

COSC 6374 Parallel Computation

Introduction to MPI (III) - Process Grouping

Edgar Gabriel
Fall 2010



Edgar Gabriel



Terminology (I)


- an `MPI_Group` is the object describing the list of processes forming a logical entity
 - a group has a size `MPI_Group_size`
 - every process in the group has a unique rank between 0 and (size of group - 1) `MPI_Group_rank`
 - a group is a local object, and cannot be used for any communication



Parallel Computation
Edgar Gabriel



Terminology (II)

- An `MPI_Comm`(unicator) is an object containing
 - one or two groups of processes (*intra* or *inter*-communicators)
 - topology information
 - attributes
- A communicator has an error handler attached to it
- A communicator can have a name 
- these slides focus on *intra*-communicators i.e. the list of participating processes can be described by a single group



Predefined communicators

- `MPI_COMM_WORLD`
 - contains all processes started with `mpirun/mpiexec`
 - exist upon exiting `MPI_Init`
 - can not be modified, freed etc.
- `MPI_COMM_SELF`
 - contains just the local process itself, size is always 1
 - exist upon exiting `MPI_Init`
 - can not be modified, freed etc.



Creating new communicators

- All communicators in MPI-1 are derived from `MPI_COMM_WORLD` or `MPI_COMM_SELF`
- Creating and freeing a communicator is a collective operation → all processes of the original communicator have to call the function with the same arguments
- Methods to create new communicators
 - splitting the original communicator into n-parts
 - creating subgroups of the original communicator
 - re-ordering of processes based on topology information

 - spawn new processes
 - connect two applications and merge their communicators



Splitting a communicator

```
MPI_Comm_split ( MPI_Comm comm, int color,
                int key, MPI_comm *newcomm);
```

- Partition `comm` into sub-communicators
 - all processes having the same `color` will be in the same subcommunicator
 - order processes with the same `color` according to the `key` value
 - if the `key` value is identical on all processes with the same `color`, the same order for the processes will be used as in `comm`.

