MPI Tutorial

Kushagra and Dhruv

What is MPI(Message Passing Interface)?

❌ A compiler

❌ A library

❌ A framework

❌ A programming language

✅ is a standard/document that defines an interface for message-passing libraries

Message Passing in Parallel Computing

- MPI is for communication among processes, which have separate address spaces.
- MPI provides a powerful, efficient, and portable way to express parallel programs

Why do we need MPI?

- Lots of interfaces and therefore lots of libraries
- Developers: Portability issues
- Hardware Vendors: Optimization

Communicator

- A communicator defines a group of processes that have the ability to communicate with one another.
- In this group of processes, each is assigned a unique rank, and they explicitly communicate with one another by their ranks.
- There is a default communicator whose group contains all initial processes, called MPI COMM WORLD.

Point to Point Communication

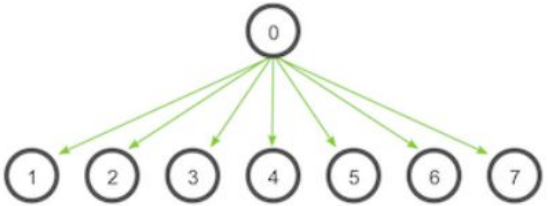
Send & Receive

```
MPI_Send(
    void* data,
    int count,
    MPI_Datatype datatype,
    int destination,
    int tag,
    MPI_Comm communicator)
```

```
MPI_Recv(
    void* data,
    int count,
    MPI_Datatype datatype,
    int source,
    int tag,
    MPI_Comm communicator,
    MPI_Status* status)
```

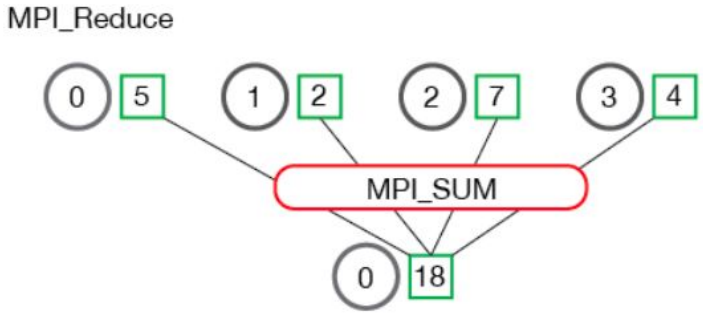
Collective Communication

Broadcast



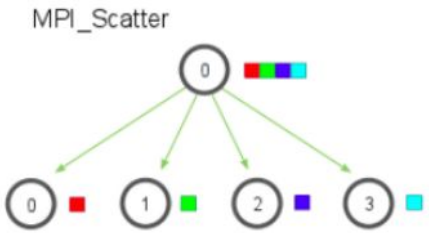

```
MPI_Bcast(
    void* data,
    int count,
    MPI_Datatype datatype,
    int root,
    MPI_Comm communicator)
```

Reduce



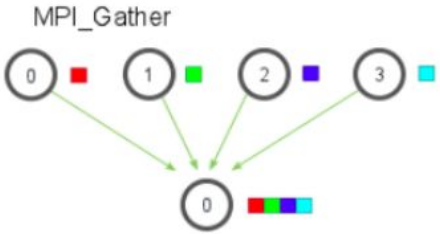

```
MPI_Reduce(
    void* send_data,
    void* recv_data,
    int count,
    MPI_Datatype datatype,
    MPI_Op op,
    int root,
    MPI_Comm communicator)
```

Scatter




```
MPI_Scatter(
    void* send_data,
    int send_count,
    MPI_Datatype send_datatype,
    void* recv_data,
    int recv_count,
    MPI_Datatype recv_datatype,
    int root,
    MPI_Comm communicator)
```

Gather




```
MPI_Gather(
    void* send_data,
    int send_count,
    MPI_Datatype send_datatype,
    void* recv_data,
    int recv_count,
    MPI_Datatype recv_datatype,
    int root,
    MPI_Comm communicator)
```

Additional Commands

- Refer to documentation

Thanks!