

MPI (Basic)

Setup and Tear Down

<code>void main(int argc, char *argv)</code>	starts up the MPI runtime environment at the beginning of a run.
<code>MPI_Finalize()</code>	shuts down the MPI runtime environment at the end of a run.

Gathering Information

<code>MPI_Comm_rank(MPI_COMM_WORLD, &myid)</code>	gets the process ID that the current process uses, which is between 0 and $N_p - 1$ inclusive.
<code>MPI_Comm_size(MPI_COMM_WORLD, &numprocs)</code>	gets the number of processes in a run.

MPI Data Types

<code>MPI_CHAR</code>	char
<code>MPI_SHORT</code>	short int
<code>MPI_INT</code>	int
<code>MPI_FLOAT</code>	float
<code>MPI_DOUBLE</code>	double
<code>MPI_LONG_DOUBLE</code>	long double
<code>MPI_BYTE</code>	consists of a byte (8 binary digits)

Compile

<code>mpicc -o file file.c</code>	compiles MPI programs written in C.
<code>mpic++ -o file file.cpp</code>	compiles MPI programs written in C++.

Run

<code>mpirun -np no_processors file</code>	run MPI compiled file with no_processors
--	--

Time

<code>MPI_Wtime()</code>	Returns an elapsed time on the calling processor
--------------------------	--

`int MPI_Bcast (*b, c,d, root,MPI_Comm)`

<code>b</code>	The message to be broadcasted
<code>c</code>	Number of elements in the message
<code>d</code>	The data type of the elements in the message
<code>root</code>	The process number that has the message to be broadcasted to others
<code>MPI_Comm</code>	The communication world

`MPI_Recv(*b, c, d, sender, t, MPI_Comm, status)`

<code>b</code>	Receive in buffer b
<code>c</code>	The number of element of data type d
<code>d</code>	The data type of element b
<code>sender</code>	The rank of the sender
<code>t</code>	The tag used in the communication
<code>MPI_Comm</code>	The communication world
<code>status</code>	the status of the reception operation

`int MPI_Scatter(sb,sc,sd,rb,rc,rd,root,MPI_Comm)`

<code>sb</code>	The buffer containing the data to disptach from the root process.
<code>sc</code>	The number of elements to send to each process, not the total number of elements in the send buffer.
<code>sd</code>	The type of one send buffer element
<code>rb</code>	The buffer in which store the data dispatched.
<code>rc</code>	The number of elements in the receive buffer
<code>rd</code>	The type of one receive buffer element.
<code>root</code>	The rank of the root process
<code>MPI_Comm</code>	The communication world

Libraries

<code><mpi.h></code>	For MPI implementation
<code><stdio.h></code>	C input&output
<code><math.h></code>	Handles Math



Terms

Blocking	return after their actions complete
Non-Blocking	return immediately.
Message Tag	carried within message and used to differentiate between different types of messages being sent
MPI_Status	represents the status of a reception operation.

MPI Point-to-Point Communication

MPI_Send()	sends a message from the current process to some other process.
MPI_Recv()	receives a message on the current process from some other process.

Collective Communication

MPI_Bcast()	Broadcast from root to all other processes
MPI_Gather()	Gather values for group of processes
MPI_Scatter()	Scatters buffer in parts to group of processes
MPI_Alltoall()	Sends data from all processes to all processes
MPI_Reduce()	Combine values on all processes to single value
MPI_Reduce_scatter()	Combine values and scatter results

MPI Nonblocking Routines

MPI_Isend()	Non-blocking send; will return "immediately"
MPI_Irecv()	Nonblocking receive; will return even if no message to accept.
MPI_Wait(request,status)	waits until operation completed and returns then.
MPI_Test()	returns with flag set indicating whether operation completed at that time.

MPI (Detailed)

MPI_Send(*d, c, m, receiver, t, MPI_Comm)

d	Send Data from address d
c	The number of elements of d
m	The Datatype of d
receiver	The rank of the receiver
t	The communication is marked with t tag
MPI_Comm_World	The communication world

int MPI_Gather(sb,sc,sd,rb,rc,rd,root,MPI_Comm)

sb	The sender buffer
sc	The number of elements in the buffer
sd	The datatype of one element in the buffer
rb	The buffer in which store the gathered data for the root process
rc	The number of elements per message received, not the total number of elements to receive from all processes altogether.
rd	The type of one receive buffer element.
root	
MPI_Comm	The communication world

int MPI_Reduce(sb,rb,c,d,op,root,MPI_Comm);

sb	A pointer on the buffer to send for reduction.
rb	A pointer on the buffer in which store the result of the reduction
c	The number of elements in the send buffer
d	The type of a buffer element.
op	The operation to apply to combine messages received in the reduction
root	The rank of the MPI process that will collect the reduction result.
MPI_Comm	The communication world

