ICHEC
Irish Centre for High-End Computing

PRACE Course: Intermediate MPI

9-11 November 2022

Inter-Communicators

Introduction

If a communicator contains ranks that are of the same group, it is called a intra-communicator.

The most obvious intra-communicator is MPI_COMM_WORLD.

An inter-communicator is one that communicates between ranks that are from different groups.

The two groups should be disjoint, that is the union is empty.

Point-to-point and collective operations can be performed.

Inter-communicators cannot be used with topologies.

A communicator can be intra or inter but not both.

MPI_Intercomm_create

To setup an inter-communicator you need to call this function.

- All ranks in both groups must call it
  - `MPI_Intercomm_create(local_comm,local_leader,peer_comm,`
  - `remote_leader,tag,*intercomm)`

Each group will have a `local_leader` rank.

All ranks from the same group must specify the same `local_leader`. The leader from the other group does not need to be the same.

The `remote_leader` will be the rank of the other group's leader.

This rank will be associated with the `peer_comm`.

Thus `peer_comm` must be an intra-communicator for the union.

Example

Three groups A_(0,1), B_(2,3) and C_(4,5), intercom A->B and A->C

```
int color,rank
MPI_Comm Inter, Inter1, Local;

MPI_Comm_split(MPI_COMM_WORLD,color,0,&Local);

color = rank/2;
if (color==0) {
   MPI_Intercomm_create(Local,0,MPI_COMM_WORLD,2,0,&Inter);
   MPI_Intercomm_create(Local,0,MPI_COMM_WORLD,4,0,&Inter1);
} else {
   MPI_Intercomm_create(Local,0,MPI_COMM_WORLD,0,0,&Inter);
}
```

Destroying Communicators

Destroying inter-communicators is the same process as destroying intra-communicators.

Use the function below to do this.

It is always good practice to destroy any custom communicators.

- All ranks in both groups must call it

```
    MPI_Comm_free(*intercomm)
```

Messages

Point-to-point messages can be sent from any member of the local group to any other in the remote group.

It is not just restricted to the leaders.

However the destination and source are the remote group's rank.

Collective communications behave slightly differently when used with an inter-communicator.

We will use some of these to illustrate the differences.

Not all collectives can be used with intercommunicators.

P2P Example

Three groups A_(0,1), B_(2,3) and C_(4,5), intercom A->B and A->C

```
int color,global_rank,local_rank,buf;
MPI_Comm Inter, Inter1, Local;
MPI_Status stat;

MPI_Comm_rank(MPI_COMM_WORLD,&global_rank);
if (color == 0) {
// Assume that ranks are not reordered
   local_rank = global_rank;
} else {
   MPI_Comm_rank(Inter,&local_rank);
}
buf = global_rank;
// Message from C(1)->A(0)
If (color==0 & local_rank==0) MPI_Recv(&buf,1,MPI_INT,1,0,Inter1,&stat);
if (color==2 & local_rank==1) MPI_Send(&buf,1,MPI_INT,0,0,Inter);
```

MPI_Bcast

Broadcast is a rooted type of collective like gather and scatter.

Only one rank in one group is involved (rooted group) and all ranks in the other (remote).

The message is broadcast from any rank within the rooted group to all the ranks in the remote group.

The arguments are the same as before and called by all ranks in the inter-communicator.

```
MPI_Bcast(*buffer,count,type,source,Inter)
```

However in the rooted group the `source` rank must have MPI_ROOT.

All other ranks in the rooted group must have MPI_PROC_NULL.

All ranks in the remote group must have the `source` rank relative to the rooted group.

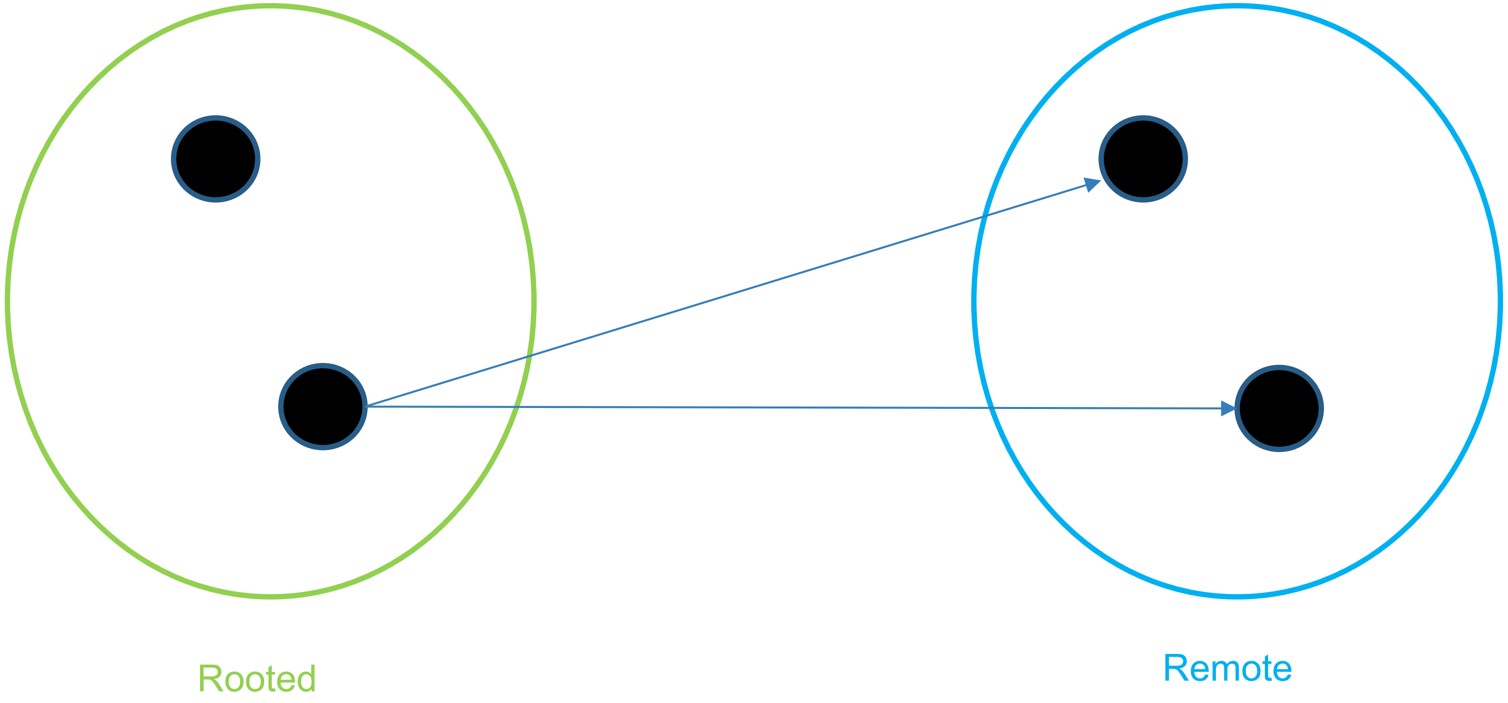
Example

Two groups Rooted_(0,1), Remote_(2,3) with intercomm.

```
int rooted,source,local_rank,root,buf;

source = 1;
if (rooted) {
  if (local_rank == source) {
     root = MPI_ROOT;
  } else {
     root = MPI_PROC_NULL;
  }
} else {
   root = source;
}
MPI_Bcast(&buf, 1, MPI_INT, root, Inter);
```

Diagram of Broadcast



MPI_Allgather

This is a all-to-all type of communication where all ranks in both groups are involved.

Other functions of this type are Allreduce and Alltoall.

Normally in Allgather a message is received from each rank which is then concatenated and broadcast.

Over an inter-communicator the local group receives messages from the remote group.

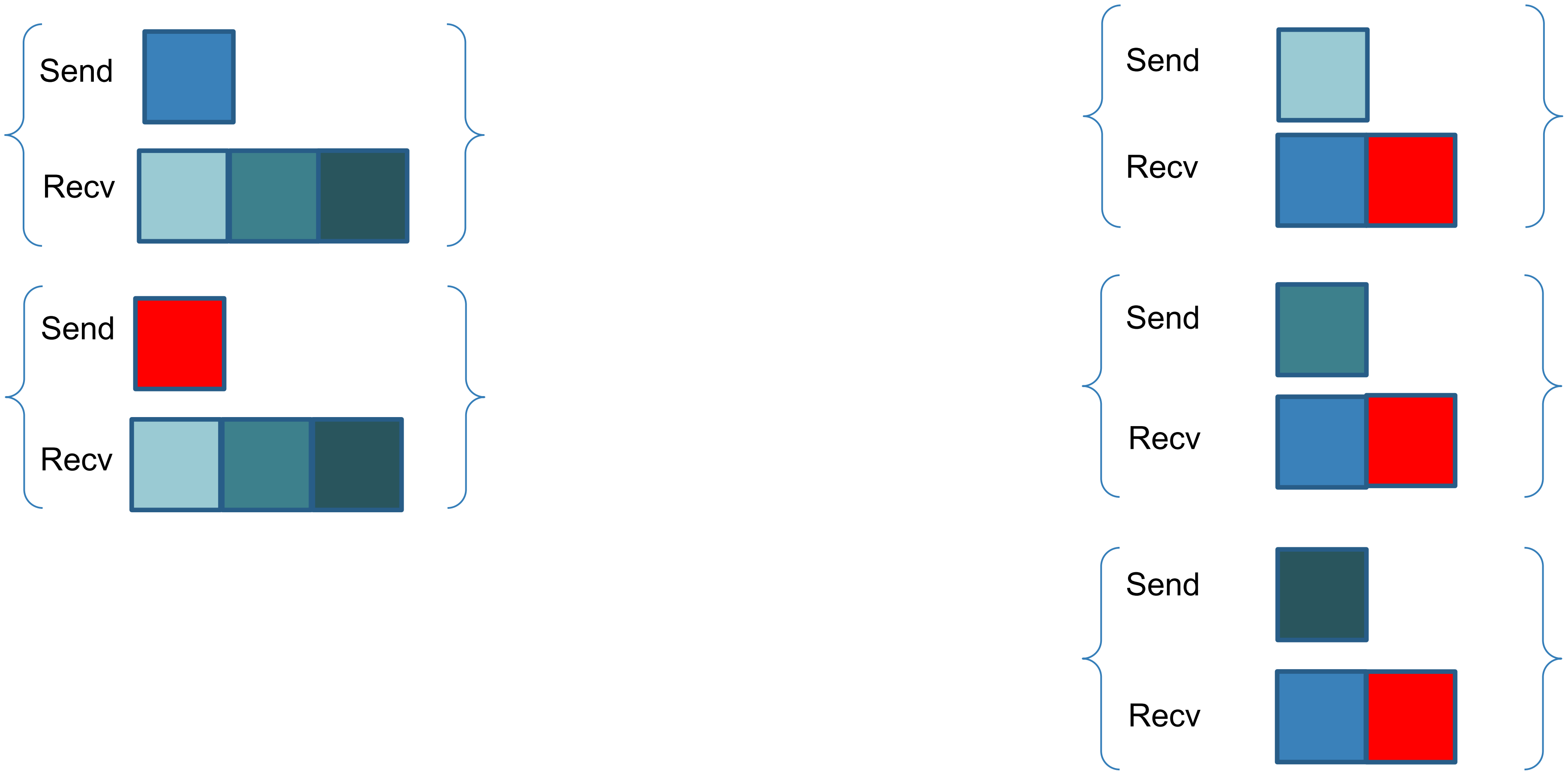
These messages are concatenated and broadcast locally.

The same is true for the remote group.

The arguments are the same as before and called by all ranks in the inter-communicator.

```
   MPI_Allgather(*sendbuf,send_count,send_type,*recv_buf,
recv_count,recv_type,Inter)
```

Diagram of Allgather



Send

Recv

Send

Recv

Send

Recv

Send

Recv

Send

Recv

MPI_Allreduce

Like Allgather the reduction from the local group is passed to the remote group.

It is very much like Allgather but with a defined operation applied to the gather data.

Again all ranks within both the local and remote groups must execute the call.

The arguments are the same as with Intracommunicator.

```
MPI_Allreduce(*sendbuf,*recvbuf,send_count,
send_type,operation,Inter)
```

Other Functions

The two groups that are associated with an inter-communicator can be merged and connected with an intra-comminucator.

```
    MPI_Intercomm_merge(Inter,high,Intra)
```

The `high` argument determines the order of the ranks within the intra-communicator.

You can test to see if the communicator is an intra or inter comm.

```
    MPI_Comm_test_inter(Comm,*flag)
```

The `flag` argument which is an int in C, returns 0 if an intra-communicator.

Finally you can get the size of the remote group.

```
    MPI_Comm_remote_size(Inter,*size)
```

Examples

There are two examples using collectives with inter-communicators.

Move to Kay.