



HPF言語の拡張について

2003.9.25

HPFPC

NEWでない変数の指定(1)

■ 背景

- NEW変数をいちいち指定するのが面倒
- 特に定義されるスカラの場合、以下の2つの場合以外は全てのスカラに対してNEW指定が必要なので、わざわざ指定したくない。
 - reduction変数
 - 最外側の INDEPENDENT DO ループの DO 変数の終値を保証したい場合

```
!hpf$ independent,new(j),reduction(s)
do i=1,n
!hpf$ independent 最外側 INDEPENDENT DO 変数
do j=1,n
s = s + a(j,i)
enddo
enddo
```

reduction 変数

最外側 INDEPENDENT DO 変数

NEWでない変数の指定(2)

- スカラ変数を簡単に指定できるようにする

!HPF\$ INDEPENDENT, NEW **SCALAR** [(配列名並び)]

- 定義されるスカラのうち、reduction節に記述されていない全てのスカラ変数名をNEWと見なす

- 配列のNEW変数がたくさんある場合

- NEWでない変数の方を指定できるようにする。

!HPF\$ INDEPENDENT, **NONEW**(NEWでない変数並び)

- 記述された変数とreduction変数以外で、定義されるものは全てNEW

さらに、

- NEWでない変数は多くの場合マップされている配列
- NEW変数はスカラとマップされていない配列

!HPF\$ INDEPENDENT, **NONEW MAPPED ARRAY**

- マップされた配列とreduction変数以外で、定義されるものは全てNEW変数

HPF標準の計時関数(1)

■ 背景

- 計時関数の仕様が各社バラバラ
- Fortran標準のsystem_clockは、単精度のため非実用的(これは、Fortranでも問題)
- HPFの場合、ユーザ見えは逐次コードと同じだが実際には複数のプロセッサで実行されるため、各プロセッサ毎に、実行時間は異なる。

例)

```
t1 = elapsed_time()  
      :  
t2 = elapsed_time()  
iteration = iCnst / (t2 - t1) ! 実行回数を決める
```

- t1,t2の計時毎にブロードキャストを発生させると計時のためのオーバーヘッドが大きすぎる。
- 通信を行わないと各プロセッサ毎にt1,t2の値が異なるので、iteration(実行回数)も各プロセッサ毎に異なってしまう。

HPF標準の計時関数(2)

1) 各プロセッサ毎の経過時間を公開する

```
!hpf$ distribute (block) :: t1,t2
  double precision, dimension(number_of_processors()) :: t1,t2
  call HPF_LOCAL_WTIME(t1)
      :
  call HPF_LOCAL_WTIME(t2)
```

- t1(i),t2(i)には、i番目のプロセッサの経過時間
- 通信はいっさい発生しない。

2-1) 途中の経過時間はシステム内に隠蔽する。

```
integer id1,id2
call HPF_CHECK_WTIME(id1)           ! id1にシステムからID値が返る
      :
call HPF_CHECK_WTIME(id2)           ! id2にシステムからID値が返る
write(*,*) "Time=", HPF_WTIME(id2,id1) ! 経過時間取得
```

- HPF_CHECK_WTIMEでは、通信はいっさい発生しない。
- HPF_WTIMEで、ブロードキャスト発生

HPF標準の計時関数(3)

■ 2-2) ループで加算したい場合

```
t = 0
do i=1, iteration
  t1 = elapsed_time()
  :
  t2 = elapsed_time()
  t = t + t2 - t1
enddo
write(*,*) "Time=", t
```

! 初期化

! 実行時間の加算



```
call HPF_ACCUM_INIT(ia1)
do i=1, iteration
  call HPF_CHECK_WTIME(id1)
  :
  call HPF_CHECK_WTIME(id2)
  call HPF_ACCUM_ADD(ia1, id2, id1)
enddo
write(*,*) "Time=", HPF_ACCUM_WTIME(ia1) ! 経過時間取得
```

! システム内アキュムレータ初期化

! システム内アキュムレータに加算

- HPF_CHECK_WTIME、HPF_ACCUM_INIT、HPF_ACCUM_ADDでは、通信はいっさい発生しない。
- HPF_ACCUM_WTIMEで、ブロードキャスト発生