Minutes of MPI meeting
January 24–26
Vienna, Austria

William Gropp

March 8, 1996

1 Introduction

This is an unedited set of minutes taken during this MPI meeting. This contains both a summary of some of the discussions and official, binding votes of the MPI Forum. Offical votes are set in bold type; for example **1-2-3**. Straw votes, which are non-binding and may be voted on by anyone present, are shown as *1-2-3*. Please send any corrections to `gropp@mcs.anl.gov`.

2 Wednesday

Rusty Lusk opens meeting with a discussion of the agenda, along with a schedule for the readings and formal votes by chapter.

3 Dynamic Processes

We began with a discussion of the dynamic process section.

A discussion of `MPI_UNIVERSE_SIZE`, a way to find the number of available processors, resulted in much discussion about what it exactly it means. A straw vote was taken to decide whether to pursue; it passed *15,6,3*.

Lyndon Clarke suggested replacing the `MPI_SIGNAL` and `MPI_SIGNAL_GROUP` routines with the more specific `MPI_KILL` and `MPI_KILL_GROUP`. *10,7,8*

4 One-sided communication

We then discussed the one-sided operations. There were a number of straw votes taken; these are sumarized here.

One question that came up for remote memory operations is exactly how data areas that to be used in remote memory operations are defined. The choices were

- Both INIT and MALLOC required *11*
- MALLOC *6*
- INIT *2*
- abstain *7*

Much discussion of `sendtype`, `recvtype`, `windowtype` (question on how `recvtype` and `windowtype` interact). Also, how much of this is caused *only* by supporting heterogeneous systems.

Also raised is the issue of when data is register/cache/memory consistent? In C, `volatile` can often be used. In Fortran, there seems to be no strictly language-conforming solution.

After lunch, continued discussion of RMA regions. Should the `RMA_INIT` region be a communicator (*11*) or a new object (*3*). Many abstentions.

Moved on to counters on `PUT`:

- Remove counters on put *0*
- Some sort of counter *16*
- One counter per window *12*
- Many counters per window *3*

5 External interfaces

Much debate about accessing information about MPI opaque objects. One issue is whether there should be a visible and unique id for communicators (useful for profiling) *11,3,9*. It was pointed out that, at least for profiling, a communicator id is not required, because a profiling library can save enough information to "reconstruct" the needed communicator matching information. After this discussion, a vote was taken on whether to continue to consider other accessor function; it passed *17,2,4*.

Next, we discussed generalized requests. Concern was raised about duplication of function with the handlers (interrupt-receive); Bill Gropp raised the issue of how waits are implemented if a generalized request is *not* using a handler (which is not required in the text), since the `Mark_completed` routine might not be called by a handler but rather by another routine (in another thread or signal handler). A straw vote was taken to see if we should keep generalized requests *3,6,12*.

At this point, we had a discussion of `MPI_Status`. Many new functions are putting data unrelated to receiving data (e.g., `MPI_TAG`, `MPI_SOURCE`, and `MPI_Get_count(&status,&size)`) in the status. A vote on reusing status fields passed *19-0-2*.

- Source, tag undefined but user can't modify value *16*
- The implementation can't reuse either *2*

Closed the day with a discussion of cancelling collective communication (for nonblocking collective routines).

6 Thursday

We opened with a discussion of extension of collective communication to intercommunicators. There are two major forms, the "half" and "full" duplex versions. The straw vote was: full *15*, half *?* (but less than 15), abstain *7*. There was no straw vote on eliminating intercommunicator collective operations.

For the collective operations on intercommunicators, in the half-duplex case, there is a single process participating in one of the two groups (for example, a broadcast from one member of one group to all members of the other group). A question is: are all members of the intercommunicator required to call the routine? A straw vote was taken to require this which passed *14-4-7*. The reasoning is that doing otherwise introduces collective routines that not all members of the communicator must call.

What value does the root process use for the root in this case? Its rank or `MPI_ROOT`. `MPI_ROOT` was prefered *6-3*.

7 Friday

Friday addressed remaining issues.

8 Miscellany

Issues of interlanguage operability were discussed. Bill Gropp noted that a proposed approach for `MPI_BOTTOM` does not work if interoperability with another language (other than Fortran) that does not support pointers is added. One such language is Java; other "network safe" languages are likely to have the same problems.

We then moved on to corrections and clarifications to MPI 1.

The first item is handling of character arguments in Fortran which was defered since the author could not attend this meeting.

An inconsistency in the MPI 1 standard about intercommunicators was resolved by deleting one of the inconsistent statements and a bogus rationale. *18-0-5* Bill Gropp agreed to provide text to replace the bogus rationale.

There was general agreement to make an `MPI_VERSION` available to users.

There was general agreement to remove Appendix B (a sample `C++` class library for MPI).

Bill Saphir presented a summary of the many issues in a Fortran 90 binding for MPI. In particular, array sections (which are often passed by implementations by value-result) are a problem for the MPI non-blocking send/receive routines.

9 Realtime

Lloyd Lewins presented the issues that the Realtime committee is working on.

10 Closing

Rusty Lusk closed the meeting; the Forum thanked the local organizers, Ian Glendening and Helga P.