Partnership for Advanced Computing in Europe

Process Topologies in MPI
Process topologies

- MPI process topologies allow for simple referencing scheme of processes
  - Cartesian and graph topologies are supported
  - Process topology defines a new communicator
- MPI topologies are virtual
  - Not necessarily related to the physical structure of the computer
  - Process mapping more natural only to the programmer
- Usually no performance benefits
  - But code becomes more compact and readable
Creating a communication topology

- New communicator with processes ordered in a Cartesian grid

```c
MPI_Cart_create(oldcomm, ndims, dims, periods, reorder, newcomm)
```

- `oldcomm`communicator
- `ndims`dimension of the Cartesian topology
- `dims`integer array (size `ndims`) that defines the number of processes in each dimension
- `periods`array that defines the periodicity of each dimension
- `reorder`is MPI allowed to renumber the ranks
- `newcomm`new Cartesian communicator
Ranks and coordinates

- Translate a rank to coordinates

\[
\text{MPI\_Cart\_coords}(\text{comm}, \text{rank}, \text{maxdim}, \text{coords})
\]

- \text{comm} Cartesian communicator
- \text{rank} rank to convert
- \text{maxdim} dimension of \text{coords}
- \text{coords} coordinates in Cartesian topology that corresponds to \text{rank}
Ranks and coordinates

- Translate a set of coordinates to a rank
  \[
  \text{MPI\_Cart\_rank}(\text{comm}, \text{coords}, \text{rank})
  \]
  
  - \text{comm}: Cartesian communicator
  - \text{coords}: array of coordinates
  - \text{rank}: a rank corresponding to \text{coords}
Creating a communication topology

dims(1)=4

dims(2)=4

period=(/ .true. , .true. /)

call mpi_cart_create(mpi_comm_world, 2,&
   dims, period, .true., comm2d, rc)
call mpi_comm_rank(comm2d, my_id, rc)
call mpi_cart_coords(comm2d, my_id, 2,&
   coords, rc)
Communication in a topology

- Counting sources/destinations on the grid
  - for e.g. elegant nearest-neighbor communication

\[
\text{MPI\_Cart\_shift}(\text{comm}, \text{direction}, \text{displ}, \text{source}, \text{dest})
\]

- \text{comm}: Cartesian communicator
- \text{direction}: shift direction (e.g. 0 or 1 in 2D)
- \text{displ}: shift displacement
- \text{source}: rank of source process
- \text{dest}: rank of destination process

Note that both source and dest are output parameters. The coordinates of the calling task is implicit input.

With a non-periodic grid, source or dest can land outside of the grid; then MPI\_PROC\_NULL is returned.
Halo exchange

dims(1)=4
dims(2)=4
period =(/ .true. , .true. /)

call mpi_cart_create(mpi_comm_world, 2,&
        dims, period, .true., comm2d, rc)
call mpi_cart_shift(comm2d,0,1,nbr_up,nbr_down,rc)
call mpi_cart_shift(comm2d,1,1,nbr_left,nbr_right,rc)
...
call mpi_sendrecv(hor_send, msglen, mpi_double_precision, nbr_left,&
        tag_left, hor_recv, msglen, mpi_double_precision, nbr_right,&
        tag_left, comm2d, mpi_status_ignore, rc)
...
call mpi_sendrecv(vert_send, msglen, mpi_double_precision, nbr_up,&
        tag_up, vert_recv, msglen, mpi_double_precision, nbr_down,&
        tag_up, comm2d, mpi_status_ignore, rc)
...