

6T-8

並列プロセスFORTRAN言語と処理系

岩澤 京子 田中 義一 梅谷 征雄  
 日立製作所 中央研究所

1. はじめに

並列計算機 H 2 P<sup>[1]</sup> に対するユーザインタフェースとして、現在、「DEQSOL」<sup>[2]</sup>、「FORTRAN+指示文」<sup>[3]</sup>、「並列プロセスFORTRAN」を用意する。

並列プロセスFORTRAN は、FORTRAN77 の文法に準拠する範囲で、並列処理についてアーキテクチャが用意する全ての機能を指示することを可能にしたものである。これはFORTRAN やDEQSOLトランスレータの出力言語でもあるが、並列計算機のハードウェア<sup>[4]</sup>の性能を十分に引き出すため、ユーザは直接この並列プロセスFORTRAN で記述することができる。

2. 並列指示機能

並列指示機能の一覧を表1に示す。

表1 並列化指示一覧

|      | 機能         | HOST実行                            | PE実行                       |                           |
|------|------------|-----------------------------------|----------------------------|---------------------------|
|      |            | 主記憶                               | ローカル記憶                     |                           |
| プロセス | 分割         | メインプログラム                          | サブルーチン                     |                           |
|      | 割り付け・削除    | CALL ¥ LOAD<br>CALL ¥ DELETE      |                            |                           |
|      | 起動・終結      | CALL ¥ START                      | CALL ¥ END                 |                           |
|      | 同期         | CALL ¥ WALL                       | CALL ¥ POST<br>CALL ¥ WAIT |                           |
|      | プログラムPE間転送 |                                   |                            |                           |
|      | H-PE       | CALL ¥ LOAD                       |                            |                           |
| データ  | 分割         | メインプログラムで宣言<br>サブルーチンの仮引数に*HOST宣言 | サブルーチンで宣言                  |                           |
|      | 割り付け       | コンパイラが静的に割り付ける。                   | コンパイラが静的に割り付ける。            |                           |
|      | 転送         | PE間                               |                            | CALL ¥ SEND<br>CALL ¥ RCV |
|      |            | H-PE                              |                            | サブルーチンの仮引数からローカル変数への代入文   |
|      | メモリの専有     | 特に専有しない                           | 常に各PEが専有                   |                           |

プロセッサごとにプログラムを分割し、これに合わせてデータも分割する。メインプログラムはHOST計算機が実行し、ここで宣言されたデータはHOST計算機の主記憶に置く。並列に実行させる要素プロセッサで動かすプログラムはサブルーチンの形にまとめ、ここで宣言されたデータは、各要素プロセッサのローカル記憶に置く。

並列に実行するために必要な処理、同期制御、要素プロセッサ間通信などは全て手続き呼び出しの形でソースプログラム中に埋め込む。

HOST計算機が要素プロセッサに起動をかける時に、実引数のパラメタリストを要素プロセッサへ渡すことができる。要素プロセッサ側は、通常のFORTRAN リンケージ規約に従い、このパラメタリストを用いて、主記憶をアクセスすることができる。要素プロセッサで実行するサブルーチンの仮引数が主記憶にあるかローカル記憶にあるかを区別するために、「\*HOST」の宣言が必要になる。「\*HOST」で宣言された仮引数が代入左辺に現われれば主記憶の更新であり、代入左辺以外であれば主記憶の参照である。

単純な記述例を次の図1に示す。

オリジナルプログラム

```
REAL X(1000),Y(1000)
DO 10 K=2,1000
10 X(K)=X(K-1)+Y(K)
```

```

HOST計算機用メインプログラム
DO 10 N=1,NPE
10 CALL ¥LOAD(N,'LOOP10')
DO 20 N=1,NPE
20 CALL ¥START(N,'LOOP10',N)
CALL ¥WALL

ある要素プロセッサ用サブルーチン
SUBROUTINE LOOP10(N)
*HOST N
REAL X(1000/NPE+1),Y(1000/NPE+1)
DO 10 I=N+1,1000,NPE
CALL ¥RCV(I-1,I-1,W)
X(I/NPE+1)=W+Y(I/NPE+1)
CALL ¥SEND(I+1,I,X(I/NPE+1))
10 CONTINUE
RETURN
```

図1 並列プロセスFORTRAN記述例

3. 処理の流れの概要

3.1 ロードモジュールの作成

(HOST計算機の処理)

手順1: HOST計算機用のメインプログラムと、要素プロセッサで並列に実行すべきサブルーチンを並列プロセスFORTRAN で記述する。

手順2: 並列プロセスFORTRAN コンパイラでコンパイルし、オブジェクトを生成する。

手順3: 各々HOST計算機用ライブラリ、要素プロセッサ用ライブラリをリンクして、HOST計算機用、要素プロセッサ用ロードモジュールを作る。

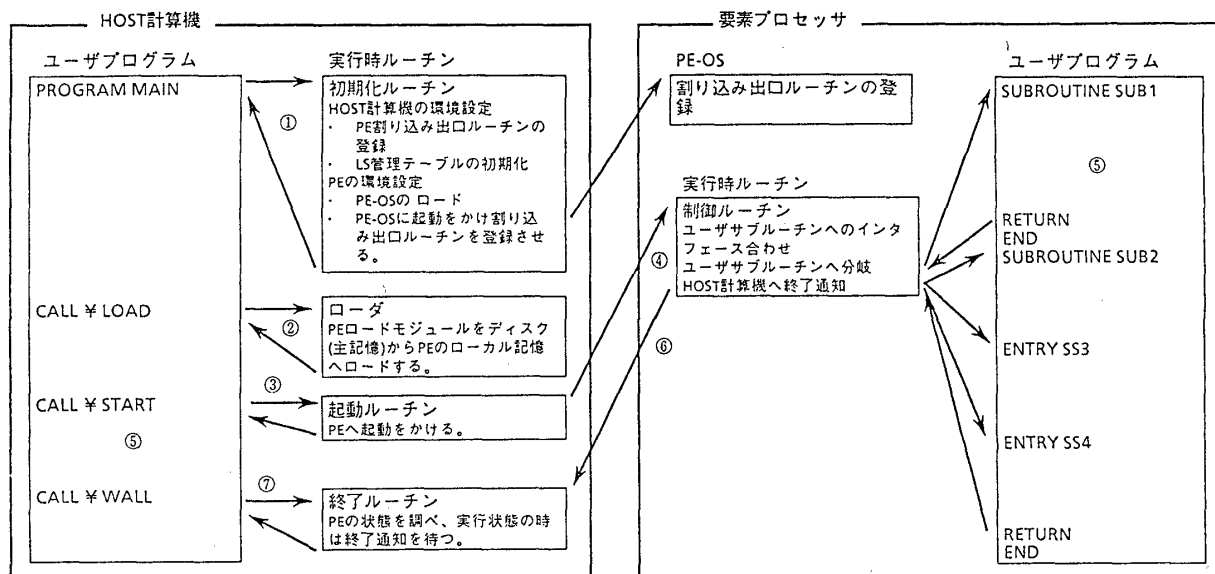


図2 処理の流れの概要

### 3. 2 ユーザプログラムの実行

処理の流れの概要を図2に示す。

- ①初期化ルーチン呼び出し、HOST計算機と要素プロセッサのFORTRAN 実行環境を設定する。
- ②HOST計算機にある要素プロセッサ用サブルーチンを各要素プロセッサのローカル記憶にロードする。各要素プロセッサには、メモリ管理のOSを置かず、ローカル記憶の管理はHOST計算機が行なう。そのために必要な情報は、ローディング時に採取し主記憶にまとめて確保する。
- ③プロセッサ番号、入り口名称、要素プロセッサに渡す主記憶データを引数として、要素プロセッサに起動をかける。
- ④要素プロセッサの制御ルーチンが走る。
- ⑤HOST計算機、要素プロセッサともユーザプログラムを実行する。
- ⑥要素プロセッサは計算の終了をHOST計算機に通知する。
- ⑦HOST計算機は要素プロセッサからの計算終了通知を待つ。

### 4. コンパイル処理の特徴

従来のシングルプロセッサ向けのコンパイルと比べて、特徴となるのは次の3つである。

#### 4. 1 並列処理に必要な手続きのインライン展開

並列実行時の同期制御やプロセッサ間のデータ通信のために2章に記した手続きを用意する。原則としてこれらは、コンパイラがインラインに展開して最適化を施すことにより、呼び出しオーバーヘッドが実行時には現われないようにする。

#### 参考文献

- [1] 濱中 勉: 並列計算機「H2P」のシステム構成 (本大会予稿集)
- [2] 佐川 勉: 並列計算機向きDEQSOLの基本検討 (本大会予稿集)
- [3] 斎藤 勉: 分散共有メモリのためのFORTRAN インタフェース (本大会予稿集)
- [4] 中越 勉: 並列計算機「H2P」のプロセッサ間非同期データ転送 (本大会予稿集)

#### 4. 2 データ送受信処理のベクトル化

各要素プロセッサにはベクトル処理機構があり、これを利用するためにコンパイラは自動ベクトル化変換を行なう。

ベクトル化対象ループ中に、他の要素プロセッサとのデータ送受信がある場合、可能であればこれをベクトル送受信に変換する。これにより、データ送受信の実行効率が上がり、また、他のベクトル演算のなかに送受信処理があるためにループ分割を施すようなオーバーヘッドが無くなる。送信、受信のいずれかのみをベクトル化することも可能である。

#### 4. 3 要素プロセッサがアクセスする

##### 主記憶データの配列化

HOST計算機が「¥START」を呼び出してパラメータを渡して要素プロセッサを起動する時刻と、要素プロセッサがこのパラメータをアクセスする時刻にはかなりのずれがある。一般にはこの間に、HOST計算機(や他の要素プロセッサ)が主記憶の内容を書き換える可能性が高い。

従って、要素プロセッサ起動時のパラメータの値を保証するために、プロセッサ台数分作業領域を用意し、起動時の値を保存する。

### 5. おわりに

並列計算機H2Pに対するユーザインタフェースの最もハードウェアに近いレベルとして、並列プロセスFORTRANの機能仕様をまとめ、このソースプログラムの処理方式について検討し、実現の見通しを得た。