

VPP500における共有メモリマシン向け
アプリケーションプログラムの並列化方法

5T-1

折居茂夫

富士通株式会社スーパーコンピュータシステム部

1. はじめに

共有メモリマシン向けに書かれたアプリケーションプログラムを、分散メモリマシン向けに並列プログラミングする（以後、並列化と呼ぶ）方法について述べる。

分散メモリマシン上で並列化を行うためには、データ分割、手続き分割、データ転送の3項目を考慮する必要がある〔1, 2〕。これらの項目は「データのUNIFY」という概念を導入して2つのフェーズに分けることができる。2つのフェーズに分けてプログラミングすることにより、共有メモリマシン向けに書かれた既存大規模アプリケーションプログラムの並列化を容易に実現することができる。

2. 並列化の問題点

共有メモリ計算機特有のプログラミングテクニックとして、配列データを始めに宣言し、これから任意の数、次元、大きさの配列切りだして利用することが多い（図-1上）。このように定義された配列を意図的に各PE(Processing Element)上に配置（データ分割）するのは、一般に困難である。一方、並列計算機向けに新規に開発されたアプリは、プログラム作成の最初にデータ分割設計を行うため、この問題を解決することができる。既存アプリのデータ分割は、通常、データ分割用変数を新しく定義し、今まで使用していた変数と交換するようにプログラムを変更する。図-1は既存アプリにおいてデータ分割用新変数を定義する方法を示す。

A Parallel Programming Method for VPP500
in the Case of Application Programs
written for Shared-Memory Machines
Shigeo ORII
Fujitsu Limited, Supercomputer Systems Dept.
1-9-3, Nakase, Mihama-ku, Chiba 261, Japan

既存アプリのプログラミング技法

```
dimension a(10000000)
m=1000
n=10000
call int(a(1), a(m+1), m, n)
. . .
subroutine int(b, c, m, n)
dimension b(m), c(m, n)
. . .
```

データ分割のため配列を別々に宣言

```
parameter(m=1000, n=10000)
common/para/ a(m), b(n)
call int
. . .
subroutine int
parameter(m=1000, n=10000)
common/para/ a(m), b(n)
. . .
```

図-1 データ分割のための既存アプリの変更

並列化ではこのように作成した配列 a, b をデータ分割する。この問題点は、データ分割用変数を導入した場合、その変数を使用する全てのサブルーチンが書き換えの対象となることである。既存の大規模アプリにおいてこれは困難な作業である。

3. 「2フェーズ並列化法」

既存アプリの分散メモリマシン上での並列化の問題は、データ分割をしなければ回避できる。これに着目して並列化を次の2フェーズに分けるプログラミング方法（今後、2フェーズ並列化法と呼ぶ）を提案する。

3.1 フェーズ1 (手続き分割、データ転送)

並列向け言語拡張 VPP FORTRAN77 EX/VPPでのプログラミング例を図-2に示し説明する。この言語は!XOCL 行で並列化の指示を行う。また実行イメージを図-3に示す。

配列b を整合配列とする。do 2で並列計算する場合。b の他のPEで計算された結果のみが次の計算に必要なだと仮定する。このため、b の計算結果のデータ転送が必要となる。

次にプログラミング手順を示す。

1. 各PEに同じプログラムとデータをコピーし起動する最適化制御行(!XOCL PARALLEL REGION)を挿入する。
2. 手続き分割の指示を出す最適化制御行(!XOCL SPREAD DO, !XOCL END SPREAD)を挿入する。
3. 計算終了後、各PEで計算したデータを他の全てのPEに転送する。このデータ転送をデータのUNIFY と呼ぶ。

```

                                !XOCL PARALLEL REGION
                                !XOCL SPREAD DO
do 2 j=1, jmax                do 2 j=1, jmax
do 1 i=1, imax                do 1 i=1, imax
    =a(i, j)                    =a(i, j)
    . . .                       . . .
    b(i, j)=                    b(i, j)=
1 CONTINUE                    1 CONTINUE
2 CONTINUE                    2 CONTINUE
                                !XOCL END SPREAD
                                (b のUNIFY)
                                !XOCL END PARALLEL
    
```

図-2 VPP FORTRAN77 EX/VPPによるフェーズ1のプログラミング例

UNIFY は、VPP FORTRAN77 EX/VPPでは手続き分割のSPREAD DO 文、データ転送のSPREAD MOVE 文の2つの最適化制御行で作成することができる〔3〕。この方法は、既存のアプリにUNIFY という手続きとそのための宣言文を付け加えるだけで並列化でき、UNIFY によって計算直後に各PEに計算結果がデータ転送されるため、並列化を局所的に実施することができる。これにより、データ構造の全体が把握できない大規模既存アプリの並列化が可能となる。

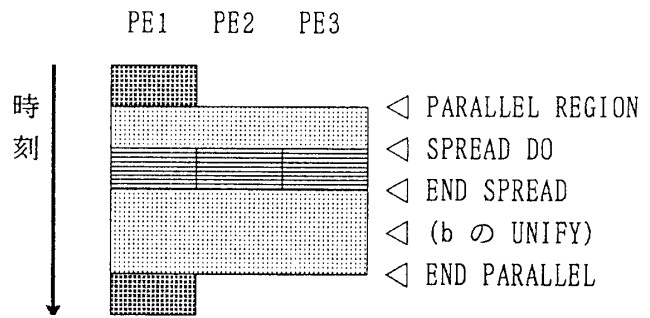


図-3 3PE によるフェーズ1の実行イメージ

3.2 フェーズ2 (データ分割)

PEのメモリがアプリの実行に不十分な場合、データ分割を実施する〔2〕。この場合は、2の問題点で示したように、データ分割をする変数を使用する全てのサブルーチンを書換える。

4. 議論

「2フェーズ並列化法」で既存アプリを並列化するために並列計算機が持つべき条件は2つある。第一は各PEのメモリが既存アプリを実行できるくらい大きいこと。第二は全てのPEにプログラムとデータをコピーし起動する機能(PARALLEL REGION)と必要な部分のみ並列実行する機能(SPREAD DO)を持つことである。

UNIFY の多用はデータ転送時間を増加するため、実行性能向上のためにはデータの参照関係の調査範囲を広げ、不必要なUNIFY を削減する必要がある。

フェーズ2のデータ分割は全ての配列に必要なではない。一般に、フェーズ1でUNIFY をした配列に必要なとなる。また各PEに十分なメモリ容量がある場合、データ分割する必要はない。

〔文献〕

- 〔1〕 吉田, 中村, 福田, 岡田, 中村: NWT 向け並列Fortran プログラミング について, 航空宇宙技術研究所特別資料SP-19, pp. 247-252(1992)
- 〔2〕 折居: 分散メモリ MIMDコンピュータにおけるデバッグ環境に必要な条件について, 情報処理学会第47回全国大会2D-4, pp. 5-149~5-150(1993)
- 〔3〕 VPP500プログラミングハンドブック, 近刊