Hello World

01-01: Write a basic Hello world program

01-02: Write an MPI hello world program
- Include different print statements before MPI_Init and after MPI_Finalize. Explain the output

02 : Obtain MPI rank and size of communicator. Print.

03 : Write a program where only process 0 prints the total number of processes.

MPI_Send, MPI_Recv

Use 2 ranks. Initialize sendbuf and recvval.
Send from 0 and receive in rank 1. Print.

```
MPI_Send(&sendbuf,sendcount,MPI_DATATYPE,
                        dest,tag,MPI_COMM_WORLD);

MPI_Recv(&recvval,recvcount,MPI_DATATYPE,

source,tag,MPI_COMM_WORLD, &status);
```

MPI_Sendrecv

Take 10 ranks.

Initialize **sendbuf** and **recvbuf** to different values. Print them.

Circulate **sendbuf** from **rank 0** through ranks till **9.** Print rank and received value

MPI_Sendrecv

Sends and receives a message

Synopsis

```
int MPI_Sendrecv(*sendbuf, sendcount, MPI_Datatype sendtype,
                 dest, sendtag,
                 *recvbuf, recvcount, MPI_Datatype recvtype,
                 source, recvtag,
                 MPI_Comm , MPI_Status *status)
```

•

MPI Collectives

Broadcast

initialize sendbuf to 33 for rank 0. sendbuf=66 on other ranks.

Broadcast sendbuf from rank 0 to all ranks.

```
MPI_Bcast(&sendbuf,count,MPI_Datatype,
                    root,MPI_COMM_WORLD);
```

MPI_Reduce

Obtain the total number of ranks, by other than MPI_Comm_size

MPI_Reduce

Let each process compute a random number. You want to print the maximum of these numbers to your screen.

```
#include <time.h>
    srand(time(NULL));
    int r = rand();
```

Each process computes a random number again. Now you want to scale these numbers by their maximum.

Let each process compute a random number. You want to print on what processor the maximum value is computed.

`MPI_Reduce(sendbuf,recvbuf,count,datatype,op,root,comm)`

MPI_SUM, MPI_MIN, MPI_MAX, MPI_PROD, logical AND, OR

MPI_Allreduce

- Normalizing a vector
- y = x/||x||
- Vectors x,y are distributed: every process has certain elements
- The norm calculation is an all-reduce: every process gets same value
- Every process scales its part of the vector.