プロセス/スレッド併用型並列処理

- MPIのライブラリの実装に依存する部分がある
- MPIによるプログラムがあるとして
  - MPI+自動並列： もっとも簡単
  - MPI+OpenMP（複合Parallel work-sharing構文）
    - doループやforループを複合Parallel work-sharing構文で並列化
    - 割と簡単で安全
  - MPI+OpenMP（OpenMPのスレッド並列処理をフル活用）
    - 現行の京大スパコンにおいて
    - MPIライブラリの通信関数はマスタスレッドからのみ呼び出す
    - 送受信するデータはSharedのデータであること

1

MPI/OpenMP併用並列処理 （1）

```
integer, parameter :: n=10000
integer :: i
double precision :: a(0:n-1),b(0:n-1)
do i=0,n-1
  a(i)=float(i)
enddo
do i=0,n-1
  b(i)=dsin(a(i))
enddo
end
```

2

MPI/OpenMP併用並列処理 （2）

```
include "mpif.h"
integer, parameter :: n=10000
integer :: i,myid,ieer,nump,ir(0:100),id(0:100),it
double precision :: a(0:n-1),b(0:n-1)
  call MPI_INIT(ieer)
  call MPI_COMM_RANK(MPI_COMM_WORLD,myid,ieer)
  call MPI_COMM_SIZE(MPI_COMM_WORLD,nump,ieer)
  it=n/nump+1
  do i=it*(myid),min(n-1, it*(myid+1)-1)
    a(i)=float(I)
  enddo
  do i=it*(myid),min(n-1, it*(myid+1)-1)
    b(i)=dsin(a(i))
  enddo
```

```
do i=0,nump-1
   id(i)=it*i
enddo

do i=0,nump-2
   ir(i)=it
Enddo
ir(nump-1)=n - it*(nump-1)

call  MPI_ALLGATHERV(b(it*(myid)), ir(myid),
    MPI_DOUBLE_PRECISION, b(0),  ir(0), id(0),
    MPI_DOUBLE_PRECISION,
    MPI_COMM_WORLD, ieer)
call MPI_FINALIZE(ieer)

end
```

3

MPI/OpenMP併用並列処理 （3）

```
include "mpif.h"
integer, parameter :: n=10000
integer :: i,myid,ieer,nump,ir(0:100),id(0:100),it
double precision :: a(0:n-1),b(0:n-1)
  call MPI_INIT(ieer)
  call MPI_COMM_RANK(MPI_COMM_WORLD,myid,ieer)
  call MPI_COMM_SIZE(MPI_COMM_WORLD,nump,ieer)
  it=n/nump+1
!$OMP PARALLEL DO
  do i=it*(myid),min(n-1, it*(myid+1)-1)
    a(i)=float(I)
  enddo
!$OMP END PARALLEL DO
!$OMP PARALLEL DO
  do i=it*(myid),min(n-1, it*(myid+1)-1)
    b(i)=dsin(a(i))
  enddo
!$OMP END PARALLEL DO
```

```
do i=0,nump-1
   id(i)=it*i
enddo

do i=0,nump-2
   ir(i)=it
Enddo
ir(nump-1)=n - it*(nump-1)

call  MPI_ALLGATHERV(b(it*(myid)), ir(myid),
    MPI_DOUBLE_PRECISION, b(0),  ir(0), id(0),
    MPI_DOUBLE_PRECISION,
    MPI_COMM_WORLD, ieer)
call MPI_FINALIZE(ieer)

end
```

4

MPI/OpenMP併用並列処理 （4）

```
include "mpif.h"
integer, parameter :: n=10000
integer :: i,myid,ieer,nump,ir(0:100),id(0:100),it
double precision :: a(0:n-1),b(0:n-1)
  call MPI_INIT(ieer)
  call MPI_COMM_RANK(MPI_COMM_WORLD,myid,ieer)
  call MPI_COMM_SIZE(MPI_COMM_WORLD,nump,ieer)
  it=n/nump+1
!$OMP PARALLEL PRIVATE(i)
!$OMP  DO
  do i=it*(myid),min(n-1, it*(myid+1)-1)
    a(i)=float(I)
  enddo
!$OMP DO
  do i=it*(myid),min(n-1, it*(myid+1)-1)
    b(i)=dsin(a(i))
  enddo
```

```
!$OMP MASTER
do i=0,nump-1
   id(i)=it*i
enddo

do i=0,nump-2
   ir(i)=it
Enddo
ir(nump-1)=n - it*(nump-1)

call  MPI_ALLGATHERV(b(it*(myid)), ir(myid),
    MPI_DOUBLE_PRECISION, b(0),  ir(0), id(0),
    MPI_DOUBLE_PRECISION,
    MPI_COMM_WORLD, ieer)
!$OMP END MASTER
!$OMP END PARALLEL
call MPI_FINALIZE(ieer)

end
```

5