MPI, Again

Last time: MPI overview

- In Java, sort of

This time: more examples

- In C
- Running code!

Using MPI on CADE Machines

Compile:

`~cs4960-01/mpi/bin/mpicc` *src-file*

Run:

`~cs4960-01/mpi/bin/mpirun -np` *P exe-file*

Run single process:

add **`/home/cs4960-01/mpi/lib`**
to **`LD_LIBRARY_PATH`**

Docs:

add **`/home/cs4960-01/mpi/share/man`**
to **`MANPATH`**

Hello, MPI

See **`hello.c`**

MPI Messages

See **msg.c**

```
int
MPI_Send(void *buf,      // array of data
         int count,      // array length
         MPI_Datatype datatype,  // array element type
         int dest,       // send to this rank
         int tag,        // your choice
         MPI_Comm comm)  // use MPI_COMM_WORLD

int
MPI_Recv(void *buf,      // array of data
         int count,      // array length
         MPI_Datatype datatype, // array element type
         int source,     // recv from this rank
         int tag,        // your choice
         MPI_Comm comm,  // use MPI_COMM_WORLD
         MPI_Status *status) // info on msg
```

MPI Non-Blocking Messages

See **imsg.c**

```
int
MPI_Isend(void *buf,      // array of data
          int count,      // array length
          MPI_Datatype datatype,  // array element type
          int dest,       // send to this rank
          int tag,        // your choice
          MPI_Comm comm,  // use MPI_COMM_WORLD
          MPI_Request *request) // remembers

int
MPI_Irecv(void *buf,      // array of data
          int count,      // array length
          MPI_Datatype datatype, // array element type
          int source,     // recv from this rank
          int tag,        // your choice
          MPI_Comm comm,  // use MPI_COMM_WORLD
          MPI_Request *request) // remembers
```

MPI Non-Blocking Finishing

See **imsg.c**

```
int
MPI_Wait(MPI_Request *request, // remembered
         MPI_Status *status)   // info on msg
```

See **imsgs.c**, **imsgs2.c**

```
int
MPI_Waitany(int count,          // array length
            MPI_Request *request, // array of remembered
            int *index,         // which was ready
            MPI_Status *status) // info on msg

MPI_Request MPI_REQUEST_NULL; // finished request
int MPI_UNDEFINED;            // finished "ready" result
```