Message Passing Interface
An Overview

Joseph Kehoe[1]

[1]Department of Computing and Networking
Institute of Technology Carlow

CDD101, 2018

INSTITIúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Messaging

- Alternative to Procedure Calls (RPC)
- More Flexibility
- better suited to distributed systems
- Asynchronous communication

Introduction to MPI

Message Passing Interface (MPI)

- Portable Message Passing Programs
- Fortran,C,C++,Python, etc.
- Designed for Ease of Use
- Thread Safe
- MPI has been standardized

Features of MPI

Message Passing Interface (MPI)

- Point to point communications
- Collective Operations
- Process groups
- Process Topologies
- Language bindings

Tutorial Code

- We will follow the MPI tutorial here
- Download the tutorial and compile the code (I assume you will be using unix)

INSTITUTE of
TECHNOLOGY
CARLOW
At the Heart of South Leinster

Hello World

- Initiialise MPI with:
- `MPI_Init(NULL, NULL);`
- Get number of processes with:
- `MPI_Comm_size(MPI_COMM_WORLD, &world_size);`
- Get our rank with:
- `MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);`
- End program with:
- `MPI_Finalize();`

Sending and Receiving

```
int number;
if (world_rank == 0) {
    number = -1;
    MPI_Send(&number, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
} else if (world_rank == 1) {
    MPI_Recv(&number, 1, MPI_INT, 0, 0,
             MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    printf("Process 1 received  %d from process 0\n", number);
}
```

Message Parameters of Send

- Address of start of data being sent
- Number of items of data being sent
- Type of data
- Rank of destination
- Message Tag (integer)
- Communicator (handle)

Message Parameters of Receive

- Address of start of receiving buffer
- Number of items of data being received
- Type of data
- Rank of source
- Message Tag (integer)
- Communicator (handle)
- A status object

MPI Primitives

MPI_bsend Append outgoing message to local send buffer

MPI_send Send a message and wait until copied to local or remote
          buffer

MPI_ssend Send a message and wait until receipt starts

MPI_sendrecv Send a message and wait for reply

MPI Primitives

MPI_isend Pass reference to outgoing message, and continue

MPI_issend Pass refence to outgoing message , and wait until receipt starts

MPI_recv Receive a message, block if there is none

MPI_irecv Check if there is an incoming message, but do not block

see here

INSTITUTE of
TECHNOLOGY
CARLOW
At the Heart of South Leinster