

6 Where To Go From Here

6.1 What We Haven't Discussed

MPI is a large library. The *Standard* [4] is over 200 pages long and it defines more than 125 functions. As a consequence, this *Guide* has covered only a small fraction of MPI, and many readers will fail to find a discussion of functions that they would find very useful in their applications. So we briefly list some of the more important ideas in MPI that we have not discussed here.

1. **Communication Modes.** We have used only the *standard* communication mode for `send`. This means that it is up to the system to decide whether the message is buffered. MPI provides three other communication modes: *buffered*, *synchronous*, and *ready*. In buffered mode, the user explicitly controls the buffering of outgoing messages. In synchronous mode, a send will not complete until a matching receive is posted. In ready mode, a send may be started only if a matching receive has already been posted. MPI provides three additional send functions for these modes.
2. **Nonblocking Communication.** We have used only *blocking* sends and receives (`MPI_Send` and `MPI_Recv`.) For the send, this means that the call won't return until the message data and envelope have been buffered or sent — i.e., until the memory referenced in the call to `MPI_Send` is available for re-use. For the receive, this means that the call won't return until the data has been received into the memory referenced in the call to `MPI_Recv`. Many applications can improve their performance by using *nonblocking* communication. This means that the calls to send/receive may return before the operation completes. For example, if the machine has a separate communication processor, a non-blocking send could simply notify the communication processor that it should begin composing and sending the message. MPI provides nonblocking sends in each of the four modes and a nonblocking receive. It also provides various utility functions for determining the completion status of a non-blocking operation.
3. **Inter-communicators.** Recollect that MPI provides two types of communicators: intra-communicators and inter-communicators. Inter-

communicators can be used for point-to-point communications between processes belonging to distinct intra-communicators.

There are many other functions available to users of MPI. If we haven't discussed a facility you need, please consult the *Standard* [4] to determine whether it is part of MPI.

6.2 Implementations of MPI

If you don't have an implementation of MPI, there are three versions that are freely available by anonymous ftp from the following sites.

- Argonne National Lab/Mississippi State University. The address is `info.mcs.anl.gov`, and the directory is `pub/mpi`.
- Edinburgh University. The address is `ftp.epcc.ed.ac.uk`, and the directory is `pub/chimp/release`.
- Ohio Supercomputer Center. The address is `tbag.osc.edu`, and the directory is `pub/lam`.

All of these run on networks of UNIX workstations. The Argonne/Mississippi State and Edinburgh versions also run on various parallel processors. Check the "README" files to see if your machine(s) are supported.

6.3 More Information on MPI

There is an MPI FAQ available by anonymous ftp at

- Mississippi State University. The address is `ftp.erc.msstate.edu`, and the file is `pub/mpi/faq`.

There are also numerous web pages devoted to MPI. A few of these are

- <http://www.epm.ornl.gov/~walker/mpi>. The Oak Ridge National Lab MPI web page.
- <http://www.erc.msstate.edu/mpi>. The Mississippi State MPI web page.
- <http://www.mcs.anl.gov/mpi>. The Argonne MPI web page.

Each of these sites contains a wealth of information about MPI. Of particular note, the Mississippi State page contains a bibliography of papers on MPI, and the Argonne page contains a collection of test MPI programs.

The *MPI Standard* [4] is currently available from each of the sites above. This is, of course, the definitive statement of what MPI is. So if you're not clear on something, this is the final arbiter. It also contains a large number of nice examples of uses of the various MPI functions. So it is considerably more than just a reference. Currently, several members of the MPI Forum are working on an annotated version of the MPI standard [5].

The book [2] is a tutorial introduction to MPI. It provides numerous complete examples of MPI programs.

The book [6] contains a tutorial introduction to MPI (on which this guide is based). It also contains a more general introduction to parallel processing and the programming of message-passing machines.

The Usenet newsgroup, `comp.parallel.mpi`, provides information on updates to all of these documents and software.

6.4 The Future of MPI

As it is currently defined, MPI fails to specify two critical concepts: I/O and the creation/destruction of processes. Work has already been started on the development of both I/O facilities and dynamic process creation. Information on the former can be obtained from <http://lovelace.nas.nasa.gov/MPI-IO/mpi-io.html>, and information on the latter can be found on the Argonne MPI web page. Significant developments are invariably posted to `comp.parallel.mpi`.