

# CSCI 251: Concepts of Parallel and Distributed Systems

Alvin Lin

October 2nd, 2017

## Message Passing Systems

MPI is an industry standard interface for message passing systems. It is supported by almost every high performance computer (HPC). It is portable and high performance. There are 7430 MPI routines, but for most programs, we only need 10-12 routines. Refer to the documentation at:

<https://www.mpich.org/static/docs/v3.1/>

## Basic Program Structure

```
MPI include file
  Declarations, prototypes, etc.
Program begins
  Serial code
Initialize environment - parallel code begins
MPI_Init
  Parallel processing code
  Message Passing
MPI_Terminate - parallel code ends
Serial code
```

## MPI Communicator

Groups of processes need to communicate with one another and a communicating processor may be part of one or more communication groups. `MPI_Comm_World` is the collection of all processes that may communicate with each other in the parallel execution. The number of processes in an MPI communicator is given by 'size', and

each process in a communicator has a 'rank', which is an integer from 0 to (size - 1). Rank serves as a unique ID for each process.

## **MPI Sending and Receiving**

MPI\_Send performs a blocking send:

[https://www.mpich.org/static/docs/v3.1/www3/MPI\\_Send.html](https://www.mpich.org/static/docs/v3.1/www3/MPI_Send.html)

MPI\_Recv performs a blocking receive:

[https://www.mpich.org/static/docs/v3.1/www3/MPI\\_Recv.html](https://www.mpich.org/static/docs/v3.1/www3/MPI_Recv.html)

## **Synchronization**

All member processes wait until they have all reached a synchronization point. For MPI, this is done with MPI\_Barrier:

[https://www.mpich.org/static/docs/v3.1/www3/MPI\\_Barrier.html](https://www.mpich.org/static/docs/v3.1/www3/MPI_Barrier.html)

## **Data Transfer**

MPI allows a specific process to broadcast a message to all other messages in the communicator group via MPI\_Bcast:

[https://www.mpich.org/static/docs/v3.1/www3/MPI\\_Bcast.html](https://www.mpich.org/static/docs/v3.1/www3/MPI_Bcast.html)

MPI\_Scatter allows an external process to send data to all processes in a communicator group:

[https://www.mpich.org/static/docs/v3.1/www3/MPI\\_Scatter.html](https://www.mpich.org/static/docs/v3.1/www3/MPI_Scatter.html)

## **Data Reduction**

MPI\_Gather gathers together values from a group of processes.

[https://www.mpich.org/static/docs/v3.1/www3/MPI\\_Gather.html](https://www.mpich.org/static/docs/v3.1/www3/MPI_Gather.html)

MPI\_Reduce allows values from processes to be reduced to a single value through operations such as max, min, sum, etc:

[https://www.mpich.org/static/docs/v3.1/www3/MPI\\_Reduce.html](https://www.mpich.org/static/docs/v3.1/www3/MPI_Reduce.html)

## **Blocking vs Nonblocking**

All the MPI functions have a non-blocking counterpart of the form MPI\_I\*. While blocking is safer, non-blocking allows for faster overlapped computation, but may result in unsafe operations.

## Reminders

The midterm is on October 11th. Refer to MyCourses for details on Project 1, which is due Friday, October 6th.

Professor Mohan Kumar:  
[mjkvcs@rit.edu](mailto:mjkvcs@rit.edu)  
<https://cs.rit.edu/~mjk>

Rahul Dashora (TA):  
[rd5476@mail.rit.edu](mailto:rd5476@mail.rit.edu)

You can find all my notes at <http://omgimanerd.tech/notes>. If you have any questions, comments, or concerns, please contact me at [alvin@omgimanerd.tech](mailto:alvin@omgimanerd.tech)