

第12章 HPF 組込み手続とライブラリ手続の 公認拡張

本章では、HPF Version 2.0 に対する拡張として公認された組込み手続とライブラリ手続に関して述べる。

拡張組込み手続には、Fortran の TRANSPOSE 組込み関数を汎用化した転置関数が含まれる。アルゴリズムによっては、多次元配列を様々な次元軸に沿って参照しなければならないことがある。今日の計算機では通常、配列要素が局所メモリ上で連続するよう、内側ループで配列の 1 次元目をなぞるようにするのがもっともよい。汎用化された転置は、このようなデータの再編成に必要とされるもので、単なるデータの再配置とは異なる。多くの場合、転置の結果は、第 1 次元が *dist-format*(*) で分散された変数に代入される。

次元数、及び入れ換えられる次元軸の集合が既知ならば、この種の操作には FORALL が適している。たとえば以下のようになる。

```
FORALL(I1 = 1:SIZE(ARRAY,1))
  FORALL(I2 = 1:SIZE(ARRAY,2))
    FORALL(I3 = 1:SIZE(ARRAY,3))
      RESULT(I3,I1,I2) = ARRAY(I1,I2,I3)
    ENDFORALL
  ENDFORALL
ENDFORALL
```

しかし、入力の次元軸と結果の次元軸との関係が可変であるときには、FORALL による表現は不自由である。そこで、TRANSPOSE 組込み関数は汎用化され、次元数が 0 でない任意の入力配列 (転置される) と、入力配列の次元数と同じ大きさを持つ整数型 1 次元配列 (次元軸の順列を与える) が引数に指定できるようになった。

引数の既定値により、この関数は既存の Fortran 関数である 1 引数の TRANSPOSE への拡張となる。

新しい 2 つの組込み問合せ関数 ACTIVE_NUM_PROCS と ACTIVE_PROCS_SHAPE は、プログラムを実行する部分プロセッサの大きさと形状を知るために利用できる。それらは ON 構文で変更される。

拡張ライブラリは、マッピング問合せサブルーチン群から成る。HPF_ALIGNMENT と HPF_TEMPLATE の拡張版では、新たに省略可能な出力引数 DYNAMIC を指定できる。これによりプログラムは、実体やその最終整列先が DYNAMIC 属性を持つかどうかを知ることができる。HPF_DISTRIBUTION の改良版と新しい 2 つのマッピング問合せサブルーチンは、GEN_BLOCK 分散形式と INDIRECT 分散形式のマッピングを知るために特に役立つ。

12.1 拡張組込み手続きの仕様

ACTIVE_NUM_PROCS(DIM)

省略可能な引数. DIM

機能. 最内側の ON ブロックで決まる、現在プログラムを実行中のプロセッサの総数、又はプロセッサ配列の指定された次元における、現在プログラムを実行中のプロセッサ数を返す。

分類. プロセッサ問合せ関数

引数.

DIM (省略可能) 整数型スカラであって、 $1 \leq \text{DIM} \leq n$ の範囲の値でなければならない。ここで n は、プロセッサ配列の次元数とする。

結果の型、型パラメタ 及び 形状. 基本整数型スカラとする。

結果の値. 結果の値は、囲まれる最内側の ON ブロックで決まるプロセッサ配列の第 DIM 次元の寸法とする。DIM が省略されたときは、そのようなプロセッサ配列の要素の総数とする。結果は常に 0 より大きいものとする。囲まれる ON ブロックがないとき、結果の値は NUMBER_OF_PROCESSORS() の結果と同じとする。

例示. 以下のプログラム断片

```
INTEGER X(16, 3)
!hpf$ TEMPLATE T(16, 8)
!hpf$ PROCESSORS PROCS(4, 4)
!hpf$ ALIGN X(I, J) WITH T(I, 3*J-1)
!hpf$ DISTRIBUTE T(CYCLIC(2), BLOCK) ONTO PROCS
!hpf$ ON      (PROCS(:, :)) BEGIN
!hpf$ ON HOME(X(2:12:10, :)) BEGIN
PRINT *, ACTIVE_NUM_PROCS()
PRINT *, ACTIVE_NUM_PROCS(DIM=1)
PRINT *, ACTIVE_NUM_PROCS(DIM=2)
!hpf$ END ON
!hpf$ END ON
```

は、プログラムを実行するハードウェアプロセッサ配列の大きさや形状とは無関係に、6、2、3 を印字する。

ACTIVE_PROCS_SHAPE()

機能. 最内側の ON ブロックで決まる、現在の活動プロセッサ配列の形状を返す。

1 分類. プロセッサ問合せ関数

2 引数. なし

3
4 結果の型、型パラメタ 及び 形状. 基本整数型 1 次元配列で、囲まれる最内側の ON ブ
5 ロックで決まるプロセッサ配列の次元数に等しい大きさとする。

6
7 結果の値. 結果の値は、囲まれる最内側の ON ブロックで決まるプロセッサ配列の形状
8 とする。囲まれる ON ブロックがないとき、結果の値は PROCESSORS_SHAPE() の結果と
9 同じとする。

10 例示. 以下のプログラム断片

```
11  
12  
13     INTEGER X(16, 3)  
14     !hpf$ TEMPLATE T(16, 8)  
15     !hpf$ PROCESSORS PROCS(4, 4)  
16     !hpf$ ALIGN X(I, J) WITH T(I, 3*J-1)  
17     !hpf$ DISTRIBUTE T(CYCLIC(2), BLOCK) ONTO PROCS  
18     !hpf$ ON      (PROCS(:, :)) BEGIN  
19         PRINT *, ACTIVE_PROCS_SHAPE()  
20         !hpf$ ON HOME(X(2:12:10, :)) BEGIN  
21             PRINT *, ACTIVE_PROCS_SHAPE()  
22         !hpf$ END ON  
23     !hpf$ END ON  
24  
25     !hpf$ END ON
```

26 は、プログラムを実行するハードウェアプロセッサ配列の大きさや形状とは無関係に、4, 4 と
27 2, 3 を印字する。

30 TRANSPOSE(ARRAY,ORDER)

31 省略可能な引数. ORDER

32
33 機能. 配列の次元軸を入れ換える (汎用の転置)。

34
35 分類. 変型関数

36
37 引数.

38
39 ARRAY どの型でもよい。配列でなければならない。

40 ORDER (省略可能) 整数型 1 次元配列でなければならない。ARRAY の次元数と等しい大
41 きさを持たなければならない。その要素は、 $(1, 2, \dots, n)$ の順列でなければ
42 ならない。ここで n は、RANK(ARRAY) とする。

43
44 結果の型、型パラメタ 及び 形状. ARRAY と同じ次元数、型、型パラメタの配列とす
45 る。形状は、RS(ORDER) == AS という関係を満たす。ここで、RS は結果の形状とし、
46 AS は SHAPE(ARRAY) とする。ORDER が省略されたとき、既定値は $(n, n - 1, \dots, 1)$ と
47 する。ここで n は、RANK(ARRAY) とする。
48

結果の値. 結果の要素 (j_1, j_2, \dots, j_n) は、 $\text{ARRAY}(j_{order(1)}, j_{order(2)}, \dots, j_{order(n)})$ とする。

例示. 2次元配列に対する $\text{TRANSPOSE}(\text{ARRAY})$ は、通常の行列の転置になる。

ARRAYの形状が $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ で、 $\text{ARRAY}(1, :, :)$ が、 $\begin{bmatrix} 111 & 112 & 113 \\ 121 & 122 & 123 \end{bmatrix}$ で、ORDERが $\begin{bmatrix} 3 & 1 & 2 \end{bmatrix}$ のとき、結果の形状は、 $\begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$ になる。Rを結果とすると、 $\text{R}(:, :, 1)$ は $\text{ARRAY}(1, :, :)$ と同じになる。ARRAYの第 i 次元が、結果の第 $\text{ORDER}(i)$ 次元となる。

12.2 拡張ライブラリ手続きの仕様

HPF_ALIGNMENT(ALIGNEE, LB, UB, STRIDE, AXIS_MAP, IDENTITY_MAP, DYNAMIC, NCOPIES)

省略可能な引数. LB, UB, STRIDE, AXIS_MAP, IDENTITY_MAP, DYNAMIC, NCOPIES

機能. 変数と、それが最終的に整列する *align-target* (配列またはテンプレート) との対応に関する情報を返す。

分類. マッピング問合せサブルーチン

引数.

ALIGNEE どの型でもよく、スカラーでも配列でもよい。大きさ引継ぎ配列であってはならない。空状態のポインタ又は割り付けられていない割り付け配列であってはならない。INTENT (IN) 引数とする。

ALIGNEEがポインタであるときは、指示先の整列に関する情報が返されるものとする。指示先は大きさ引継ぎの仮引数又は大きさ引継ぎの仮引数の部分配列であってはならない。

LB (省略可能) 基本整数型の1次元配列でなければならない。少なくともALIGNEEの次元数に等しい大きさを持たなければならない。INTENT (OUT) 引数とする。ALIGNEEの i 番目の次元軸の最初の要素が、対応する *align-target* の次元軸の $\text{LB}(i)$ 番目の要素に最終的に整列している。ALIGNEEの i 番目の次元軸が縮退しているとき、 $\text{LB}(i)$ に設定される値は実装依存とする。

UB (省略可能) 基本整数型の1次元配列でなければならない。少なくともALIGNEEの次元数に等しい大きさを持たなければならない。INTENT (OUT) 引数とする。ALIGNEEの i 番目の次元軸の最後の要素が、対応する *align-target* の次元軸の $\text{UB}(i)$ 番目の要素に最終的に整列している。ALIGNEEの i 番目の次元軸が縮退しているとき、 $\text{UB}(i)$ に設定される値は実装依存とする。

STRIDE (省略可能) 基本整数型の1次元配列でなければならない。少なくともALIGNEEの次元数に等しい大きさを持たなければならない。INTENT (OUT) 引数とする。STRIDEの i 番目の要素には、ALIGNEEの i 番目の次元軸に沿って各要素が整列するときの刻み幅が設定される。ALIGNEEの i 番目の次元軸が縮退しているとき、 $\text{STRIDE}(i)$ は0に設定される。

1 AXIS_MAP (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも
2 ALIGNEE の次元数に等しい大きさを持たなければならない。INTENT (OUT)
3 引数とする。AXIS_MAP の i 番目の要素には、ALIGNEE の i 番目の次元軸に
4 対応する *align-target* の次元軸が設定される。ALIGNEE の i 番目の次元軸が
5 縮退しているとき、AXIS_MAP(i) は 0 に設定される。
6

7 IDENTITY_MAP (省略可能) 基本論理型スカラでなければならない。INTENT (OUT) 引
8 数とする。ALIGNEE に対応する最終 *align-target* が ALIGNEE と同じ形状を
9 持ち、次元軸が恒等置換を使ってマップされ、刻み幅がすべて正である (し
10 たがって形状の制限から 1 に等しい) とき、真が設定される。それ以外の
11 とき、偽が設定される。変数が、ALIGN 指示文や REALIGN 指示文の *alignee*
12 に現れず、INHERIT 属性も持たないとき、IDENTITY_MAP は真に設定され
13 る。ただし、その他の状況でも真となることがある。
14

15 DYNAMIC (省略可能) 基本論理型スカラでなければならない。INTENT (OUT) 引数とす
16 る。ALIGNEE が DYNAMIC 属性を持つとき、真が設定される。それ以外のと
17 き、偽が設定される。
18

19 NCOPIES (省略可能) 基本整数型スカラでなければならない。INTENT (OUT) 引数とす
20 る。*align-target* に最終的に整列している ALIGNEE の複製の数が設定され
21 る。複製がないときは 1 が設定される。
22

23 ALIGNEE がスカラのとき、LB、UB、STRIDE、AXIS_MAP には何も設定されない。

24
25 例示. 以下の宣言が与えられ、

```
26  
27           REAL PI = 3.1415927  
28           DIMENSION A(10,10),B(20,30),C(20,40,10),D(40)  
29           !HPF$ TEMPLATE T(40,20)  
30           !HPF$ DYNAMIC A  
31           !HPF$ ALIGN A(I,:) WITH T(1+3*I,2:20:2)  
32           !HPF$ ALIGN C(I,*,J) WITH T(J,21-I)  
33           !HPF$ ALIGN D(I) WITH T(I,4)  
34           !HPF$ PROCESSORS PROCS(4,2), SCALARPROC  
35           !HPF$ DISTRIBUTE T(BLOCK,BLOCK) ONTO PROCS  
36           !HPF$ DISTRIBUTE B(CYCLIC,BLOCK) ONTO PROCS  
37           !HPF$ DISTRIBUTE ONTO SCALARPROC :: PI  
38  
39
```

40 実際のマッピングが宣言通りに行なわれたとすると、HPF_ALIGNMENT の呼出しの結果は以下
41 のようになる。
42
43
44
45
46
47
48

	A	B	C	D
LB	[4, 2]	[1, 1]	[20, N/A, 1]	[1]
UB	[31, 20]	[20, 30]	[1, N/A, 10]	[40]
STRIDE	[3, 2]	[1, 1]	[-1, 0, 1]	[1]
AXIS_MAP	[1, 2]	[1, 2]	[2, 0, 1]	[1]
IDENTITY_MAP	偽	真	偽	偽
DYNAMIC	真	偽	偽	偽
NCOPIES	1	1	1	1

ここで、「N/A」は実装依存の結果を表わす。NCOPIES の利用法を次の例で見る。

```
LOGICAL BOZO(20,20),RONALD_MCDONALD(20)
!HPF$ TEMPLATE EMMETT_KELLY(100,100)
!HPF$ ALIGN RONALD_MCDONALD(I) WITH BOZO(I,*)
!HPF$ ALIGN BOZO(J,K) WITH EMMETT_KELLY(J,5*K)
```

このとき、CALL HPF_ALIGNMENT(RONALD_MCDONALD, NCOPIES = NC) は NC に 20 を設定する。また次の例では、

```
LOGICAL BOZO(20,20),RONALD_MCDONALD(20)
!HPF$ TEMPLATE WILLIE_WHISTLE(100)
!HPF$ ALIGN RONALD_MCDONALD(I) WITH BOZO(I,*)
!HPF$ ALIGN BOZO(J,*) WITH WILLIE_WHISTLE(5*J)
```

このとき、CALL HPF_ALIGNMENT(RONALD_MCDONALD, NCOPIES = NC) は NC に 1 を設定する。

HPF_DISTRIBUTION(DISTRIBUTE, AXIS_TYPE, AXIS_INFO, PROCESSORS_RANK, PROCESSORS_SHAPE, PLB, PUB, PSTRIDE, LOW_SHADOW, HIGH_SHADOW)

省略可能な引数. AXIS_TYPE, AXIS_INFO, PROCESSORS_RANK, PROCESSORS_SHAPE, PLB, PUB, PSTRIDE, LOW_SHADOW, HIGH_SHADOW.

機能. 変数の最終 *align-target* の分散に関する情報を返す。

分類. マッピング問合せサブルーチン

引数.

DISTRIBUTE どの型でもよく、スカラーでも配列でもよい。連続であってはならない。空状態のポインタ又は割り付けられていない割付け配列であってはならない。INTENT (IN) 引数とする。

AXIS_TYPE (省略可能) 基本文字型の 1 次元配列でなければならない。長さは任意であるが、完全な値を得るためには少なくとも長さが 9 必要である。各要素には以下の値が文字組込み代入文と同じ方法で設定される。少なくとも

1 DISTRIBUTEE が最終的に整列している *align-target* の次元数と等しい大き
2 さを持たなければならない (これは HPF_TEMPLATE が TEMPLATE_RANK に返
3 す値である)。INTENT (OUT) 引数とする。 *i* 番目の要素には、 *align-target*
4 の *i* 番目の次元軸の分散に関する情報が設定される。HPF では以下の値が
5 定義されている (実装によってはその他の値が定義されていることもある)。

6
7 'BLOCK' この次元軸は BLOCK 分散されている。AXIS_INFO の対応する要
8 素にはブロックの大きさが設定される。

9 'GEN_BLOCK' この次元軸は、GEN_BLOCK(array)¹ で分散されている。
10 AXIS_INFO の対応する要素の値は実装依存とする。

11 'COLLAPSED' この次元軸は縮退している (「*」指定により分散されてい
12 る)。AXIS_INFO の対応する要素の値は実装依存とする。

13
14 'CYCLIC' この次元軸は CYCLIC 分散されている。AXIS_INFO の対応する
15 要素にはブロックの大きさが設定される。

16
17 'INDIRECT' この次元軸は、INDIRECT(map-array) で分散されている。
18 AXIS_INFO の対応する要素の値は実装依存とする。

19
20 AXIS_INFO (省略可能) 基本整数型の 1 次元配列でなければならない。少なくと
21 も DISTRIBUTEE が最終的に整列している *align-target* の次元数と等しい大き
22 さを持たなければならない (これは HPF_TEMPLATE が TEMPLATE_RANK に返
23 す値である)。INTENT (OUT) 引数とする。AXIS_INFO の *i* 番目の要素には、
24 DISTRIBUTEE の最終 *align-target* の *i* 番目の次元軸の、ブロック分散又は
25 サイクリック分散でのブロックの大きさが設定される。次元軸が縮退して
26 いるとき、設定される値は実装に依存する。

27
28 PROCESSORS_RANK (省略可能) 基本整数型スカラでなければならない。DISTRIBUTEE
29 が分散されるプロセッサ構成の次元数が設定される。INTENT (OUT) 引数
30 とする。

31
32 PROCESSORS_SHAPE (省略可能) 基本整数型の 1 次元配列でなければならない。少なくと
33 も PROCESSORS_RANK の返す値 *m* に等しい大きさを持たなければならない。
34 INTENT (OUT) 引数とする。最初の *m* 個の要素に、DISTRIBUTEE がマップさ
35 れるプロセッサ構成の形状が設定される (HPF_DISTRIBUTION は 2 度呼び出
36 す必要があるかもしれない。すなわち最初の呼び出しで PROCESSORS_RANK
37 の値を得、それにより PROCESSORS_SHAPE を割り付けるのである)。

38
39 PLB (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも DISTRIBUTEE
40 の最終 *align-target* の次元数に等しい大きさを持たなければならない。
41 INTENT (OUT) 引数とする。PLB の *i* 番目の要素には、DISTRIBUTEE の最
42 終 *align-target* の *i* 番目の次元軸がマップされているプロセッサの最小の添
43 字が設定される。次元軸が縮退しているとき、設定される値は実装に依存
44 する。設定される値は 1 から e_i の間とする。ここで e_i は、DISTRIBUTEE
45 の最終 *align-target* の選択された次元軸がマップされているプロセッサ構
46 成の次元軸の寸法とする。
47

48 ¹ 訳注: 原文では「BLOCK(array) となっているが、誤りと思われる

PUB (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも DISTRIBUTE の最終 *align-target* の次元数に等しい大きさを持たなければならない。INTENT (OUT) 引数とする。PUB の *i* 番目の要素には、DISTRIBUTE の最終 *align-target* の *i* 番目の次元軸がマップされているプロセッサの最大の添字が設定される。次元軸が縮退しているとき、設定される値は実装に依存する。設定される値は 1 から e_i の間とする。ここで e_i は、DISTRIBUTE の最終 *align-target* の選択された次元軸がマップされているプロセッサ構成の次元軸の寸法とする。

PSTRIDE (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも DISTRIBUTE の次元数に等しい大きさを持たなければならない。INTENT (OUT) 引数とする。PSTRIDE の *i* 番目の要素には、DISTRIBUTE の *i* 番目の次元軸がマップされる ONTO 節中の、プロセッサ間の刻み幅が設定される。次元軸が縮退しているときは、0 が設定される。

LOW_SHADOW (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも DISTRIBUTE の最終 *align-target* の次元数に等しい大きさを持たなければならない (これは HPF_TEMPLATE が TEMPLATE_RANK に返す値である)。INTENT (OUT) 引数とする。LOW_SHADOW の *i* 番目の要素には、DISTRIBUTE の最終 *align-target* の *i* 番目の次元軸の、ブロック分散又はサイクリック分散での下方のシャドウ幅が設定される。次元軸が縮退しているとき、設定される値は実装に依存する。

HIGH_SHADOW (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも DISTRIBUTE の最終 *align-target* の次元数に等しい大きさを持たなければならない (これは HPF_TEMPLATE が TEMPLATE_RANK に返す値である)。INTENT (OUT) 引数とする。HIGH_SHADOW の *i* 番目の要素には、DISTRIBUTE の最終 *align-target* の *i* 番目の次元軸の、ブロック分散又はサイクリック分散での上方のシャドウ幅が設定される。次元軸が縮退しているとき、設定される値は実装に依存する。

例示. HPF_ALIGNMENT の例に示す宣言が与えられ、実際のマッピングが宣言通りに行なわれたとすると、HPF_DISTRIBUTE の呼出しの結果は以下ようになる。

	A	B	PI
AXIS_TYPE	['BLOCK', 'BLOCK']	['CYCLIC', 'BLOCK']	[]
AXIS_INFO	[10, 10]	[1, 15]	[]
PROCESSORS_SHAPE	[4, 2]	[2, 2]	[]
PROCESSORS_RANK	2	2	0
PLB	[4, 1]	[2, 1]	[]
PUB	[1, 2]	[3, 2]	[]
PSTRIDE	[-1,1]	[1, 1]	[]

1 HPF_TEMPLATE(ALIGNEE, TEMPLATE_RANK, LB, UB,
2 AXIS_TYPE, AXIS_INFO, NUMBER_ALIGNED, DYNAMIC)

3 省略可能な引数. LB, UB, AXIS_TYPE, AXIS_INFO, NUMBER_ALIGNED,
4 TEMPLATE_RANK, DYNAMIC
5

6
7 機能. 変数の最終 *align-target* に関する情報を返す。HPF_ALIGNMENT が変数の観点から
8 の情報を返すのに対し、HPF_TEMPLATE は最終 *align-target* の観点からの変数に関する
9 情報を返す。
10

11 分類. マッピング問合せサブルーチン

12
13 引数.

14
15 ALIGNEE どの型でもよく、スカラでも配列でもよい。大きさ引継ぎ配列であっては
16 ならない。空状態のポインタ又は割り付けられていない割付け配列であっ
17 てはならない。INTENT (IN) 引数とする。

18 ALIGNEE がポインタであるときは、指示先の整列に関する情報が返される
19 ものとする。指示先は大きさ引継ぎの仮引数又は大きさ引継ぎの仮引数の
20 部分配列であってはならない。
21

22 TEMPLATE_RANK (省略可能) 基本整数型スカラでなければならない。INTENT (OUT) 引
23 数とする。最終 *align-target* の次元数が設定される。この値は、縮退や複
24 製により ALIGNEE の次元数と異なることがある。
25

26 LB (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも ALIGNEE が
27 最終的に整列している *align-target* の次元数と等しい大きさを持たなければ
28 ならない (これは TEMPLATE_RANK に設定される値である)。INTENT (OUT)
29 引数とする。LB の *i* 番目の要素には、テンプレートの *i* 番目の次元軸に対
30 する *align-target* の宣言時の下限が設定される。
31

32 UB (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも ALIGNEE が
33 最終的に整列している *align-target* の次元数と等しい大きさを持たなければ
34 ならない (これは TEMPLATE_RANK に設定される値である)。INTENT (OUT)
35 引数とする。UB の *i* 番目の要素には、テンプレートの *i* 番目の次元軸に対
36 する *align-target* の宣言時の上限が設定される。
37

38 AXIS_TYPE (省略可能) 基本文字型の 1 次元配列でなければならない。長さは任意で
39 あるが、完全な値を得るためには少なくとも長さが 10 必要である。各要
40 素には以下の値が文字組み代入文と同じ方法で設定される。少なくと
41 も ALIGNEE が最終的に整列している *align-target* の次元数と等しい大き
42 さを持たなければならない (これは TEMPLATE_RANK に設定される値である)。
43 INTENT (OUT) 引数とする。AXIS_TYPE の *i* 番目の要素には、*align-target*
44 の *i* 番目の次元軸に関する情報が設定される。HPF では以下の値が定義さ
45 れている (実装によりその他の値が定義されていることがある)。
46
47
48

'NORMAL' *align-target*のこの次元軸には、ALIGNEEのある次元軸が整列している。この値が代入された `AXIS_TYPE` の要素に対しては、対応する `AXIS_INFO` の要素に、この *align-target* の次元軸に整列する ALIGNEE の次元軸の番号が設定される。

'REPLICATED' *align-target*のこの次元軸には、ALIGNEEが複製されている。この値が代入された `AXIS_TYPE` の要素に対しては、対応する `AXIS_INFO` の要素に、この *align-target* の次元軸に複製された ALIGNEE の複製の数が設定される。

'SINGLE' *align-target* のこの次元軸のある座標に、ALIGNEE が整列している。この値が代入された `AXIS_TYPE` の要素に対しては、対応する `AXIS_INFO` の要素に、ALIGNEE が整列している *align-target* の座標が設定される。

`AXIS_INFO` (省略可能) 基本整数型の 1 次元配列でなければならない。少なくとも ALIGNEE が最終的に整列している *align-target* の次元数と等しい大きさを持たなければならない (これは `TEMPLATE_RANK` に設定される値である)。`INTENT (OUT)` 引数とする。上記 `AXIS_TYPE` の項を参照されたい。

`NUMBER_ALIGNED` (省略可能) 基本整数型スカラでなければならない。`INTENT (OUT)` 引数とする。この最終 *align-target* に整列している変数の総数が設定される。これは、この *align-target* が再分散されたときに同時に移動する変数の数である。

`DYNAMIC` (省略可能) 基本論理型スカラでなければならない。`INTENT (OUT)` 引数とする。*align-target* が `DYNAMIC` 属性を持つとき、真が設定される。それ以外するとき、偽が設定される。

例示. `HPF_ALIGNMENT` の例に示す宣言が与えられ、実際のマッピングが宣言通りに行なわれたとすると、`HPF_TEMPLATE` の呼出しの結果は以下のようになる。

	A	C	D
LB	[1, 1]	[1, 1]	[1, 1]
UB	[40, 20]	[40, 20]	[40, 20]
AXIS_TYPE	['NORMAL', 'NORMAL']	['NORMAL', 'NORMAL']	['NORMAL', 'SINGLE']
AXIS_INFO	[1, 2]	[3, 1]	[1, 4]
NUMBER_ALIGNED	3	3	3
TEMPLATE_RANK	2	2	2
DYNAMIC	false	false	false

HPF_MAP_ARRAY(`ARRAY`, `TEMPLATE_DIM`, `MAP_ARRAY`)

機能. `ARRAY` の最終 *align-target* の第 `TEMPLATE_DIM` 次元について、間接分散で用いられるマップ配列を返す。

分類. マッピング問合せサブルーチン

1 引数.

2 ARRAY どの型でもよい。スカラであってはならない。連続であってはならない。
3 空状態のポインタ又は割り付けられていない割付け配列であってはならない。
4
5 INTENT (IN) 引数とする。

6 TEMPLATE_DIM 基本整数型スカラでなければならない。1 と ARRAY の最終 *align-target*
7 の次元数の間の値でなければならない。INTENT (IN) 引数とする。
8

9 MAP_ARRAY 基本文字型の 1 次元配列でなければならない。ARRAY の最終 *align-target*
10 が分散しているプロセッサ構成の、第 PROCESSORS_DIM 次元の寸法以上の
11 大きさを持たなければならない。INTENT (OUT) 引数とする。
12

13 MAP_ARRAY の *i* 番目の要素には、ARRAY の最終 *align-target* の第
14 TEMPLATE_DIM 次元の、*i* 番目の要素がマップされるプロセッサの添
15 字が設定される。ARRAY の最終 *align-target* の第 TEMPLATE_DIM 次元が縮
16 退しているとき、すべての要素に 1 が設定される。
17

18 例示. 以下の宣言が与えられ、
19

```
20     DIMENSION A(2)  
21     !HPF$ TEMPLATE T(4,8)  
22     !HPF$ ALIGN A(I,*) WITH T(2*I,5)  
23     !HPF$ PROCESSORS PROCS(2,2)  
24     !HPF$ DISTRIBUTE T(INDIRECT( (/1,2,2,1/) ), BLOCK( (/3,5/) )) ONTO PROCS
```

25
26 実際のマッピングが宣言通りに行なわれたとすると、HPF_MAP_ARRAY(A,TEMPLATE_DIM=1,
27 MAP_ARRAY=M) の呼出しの結果、M の値は $\begin{bmatrix} 1 & 2 & 2 & 1 \end{bmatrix}$
28 になる。また HPF_MAP_ARRAY(A,TEMPLATE_DIM=2, MAP_ARRAY=M) の呼出しの結果、M の値は
29 $\begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$ になる。
30
31

32 HPF_NUMBER_MAPPED(ARRAY, PROCESSORS_DIM,
33 NUMBER_MAPPED)
34

35 機能. ARRAY の最終 *align-target* が分散されるプロセッサ構成の、第 PROCESSORS_DIM
36 次元の各要素にマップされる、ARRAY の最終 *align-target* の要素数を返す。
37

38 分類. マッピング問合せサブルーチン
39

40 引数.

41
42 ARRAY どの型でもよい。スカラであってはならない。連続であってはならない。空
43 状態のポインタ又は割り付けられていない割付け配列であってはならない。
44

45 PROCESSORS_DIM 基本整数型スカラでなければならない。1 と ARRAY の最終 *align-*
46 *target* が分散されるプロセッサ構成の次元数との間の値でなければならない。
47 INTENT (IN) 引数とする。
48

NUMBER_MAPPED 基本整数型の 1 次元配列でなければならない。ARRAY の最終 *align-target* が分散しているプロセッサ構成の、第 PROCESSORS_DIM 次元の寸法以上の大きさを持たなければならない。NUMBER_MAPPED の *i* 番目の要素には、ARRAY の最終 *align-target* が分散されるプロセッサ構成の、第 PROCESSORS_DIM 次元の *i* 番目のプロセッサにマップされる、ARRAY の最終 *align-target* の対応する次元の要素数が設定される。ARRAY の最終 *align-target* が分散されるプロセッサ構成の第 PROCESSORS_DIM 次元が、BLOCK 分散の次元と対応づけられているとき、MAP_ARRAY にはその次元を分散するのに用いられるブロックの大きさの配列が設定される。

例示. 以下の宣言が与えられ、

```
DIMENSION A(2,40)
!HPF$ TEMPLATE T(4,8,4,16)
!HPF$ ALIGN A(I,*) WITH T(2*I, 5, *, *)
!HPF$ PROCESSORS PROCS(2,2,3)
!HPF$ DISTRIBUTE T(INDIRECT((/2,2,1,2/)), BLOCK((/3,5/)), *, BLOCK) &
!HPF$      ONTO PROCS
```

実際のマッピングが宣言通りに行なわれたとすると、HPF_NUMBER_MAPPED(A, PROCESSORS_DIM=1, NUMBER_MAPPED = M) の呼出しの結果、M の値は $\begin{bmatrix} 1 & 3 \end{bmatrix}$ になり、HPF_NUMBER_MAPPED(A, PROCESSORS_DIM=2, NUMBER_MAPPED = M) の呼出しの結果、M の値は $\begin{bmatrix} 3 & 5 \end{bmatrix}$ になり、HPF_NUMBER_MAPPED(A, PROCESSORS_DIM=3, NUMBER_MAPPED = M) の呼出しの結果、M の値は $\begin{bmatrix} 6 & 6 & 4 \end{bmatrix}$ になる。