5D−2

超並列計算機用言語 Bee-Fortran における配列配置方式の検討

小前 晋† 杉森 英夫† 大谷 浩司‡ 渦原 茂§ 安村 通晃§

†住友金属工業(株) ‡(有)AXE §慶應義塾大学

1 はじめに

現在，我々は超並列計算機 SM-1[1] 用言語 Bee-Fortran[2] を開発している．SM-1 などの 2 次元メッシュのネットワークを備えた SIMD 分散メモリ型超並列計算機では，配列の配置が実行性能に大きく影響する．Bee-Fortran は HPF (High Performance Fortran)[3] に準拠しており，ALIGN, DISTRIBUTE などのコンパイラ制御命令によって，ユーザは陽に配列の配置を指定できる．しかし，熟練したユーザでさえも，最適な配列配置を選択することは難しい．そこで，コンパイラによって最適な配列配置を自動的に選択する方式が望まれる．配列配置方式を決めるには，どの次元をプロセッサ・アレイ上に配置し，配置された次元をどう分割するかを決めなければならない．これらについて，プログラム中の配列へのアクセス・パターンやマシンの演算と通信の性能比を考慮し，検討したことを述べる．

2 プロセッサ・アレイへの配置

配列演算で，各配列の対応する次元の配置が異なる場合，演算前に配列要素を移動しなければならない．この移動は PE 間の通信となり，一般的にコストが高いので，配列同士を align する必要がある．配列同士の align には CAG(Component Affinity Graph)[4] を用いる．CAG は，配列の次元をノードとし，それぞれの次元間で align すべきもの同士をエッジによってつなぎ，そのエッジに weight を与えたものである．今回は，この weight には並列に扱われる回数とそのサイズを用いる．次に，配列をできるだけ並列に扱うために，weight の値の大きいものからプロセッサ・アレイの次元に割り当て，矛盾の起こる場合は，weight の値の大きいほうを優先する．また，サイズが PE 数よりも極端に少ない次元や，並列にアクセスされない次元は PE 上に並べる．

また，以下のような，特別な配列参照の場合に有効な配置方法も検討した．2 次元のプロセッサ・アレイにおいて，以下のようなコード中の配列 A の 1, 2 次元目をプロセッサ・アレイに配置させた場合，S2 と S3 は配列の再配置を行なわない限り，プロセッサ・アレイ全てを使うようにはならない．

```
A(:,:,1) = A(:,:,1) + ....    S1
A(:,1,:) = A(:,1,:) + ....    S2
A(1,:,:) = A(1,:,:) + ....    S3
```

ここで，図 1 のように 1, 2 次元方向に一つずつずらして配置すると，三つの次元のうち任意の二つの次元に対して，並列に扱うことができる．これを一般化すると，n 次元メッシュ型プロセッサ・アレイへ n+1 次元配列を対応させる場合，n 次元方向の要素を一つずつずらして対応させることによって，任意の n 次元に関して並列に扱えることになる．

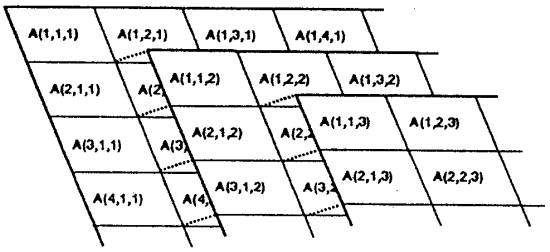

図 1: 3 次元配列の割当

3 各次元の分割方法

次に，プロセッサ・アレイに配置した次元をどのように分割するかを決定しなければならない．次元ごとの分割の方式には図 2 に示すようにサイクリック分割，ブロック分割，サイクリック・ブロック分割などがある．サイクリック分割，ブロック分割の利点は以下の点である．

The study of array distribution scheme in Bee-Fortran language for massively parallel machines

Susumu KOMAE†, Hideo SUGIMORI†, Koji OTANI‡, Shigeru UZUHARA§, Michiaki YASUMURA§

†SUMITOMO Metal Industries, LTD.

‡AXE Inc.

§KEIO University

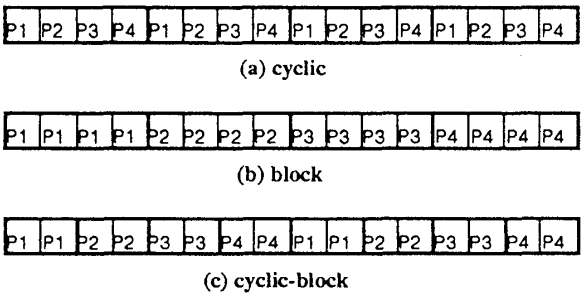

図 2: 分割方式

- サイクリック分割
  配列要素の一部の領域だけの演算が多い場合や，もしくは演算の領域が徐々に狭まっていく場合などに，負荷が分散される．ブロック分割では，演算しなければならない配列要素が一部の PE に集中してしまうが，サイクリックに分割することによって分散される．

- ブロック分割
  近傍の要素との計算が多い場合は，通信量が少なく済む．サイクリック分割では，近傍の要素は他の (隣接) PE にあるので，各要素ごとに通信が必要であるが，ブロック分割では，担当する領域の要素のうち，境界にある要素のみを通信するだけでよい．

サイクリック・ブロック分割は，ブロックサイズが 1 の場合にサイクリック分割となり，$\lceil$(全体の要素数)$/(PE$数$)\rceil$ の場合にブロック分割となるので，以下では，サイクリック・ブロック分割について述べる．対象とする配列のプログラム中での演算領域の偏りと近傍計算との割合によって，上記のサイクリック分割，ブロック分割の利点を合わせ持つようなブロックサイズを選べばよい．演算領域の偏りと近傍計算とは，各演算の回数と PE 間通信の回数によって明確にできる．これらをパラメータとして，最適な分割方法を求めることができる．

```
integer, parameter :: m=16384
integer, dimension(m) :: X
integer i

do i = 1,m
  X(1:i) = X(1:i) + i
enddo
do i = 1,m
  X(1:m-1) = X(1:m-1) + X(2:m)
enddo
end
```

上の例で，サイクリックに折り畳まれた回数を $fold$，一つのブロックのサイズを $chunk$ とし，隣接 PE 間のデータ移動 (通信)，PE 内部でのデータ移動 (代入)，足し算の一回あたりの時間を，それぞれ T_{NEWS}，T_{MOVE}，T_{PLUS} とすると，PE ごとの主な演算に必要な時間は以下のようになる．

- 隣接 PE 間のデータ移動 (通信):
  $(fold \times m) \times T_{NEWS}$

- PE 内部でのデータ移動 (代入):
  $(\sum_{i=1}^{m} \lceil \frac{fold}{m} \times i \rceil \times chunk + fold \times (chunk - 1) \times m + fold \times chunk \times m) \times T_{MOVE}$

- 演算:
  $(\sum_{i=1}^{m} \lceil \frac{fold}{m} \times i \rceil \times chunk + fold \times chunk \times m) \times T_{PLUS}$

SM-1 の場合，$T_{NEWS} = 64, T_{MOVE} = 45, T_{PLUS} = 23$ で，PE 数が 1024 個である．全 PE を有効に使うには各 PE が $16384/1024 = 16$ 個の配列要素を担当する必要があるので，$fold \times chunk = 16$ となる．これらより，この場合の最適な分割方法は $fold = 4, chunk = 4$ となることが求められる．

4 おわりに

分散メモリ型超並列計算機用言語 Bee-Fortran の配列配置方式について検討した．その結果，プログラム中のアクセス・パターンによって，最適な配置を選択する方法の一つが明らかになった．今回は，2 次元メッシュのネットワークを備えた SM-1 について検討したが，他の同様なアーキテクチャの超並列計算機にも適用できると思われる．

参考文献

[1] 松田, 湯淺: SIMD 型超並列計算機 SM-1(仮称) の概要とその性能, 情報処理学会研究報告, 92-ARC-87, *August, 1992*

[2] 大谷, 小前, 杉森, 渦原, 安村: 超並列計算機用 Fortran コンパイラの設計と試作, 電子情報通信学会研究報告, COMP92-94, *May 11, 1993*

[3] High Performance Fortran Language Specification Version 1.0, High Performance Fortran Forum, *May 3, 1991*

[4] J. Li, M. Chen: The Data Alignment Phase in Compiling Programs for Distributed-Memory Machines, Journal of Parallal and Distributed Computing, Vol.13, pp.213-221, *1991*