HPC-14-2

February 14, 2024

1 Assignment 8

1. Write a program to pass message from one process to another and print output.
   - In synchronous communication
   - In asynchronous communication. Show using overlapping of task in non-blocking mode.

[1]:

```
import mpi4py
from mpi4py import MPI
```

[2]:

```
import numpy as np
```

[3]:

```
comm = MPI.COMM_WORLD # get the communicator object
rank = comm.Get_rank() # get the rank of the current process
name = MPI.Get_processor_name() # get the name of the current processor
size = comm.Get_size() # get the number of processes
```

[4]:

```
randNum = np.zeros(1)
```

[ ]:

```
if rank == 0:
    message = "Hello from process 0"
    comm.send(message, dest=1)

    received_message = comm.recv(source=1)
    print(f"Process 0 received message: {received_message}")

elif rank == 1:
    received_message = comm.recv(source=0)
    print(f"Process 1 received message: {received_message}")

    reply = "Hello from process 1"
    comm.send(reply, dest=0)
```

[5]:

```
!mpiexec -n 2 python hpc-12-2.py
```

```
Process 1 received message: Hello from process 0
Process 0 received message: Hello from process 1
```

[ ]:

```python
if rank == 0:
    message = "Hello from process 0 (Async)"
    req_send = comm.isend(message, dest=1)
    print(f"Process {rank} sent message: {message}")
    time.sleep(1)
    req_send.wait()
elif rank == 1:
    req_recv = comm.irecv(source=0)
    time.sleep(0.5)
    print(f"Process {rank} waiting to receive message...")
    received_message = req_recv.wait()
    print(f"Process {rank} received message: {received_message}")
```

[8]:

```
!mpiexec -n 2 python hpc-async.py
```

```
Process 0 sent message: Hello from process 0 (Async)
Process 1 waiting to receive message…
Process 1 received message: Hello from process 0 (Async)
```