Lesson 5.8

Writing Hello World in MPI

Learning Objectives

- Students will write their first MPI program
- Students will become familiar with what is a rank

Your First MPI Program (in Python)

Your code will go to every process so it must tell each process what it should do!

As a result the program needs to differentiate each process

Telling Each Process Who They Are

Open a new text editor file (saved as ‘Your_Program.py’) and type:

```
from mpi4py import MPI                          # Imports the MPI Framework

comm = MPI.COMM_WORLD                           # Saves information about all processes

rank = comm.Get_rank()                          # Tells this process what rank it is

print(‘My rank is {}’.format(rank))             # Prints to the screen a process’ rank
```

Each process is numbered by its rank

Rank 0

Rank 1

Rank 2

Differentiating between ranks

One way for certain ranks to do different jobs is to set up if/else statements. Adding to the code from before:

```
if rank == 0:
        print(“Hello World from rank {}”.format(rank))
else:
        print(“I am rank {}”.format(rank))
```

Feel free to change which rank you select to say ‘Hello World’

What each process does from the example code

Rank 0

“Hello World from rank 0”

Rank 1

“I am rank 1”

Rank 2

“I am rank 2”

A note on the order things will be printed in

When running the program, the order each rank reports back will be random:

Sample output:

“I am rank 1”

“Hello World from rank 0”

“I am rank 2”

Each process operates independently and will have different timings. This random execution leads to something called a Race Condition. To avoid this, when the order of operations is critical, code must be written to follow a proper order

Conclusion

MPI programs differentiate different processes by assigning them a rank

In your MPI program, the coder must specify which rank does what.