデータをファイルに落とし
  // その内の最後のn行が返される。
}
```

5 まとめ

NET/Cによって形成されるネットワークモデルは、格子結合されたセル上で行われるSOFTの処理と同様の構造を形成することが出来、これによりNET/Cは、効率よくSOFTの処理を実行することの出来る並行プログラミング言語となった。

また、このネットワークモデルは、データフローモデルの実用的な拡張であり、さらに広い分野への応用が可能である。

今後は、さらにオブジェクト指向の概念を導入し、より複雑かつ実用的なプログラミング言語への発展を目指す予定である。

参考文献

- [1] 玉木淳一 森秀樹 上原稔 "ストリーム計算のためのフォールトトレラントプロセッサ接続方式" 電子情報通信学会技術研究報告 1994年8月
- [2] Minoru Uehara. NET/C: Toward the Fine Grained UNIX-like OS, OOIS'94, December 1994