グループ通信による計算例

- ベクトルの内積
- Scatter/Gather
- 分散ファイルの読み込み
- MPI_Allgatherv

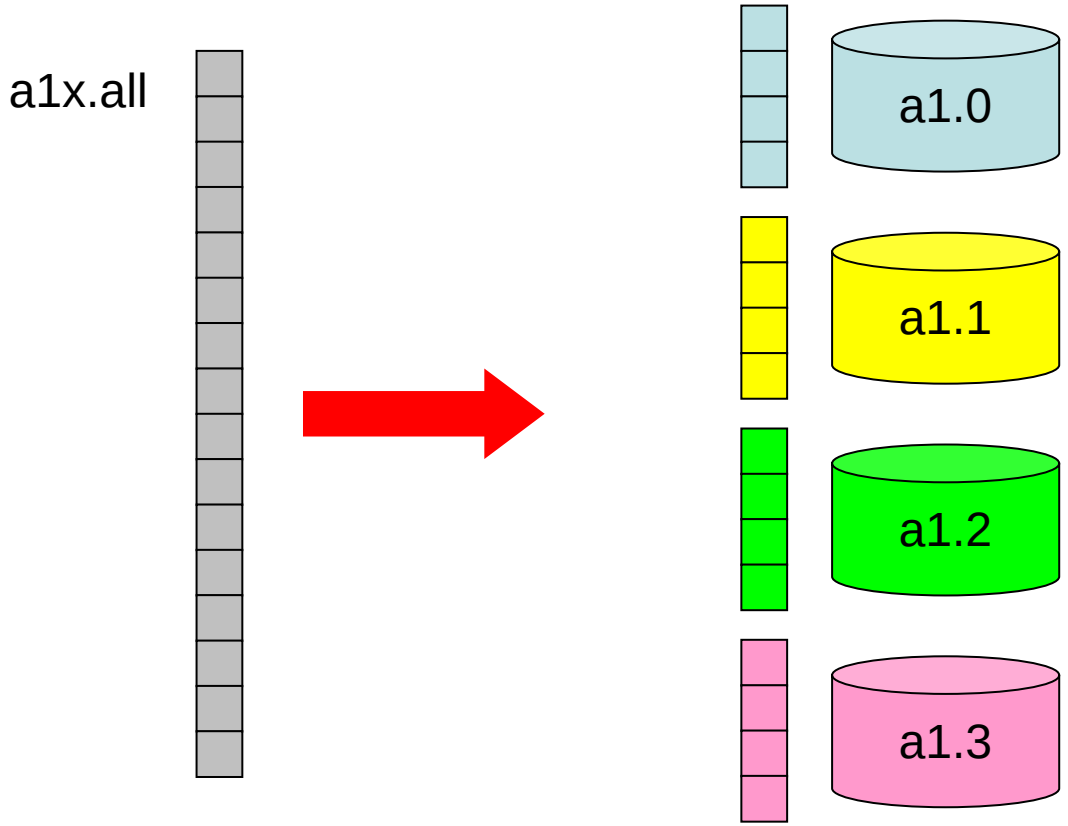
分散ファイルを使用したオペレーション

- Scatter/Gatherの例では，PE#0から全体データを読み込み，それを全体にScatterして並列計算を実施した。

- 問題規模が非常に大きい場合，1つのプロセッサで全てのデータを読み込むことは不可能な場合がある。
  - 最初から分割しておいて，「局所データ」を各プロセッサで独立に読み込む
  - あるベクトルに対して，全体操作が必要になった場合は，状況に応じてMPI_Gatherなどを使用する

分散ファイルの操作

- 「a1.0~a1.3」は全体ベクトル「a1x.all」を領域に分割したもの，と考えることができる。



分散ファイル読み込み：等データ長

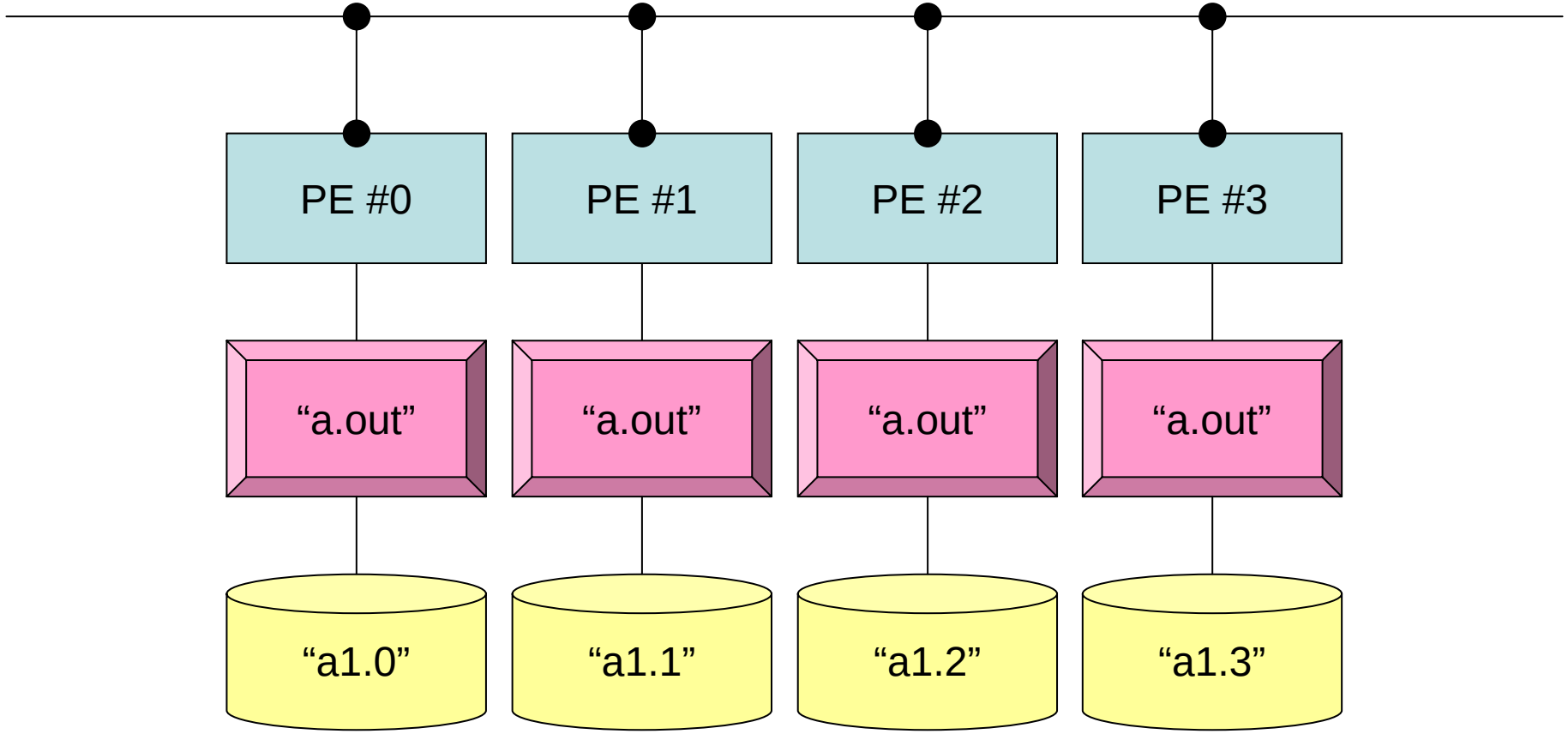
<$S1>/file.f

```
implicit REAL*8 (A-H,O-Z)
include 'mpif.h'
integer :: PETOT, my_rank, ierr
real(kind=8), dimension(8) :: VEC
character(len=80)          :: filename

call MPI_INIT      (ierr)
call MPI_COMM_SIZE (MPI_COMM_WORLD, PETOT, ierr )
call MPI_COMM_RANK (MPI_COMM_WORLD, my_rank, ierr )
call MPI_COMM_DUP  (MPI_COMM_WORLD, SOLVER_COMM, ierr)

if (my_rank.eq.0) filename= 'a1.0'
if (my_rank.eq.1) filename= 'a1.1'
if (my_rank.eq.2) filename= 'a1.2'
if (my_rank.eq.3) filename= 'a1.3'

open (21, file= filename, status= 'unknown')
  do i= 1, 8
    read (21,*) VEC(i)
  enddo
close (21)

call MPI_FINALIZE (ierr)

stop
end
```

Hello とそんなに変わらない

「局所番号(1~8)」で読み込む

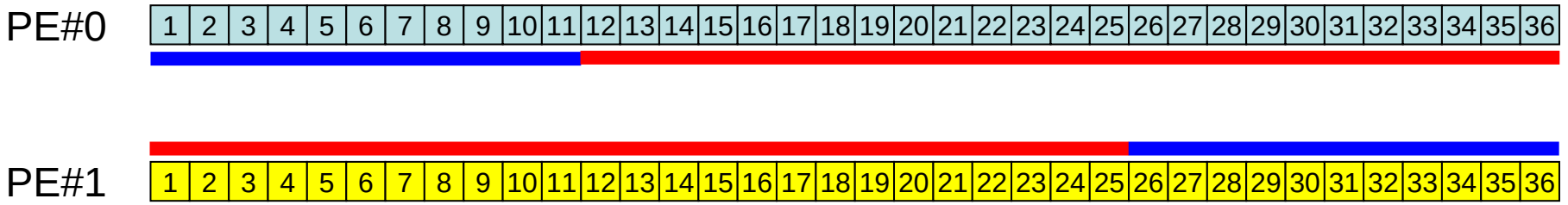
SPMDの典型例




```
mpirun -np 4 a.out
```

利用例(2):配列の送受信(1/4)

- PE#0，PE#1間 で8バイト実数配列VECの値を交換する。
- PE#0⇒PE#1
  - PE#0:VEC(1)～VEC(11)の値を送る(長さ:11)
  - PE#1:VEV(26)~VEC(36)の値として受け取る
- PE#1⇒PE#0
  - PE#1:VEC(1)～VEC(25)の値を送る(長さ:25)
  - PE#0:VEV(12)~VEC(36)の値として受け取る



演　習

- VEC(:)の初期状態を以下のようにする:
  - PE#0　VEC(1-36)= 100
  - PE#1　VEC(1-36)= 200

- 以下のそれぞれを使用したプログラムを作成せよ
  - MPI_Isend/Irecv/Waitall
  - MPI_Sendrecv

- 正解は以下にある

```
<$S2>/ex2a.f, <$S2>/ex2b.f
<$S2>/ex2a.c, <$S2>/ex2b.c
```

配列送受信（ex2a.f）（1/3）

Isend/Irecv/Waitall

```
$> cd <$S2>
$> mpif90 -O3 ex2a.f
$> mpirun -np 2 a.out
```

```
      integer(kind=4) :: my_rank, PETOT, NEIB
      real   (kind=8) :: VEC(36)

      integer(kind=4), dimension(MPI_STATUS_SIZE),1) :: stat_send
      integer(kind=4), dimension(MPI_STATUS_SIZE),1) :: stat_recv
      integer(kind=4), dimension(1)    :: request_send
      integer(kind=4), dimension(1)    :: request_recv

      integer(kind=4) :: start_send, length_send
      integer(kind=4) :: start_recv, length_recv

      call MPI_INIT      (ierr)
      call MPI_COMM_SIZE (MPI_COMM_WORLD, PETOT, ierr )
      call MPI_COMM_RANK (MPI_COMM_WORLD, my_rank, ierr )
```

配列送受信（ex2a.f）（2/3）

Isend/Irecv/Waitall

```
      if (my_rank.eq.0) then
        NEIB= 1
        start_send=   1
        length_send=  11
        start_recv=  length_send + 1
        length_recv=  25
        VEC= 100.
      endif

      if (my_rank.eq.1) then
        NEIB= 0
         start_send=  1
        length_send= 25
         start_recv= length_send + 1
        length_recv= 11
        VEC= 200.
      endif
```

配列送受信（ex2a.f）(3/3)

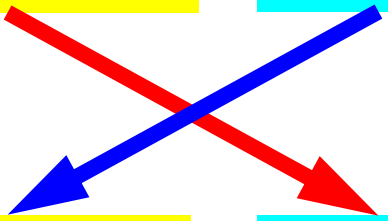
Isend/Irecv/Waitall

```
      do i= 1, 36
        write (*,'(i1,a,i2,f10.0)') my_rank, ' #BEFORE# ', i,VEC(i)
      enddo

      call MPI_ISEND (VEC(start_send), length_send,                     &
     &                MPI_DOUBLE_PRECISION, NEIB, 0, MPI_COMM_WORLD,    &
     &                request_send(1), ierr)
      call MPI_IRECV (VEC(start_recv), length_recv,                     &
     &                MPI_DOUBLE_PRECISION, NEIB, 0, MPI_COMM_WORLD,    &
     &                request_recv(1), ierr)

      call MPI_WAITALL (1, request_recv, stat_recv, ierr)
      call MPI_WAITALL (1, request_send, stat_send, ierr)

      do i= 1, 36
        write (*,'(i1,a,i2,f10.0)') my_rank, ' #AFTER # ', i,VEC(i)
      enddo

      call MPI_FINALIZE (ierr)

      end
```

配列の送受信:注意

```
#PE0
  send:
    VEC(start_send)~
    VEC(start_send+length_send-1)
```

```
#PE1
  send:
    VEC(start_send)~
    VEC(start_send+length_send-1)
```



```
#PE0
  recv:
    VEC(start_recv)~
    VEC(start_recv+length_recv-1)
```

```
#PE1
  recv:
    VEC(start_recv)~
    VEC(start_recv+length_recv-1)
```

- 送信側の「length_send」と受信側の「length_recv」は一致している必要がある。
  - PE#0⇒PE#1，PE#1⇒PE#0
- 「送信バッファ」と「受信バッファ」は別のアドレス

結　果

0 #BEFORE# 1 100.	0 #AFTER # 1 100.	1 #BEFORE# 1 200.	1 #AFTER # 1 200.
0 #BEFORE# 2 100.	0 #AFTER # 2 100.	1 #BEFORE# 2 200.	1 #AFTER # 2 200.
0 #BEFORE# 3 100.	0 #AFTER # 3 100.	1 #BEFORE# 3 200.	1 #AFTER # 3 200.
0 #BEFORE# 4 100.	0 #AFTER # 4 100.	1 #BEFORE# 4 200.	1 #AFTER # 4 200.
0 #BEFORE# 5 100.	0 #AFTER # 5 100.	1 #BEFORE# 5 200.	1 #AFTER # 5 200.
0 #BEFORE# 6 100.	0 #AFTER # 6 100.	1 #BEFORE# 6 200.	1 #AFTER # 6 200.
0 #BEFORE# 7 100.	0 #AFTER # 7 100.	1 #BEFORE# 7 200.	1 #AFTER # 7 200.
0 #BEFORE# 8 100.	0 #AFTER # 8 100.	1 #BEFORE# 8 200.	1 #AFTER # 8 200.
0 #BEFORE# 9 100.	0 #AFTER # 9 100.	1 #BEFORE# 9 200.	1 #AFTER # 9 200.
0 #BEFORE# 10 100.	0 #AFTER # 10 100.	1 #BEFORE# 10 200.	1 #AFTER # 10 200.
0 #BEFORE# 11 100.	0 #AFTER # 11 100.	1 #BEFORE# 11 200.	1 #AFTER # 11 200.
0 #BEFORE# 12 100.	**0 #AFTER # 12 200.**	1 #BEFORE# 12 200.	1 #AFTER # 12 200.
0 #BEFORE# 13 100.	**0 #AFTER # 13 200.**	1 #BEFORE# 13 200.	1 #AFTER # 13 200.
0 #BEFORE# 14 100.	**0 #AFTER # 14 200.**	1 #BEFORE# 14 200.	1 #AFTER # 14 200.
0 #BEFORE# 15 100.	**0 #AFTER # 15 200.**	1 #BEFORE# 15 200.	1 #AFTER # 15 200.
0 #BEFORE# 16 100.	**0 #AFTER # 16 200.**	1 #BEFORE# 16 200.	1 #AFTER # 16 200.
0 #BEFORE# 17 100.	**0 #AFTER # 17 200.**	1 #BEFORE# 17 200.	1 #AFTER # 17 200.
0 #BEFORE# 18 100.	**0 #AFTER # 18 200.**	1 #BEFORE# 18 200.	1 #AFTER # 18 200.
0 #BEFORE# 19 100.	**0 #AFTER # 19 200.**	1 #BEFORE# 19 200.	1 #AFTER # 19 200.
0 #BEFORE# 20 100.	**0 #AFTER # 20 200.**	1 #BEFORE# 20 200.	1 #AFTER # 20 200.
0 #BEFORE# 21 100.	**0 #AFTER # 21 200.**	1 #BEFORE# 21 200.	1 #AFTER # 21 200.
0 #BEFORE# 22 100.	**0 #AFTER # 22 200.**	1 #BEFORE# 22 200.	1 #AFTER # 22 200.
0 #BEFORE# 23 100.	**0 #AFTER # 23 200.**	1 #BEFORE# 23 200.	1 #AFTER # 23 200.
0 #BEFORE# 24 100.	**0 #AFTER # 24 200.**	1 #BEFORE# 24 200.	1 #AFTER # 24 200.
0 #BEFORE# 25 100.	**0 #AFTER # 25 200.**	1 #BEFORE# 25 200.	1 #AFTER # 25 200.
0 #BEFORE# 26 100.	**0 #AFTER # 26 200.**	1 #BEFORE# 26 200.	**1 #AFTER # 26 100.**
0 #BEFORE# 27 100.	**0 #AFTER # 27 200.**	1 #BEFORE# 27 200.	**1 #AFTER # 27 100.**
0 #BEFORE# 28 100.	**0 #AFTER # 28 200.**	1 #BEFORE# 28 200.	**1 #AFTER # 28 100.**
0 #BEFORE# 29 100.	**0 #AFTER # 29 200.**	1 #BEFORE# 29 200.	**1 #AFTER # 29 100.**
0 #BEFORE# 30 100.	**0 #AFTER # 30 200.**	1 #BEFORE# 30 200.	**1 #AFTER # 30 100.**
0 #BEFORE# 31 100.	**0 #AFTER # 31 200.**	1 #BEFORE# 31 200.	**1 #AFTER # 31 100.**
0 #BEFORE# 32 100.	**0 #AFTER # 32 200.**	1 #BEFORE# 32 200.	**1 #AFTER # 32 100.**
0 #BEFORE# 33 100.	**0 #AFTER # 33 200.**	1 #BEFORE# 33 200.	**1 #AFTER # 33 100.**
0 #BEFORE# 34 100.	**0 #AFTER # 34 200.**	1 #BEFORE# 34 200.	**1 #AFTER # 34 100.**
0 #BEFORE# 35 100.	**0 #AFTER # 35 200.**	1 #BEFORE# 35 200.	**1 #AFTER # 35 100.**
0 #BEFORE# 36 100.	**0 #AFTER # 36 200.**	1 #BEFORE# 36 200.	**1 #AFTER # 36 100.**