



HPF入門編2

その2

2009年7月

NEC 第一コンピュータソフトウェア事業部

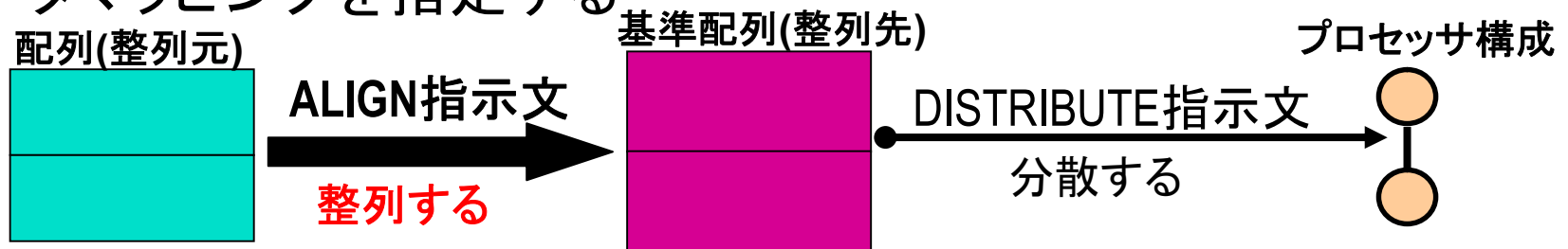
林 康晴

HPFの基本機能2

配列同士の相対的位置関係(整列)により 間接的にマッピングを指定する	ALIGN 指示文
メモリを消費しない仮想配列を宣言する	TEMPLATE 指示文

ALIGN指示文(1)

- 基準配列に対する相対的な位置関係(整列)により、間接的にデータマッピングを指定する



書式: 配列 a_n (整列元)を、基準配列 t (整列先)の対応する要素と同じ抽象プロセッサへマップする

- 1つの配列の整列

ALIGN a_1 ($\begin{array}{|c|} \hline \text{変数} \\ \hline * \end{array}$,...) WITH t ($\begin{array}{|c|} \hline \text{整数式} \\ \hline * \end{array}$,...)

- 複数の配列の整列

ALIGN ($\begin{array}{|c|} \hline \text{変数} \\ \hline * \end{array}$,...) WITH t ($\begin{array}{|c|} \hline \text{整数式} \\ \hline * \end{array}$,...) :: a_1, a_2, \dots

- ◆ 4通りの整列方法が指定可能

- 3つ組、縮退、複製、Single整列

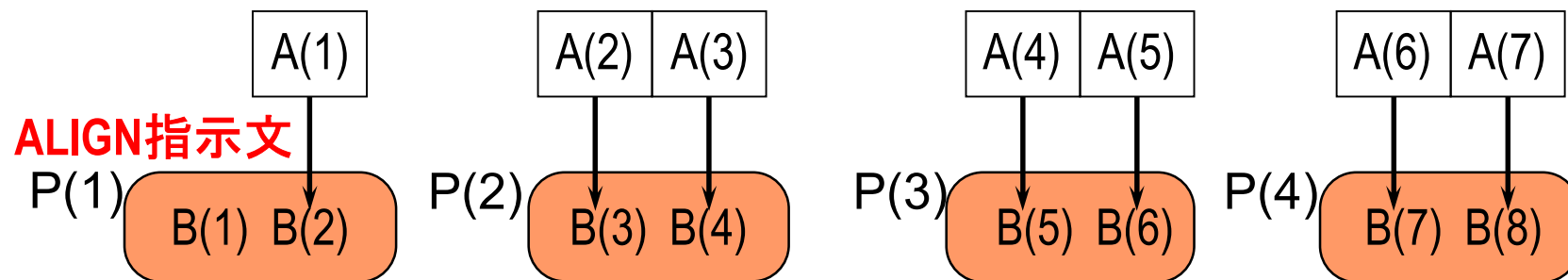
ALIGN指示文(2) 3つ組

- 整列元と整列先の関連する次元の添字を一次式で対応させる

用途: 関連する配列要素の添字にずれがある場合など

例:

```
REAL A(7), B(8)
!HPF$ PROCESSORS P(4)
!HPF$ DISTRIBUTE B(BLOCK) ONTO P
!HPF$ ALIGN A(I) WITH B(I+1)  !A(I)をB(I+1)と同じプロセッサへマップする
DO I=1,7
  A(I)=B(I+1)
ENDDO
```

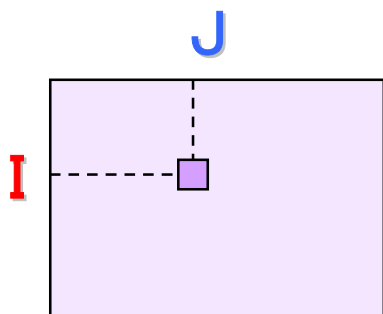


ALIGN指示文(3) 3つ組(多次元の場合)

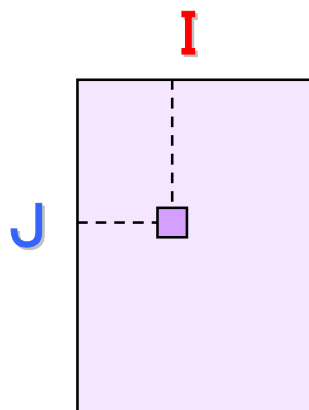
- 同じ変数(*align-dummy*)が記述された次元同士が対応する

例: `ALIGN A(I,J) WITH B(J,I)`

A(m,n)



B(n,m)



適用例:

```
DO I=1,m  
  DO J=1,n  
    A(I,J)=B(J,I)  
  END DO  
END DO
```

ALIGN指示文(4) 縮退

- マップしない次元: 整列元の縮退
- ‘*’ が指定された整列元の次元は、分散されない(整列の相対的位置に影響しない)

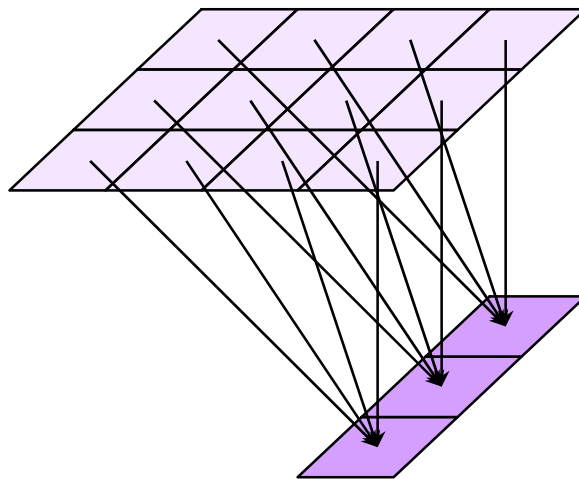
用途:

- マップしない次元に指定する
- 整列先の方が次元数が小さい場合、整列元の余った次元に指定する

例: **ALIGN A(I,*) WITH B(I)**

A(3,4)

B(3)



適用例:

```
DO I=1,3  
  DO J=1,4  
    B(I)=B(I)+A(I,J)  
  END DO  
END DO
```

ALIGN指示文(5) 複製

- 複数のプロセッサに配置したい場合: 整列先への複製
- ‘*’ が指定された整列先の次元に沿って、整列元を複製する

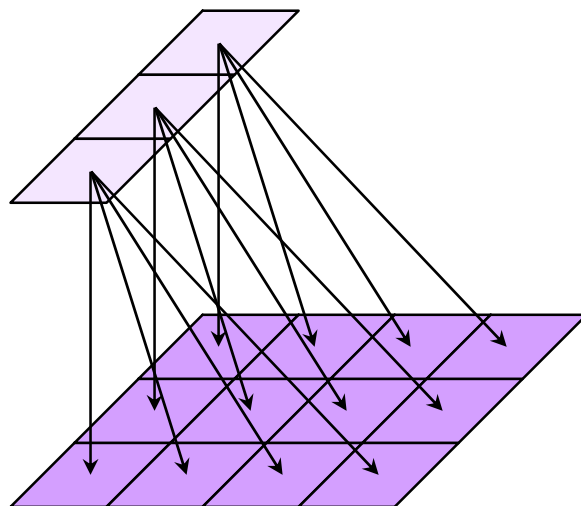
用途:

- 整列元が、整列先の複数の要素に対応する場合
- 整列先の方が次元数大きい場合、整列先の余った次元に指定する

例: `ALIGN A(I) WITH B(I,*)`

A(3)

B(3,4)



適用例:

```
DO J=1,4  
  DO I=1,3  
    B(I,J)=A(I)  
  END DO  
END DO
```

ALIGN指示文を使った方がよい場合(1)

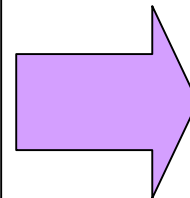
■宣言範囲が異なると、DISTRIBUTE指示文だけでは、各プロセッサに配置される複数の配列の割付け範囲にずれが発生する。

例: 関連する配列の宣言範囲が異なる場合

```
real a(11),b(0:11)
!hpf$ processors p(2)
!hpf$ distribute (block) onto p :: a,b

do i = 1,11
  a(i) = b(i)  ! 通信発生
```

	P(1)					P(2)						
A	1	2	3	4	5	6	7	8	9	10	11	
B	0	1	2	3	4	5	6	7	8	9	10	11

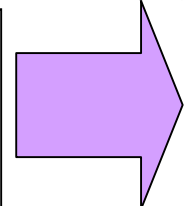


```
real a(11),b(0:11)
!hpf$ processors p(2)
!hpf$ distribute (block) onto p :: b
!hpf$ align a(i) with b(i)
do i = 1,11
  a(i) = b(i)  ! 通信不要
```

	P(1)					P(2)						
A	1	2	3	4	5	6	7	8	9	10	11	
B	0	1	2	3	4	5	6	7	8	9	10	11

◆関連する配列の宣言範囲が見かけ上異なる場合も同様

```
subroutine sub(n,m)
  real a(n),b(m)
!hpf$ distribute (block) :: a,b
```



```
subroutine sub(n,m)
  real a(n),b(m)
!hpf$ distribute (block) :: b
!hpf$ align a(i) with b(i)
```

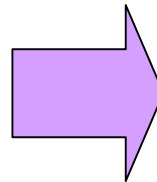

ALIGN指示文を使った方がよい場合(2)

■上下限値が翻訳時に不明の場合、DISTRIBUTE指示文のみでは、複数の配列の分散ブロック幅が一致するかどうか翻訳時には不明

例: 割付け配列の場合

```
real,allocatable :: a(:),b(:),c(:)
!hpf$ distribute (block) :: a,b,c

do i = 1,n
  a(i) = b(i) + c(i) !通信発生
```



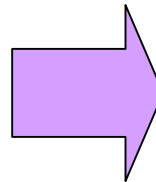
```
real,allocatable :: a(:),b(:),c(:)
!hpf$ distribute (block) :: c
!hpf$ align (i) with c(i) :: a,b
do i = 1,n
  a(i) = b(i) + c(i) !通信不要
```

•例えば、a(1:10),b(1:50),c(1:100)のように、割付け範囲がばらばらになる可能性があるため、DISTRIBUTE指示文だけでマップすると、効率の悪い実行コードが生成される。

例: 形状引継ぎ配列の場合

```
subroutine sub(a,b,c)
  real a(:),b(:),c(:)
!hpf$ distribute (block) :: a,b,c

do i = 1,n
  a(i) = b(i) + c(i) !通信発生
```



```
subroutine sub(a,b,c)
  real a(:),b(:),c(:)
!hpf$ distribute (block) :: c
!hpf$ align (i) with c(i) :: a,b
do i = 1,n
  a(i) = b(i) + c(i) !通信不要
```

HPFの基本機能2

配列同士の相対的位置関係(整列)により 間接的にマッピングを指定する	ALIGN指示文
メモリを消費しない仮想配列を宣言する	TEMPLATE指示文

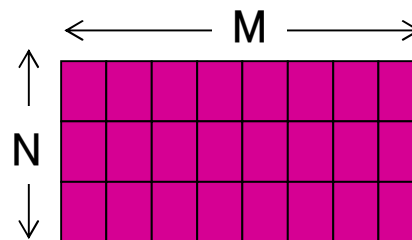
TEMPLATE指示文

- メモリを消費しない仮想配列(テンプレート)を宣言する

```
!HPF$ TEMPLATE T(N)
```



```
!HPF$ TEMPLATE T(N,M)
```



書式: テンプレート T_n を宣言する(配列の宣言と同様)

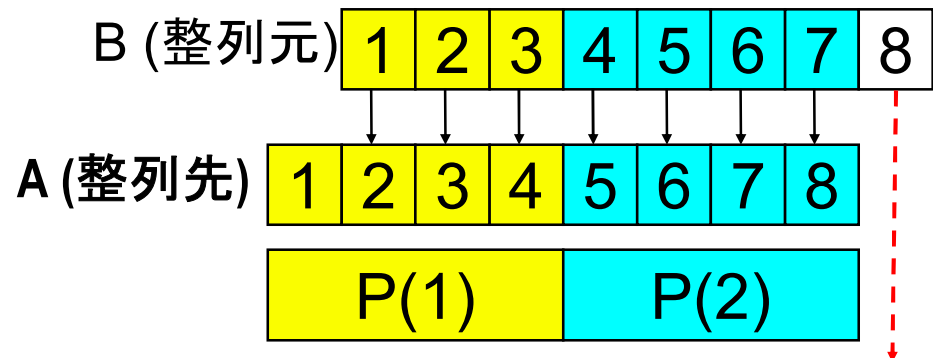
```
TEMPLATE  $T1$ (上下限,...),  $T2$ (上下限,...), ...
```

- 用途:
- テンプレートはHPF指示文中のみで使用可能(記憶領域を持たないため)
 - ALIGN指示文の基準配列(整列先)として適当な配列がない場合に、テンプレートを整列先にして、そのテンプレートをDISTRIBUTE指示文により分散する。
 - ON指示構文中に指定する適当な配列がない場合に、テンプレートが配置されている抽象プロセッサ上での実行を指示する。

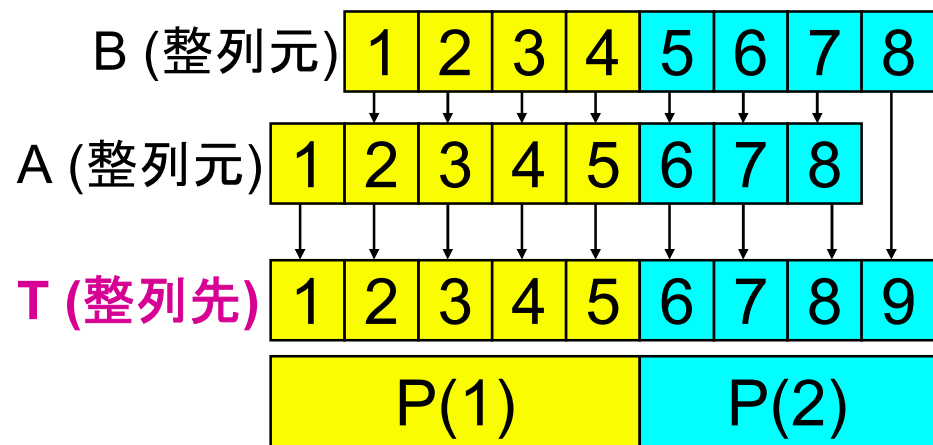
TEMPLATE指示文: 整列先としての利用例

■ALIGN指示文では、整列元が整列先から「はみだす」様な指定は禁止されているため、全配列の宣言範囲をカバーする**テンプレート**を宣言し、整列先とする

```
INTEGER A(8),B(8)
!HPF$ PROCESSORS P(2)
!HPF$ ALIGN B(I) WITH A(I+1) !エラー
!HPF$ DISTRIBUTE A(BLOCK) ONTO P
DO I=1,7
  B(I) = A(I+1)
```



```
INTEGER A(8),B(8)
!HPF$ PROCESSORS P(2)
!HPF$ TEMPLATE T(9)
!HPF$ ALIGN B(I) WITH T(I+1)
!HPF$ ALIGN A(I) WITH T(I)
!HPF$ DISTRIBUTE T(BLOCK) ONTO P
DO I=1,7
  B(I) = A(I+1)
```



HPFのデータマッピングまとめ

1. どのループを並列化するかを決定する
2. 並列化するループによりアクセスされる次元を分散する

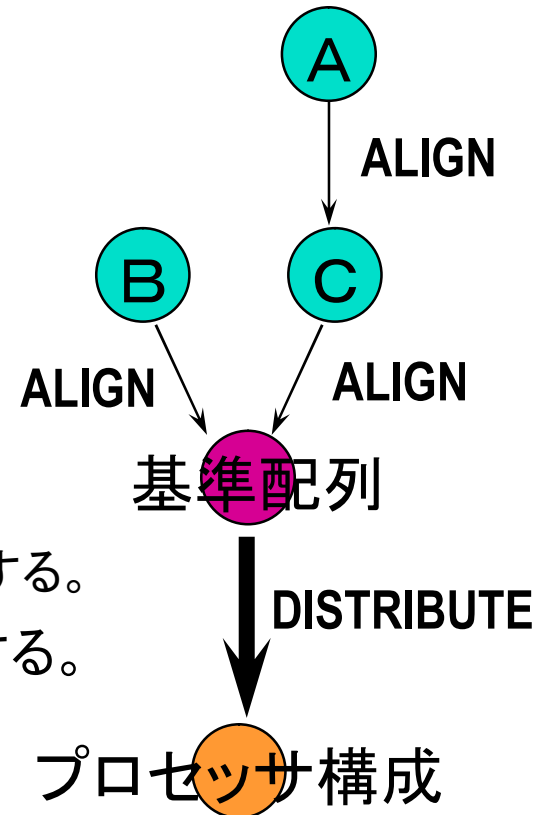
```
DO I=1,N  ! 並列化
DO J=1,N
  A(J,I) = B(J,I+1)
```

Aの2次元目を分散

- ◆ 関連の深いデータは整列(ALIGN)する。
 - 参照パターンにより芋づる式に決定可能
- 例: Aの2次元目 "I"と、Bの2次元目の "I+1"

```
!HPF$ ALIGN B(J,I) WITH A(J,I-1)
```

- 必要に応じてテンプレートを基準配列(整列先)とする。
- ◆ 基準配列(テンプレート等)を分散(DISTRIBUTE)する。
 - 他の配列は整列関係を守ってマップされる。



今回 説明したHPF指示文一覧

プロセッサ構成を宣言する	PROCESSORS指示文	データマッピング
配列を分散する	DISTRIBUTE指示文	
配列同士の相対的位置関係(整列)により間接的にマッピングを指定する	ALIGN指示文	
メモリを消費しない仮想配列を宣言する	TEMPLATE指示文	
並列化可能であることを明示する	INDEPENDENT指示文 [オプション] ◆作業変数指定: NEW節 ◆集計変数指定: REDUCTION節	並列性明示
実行プロセッサを指定する	ON指示構文 [オプション] ◆通信不要を明示: LOCAL節	実行制御
袖(シャドウ)領域の割付けを宣言する	SHADOW指示文	
シフト通信の実行を指示する	REFLECT指示文	