Minutes of MPI meeting
October 23–25
Denver CO

Rusty Lusk

November 10, 1995

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. Official 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*. The Monday minutes are pretty rough; the Wednesdy summaries are a little better. Please send any corrections to `lusk@mcs.anl.gov`.

2 Monday

Rusty Lusk opened meeting with a discussion of the agenda. Then we had runthroughs of the state of each chapter

2.1 Miscellaneous

Rusty Lusk will take charge of this chapter.

MPI_Type_Contiguous_Struct - Lloyd Lewins.

Issue: Use of memcpy wants to block moves, allowing holes to be overwritten.

Alternative: could be attribute of type.

MPI_Init proposal discussed by Steve Lederman. See proposal.

Interlanguage handle translations, Leslie Hart, see proposal, except for ¡X¿ is gone and there is a third argument which is the datatype.

Steve Huss-LedermanL represent Clint's proposal to have a neutral format. Cached attributes are not lost in translation. See proposal.

Marc Snir's interlanguage part.

See proposal. "MPI_Objects" is alternative to above proposal.

Also constants like MPI_Proc_Null, etc. Problems with MPI_BOTTOM. If we say MPI_BOTTOM could not be used in initialization (contrary to MPI-1) then this problem would go away.

Heidi Poxon presented the problem with character strings. See proposal.

Steve presents Clint Whaley's proposal on MPI_ERR_NORESOURCE. Not enthusiastically received.

Null status, making arguments more robust.

Steve Huss-Lederman presents Whaley proposal on communicator formation without using whole parent communicator. Not enthusiastically received because not MPI-like. This problem might be solved with interlanguage passing of communicators.

But if want to call same library made with either MPI or PVM communication library.

2.2 Dynamic

Bill Saphir summarizes history of dynamic chapter.

Presentation of the chapter.

Discussion:

Spawn:

What about partial success?

Should universe size be attribute on input communicator or global constant?

Full interface:

Holding state awkward between init and start

Maybe they could be joined in a multispawn, which has arrays for all of the arguments of MPI-Spawn.

Needs root argument.

Non-blocking versions.

Discussion wants to put "what" back in.

MPI_Request_notification of asynchronous_event.

Attaching to independent processes. See proposal. Will need non-blocking versions, at least for accept. "connect" instead of "contact".

2.3 One-sided communication

Marc Snir. See proposal.

Note in current definition, remote-memory-copy communicators are not usable for message-passing. Lloyd thinks than then they shouldn't be communicators.

Collective operations (MPI_IGET_GATHER) aren't as necessary as for message-passing. May disappear.

MPI_TYPE_GLOBAL suggested by Paul Pierce to save time in sending the remote process the receive datatype.

hsend, hrecv could be replaced by attaching handlers to request in general.

MPI-hwait(request, extra_info, status, handler) posts handler, which is executed when the request is satisfied.

Need to discuss fairness.

2.4 Bindings

Andrew Lumsdaine presented language bindings: C++ bindings will be just the same as C bindings, but there will be a "recommended" class library.

2.5 External interface

Steve Huss-Lederman presented external interface chapter. See proposal.

Generalized requests built on hsend, hrecv. Might want to shift to hwait. Saphir: can't use these for say, iallocate.

3 Tuesday

All subcommittees met on Tuesday for detailed discussions. Summaries of their discussions were given on Wednesday morning.

4 Wednesday morning

Future meetings: The Vienna meeting is Jan 24-26, meeting all day Thurs-Fri.

Eric Salo proposed to delete the One-sided communication chapter, with the possible exception of hrecv. Discussion indicated that there was support for the functionality contained in this chapter. One of the key issues is whether the get/put area should be constrained to be a special area of memory or unconstrained (so that it could be on a process's stack, for example). If it is to be a specially-initialized area of memory, how is it to be obtained and identified in Fortran.

It was agreed that there may be a major counter-proposal for the one-sided operations for next time.

The rest of the Miscellaneous meeting was held, as a subcommittee of the whole. Heidi Poxon led the discussion of character data in Fortran, and Lloyd Lewins led the discussion of the "contiguous struct" datatype proposal.

4.1 Subcommittee reports

4.2 Report from One-sided

Report from Marc Snir.

Issues Discussed:

- restriction of windows

- How to do RMC_MALLOC in Fortran.
- took out collective collective RMC.
- took out active messages.
- no hsend/hrecv, introduced hwait instead.

  ```
  hwait(request, extra-info, handler).  handler(status, extra\_info).
  ```

  can be used with persistent requests. No restriction on what handler can do.

  Question on what thread runs the handler. If we restrict handler to be executed by thread that posted it.

  Is the request "consumed" by the hwait. If so... If not...

4.3 Report from Dynamic

Report from Bill Saphir.

There are many changes from last time, which now seem settled in.

- No resources.
- One can find number of processes that might be usefully spawned by predefined variable MPI_UNIVERSE_SIZE. (Better name needed.)
- No process object, only group-rank.
- We hashed out the details of MPI_SPAWN and MPI_SPAWN_INDEPENDENT.
- We took a convincing vote on including management of non-MPI processes (Spawn-independent and Signal, Notify). We definitely do want this functionality.
- The Client-Server material needs more work.

4.4 Report from Collective

Summary by Tony Skjellum.

Progress to report:

Asynchronous collective:

- Symmetry: Can MPI-Barriers and MPI_Ibarrier-MPI_Waits be mixed?

Straw vote *7-6-4+*.

Strong Ordering

- per communicator
- all operation specified in same order in each process
- arbitrary number of outstanding requests
- how many progress concurrently is implementation-defined, (but at least one).

Straw vote: *14-0-0*

Additional planned actions, by consensus:

- Extend functionality to intercommunicators
- address implementation and application "thread compliance" separately.

Summary

Asynchronous collective operations "in" MPI-2, but schedule needs discussion. E.g. basic support defined by Dec 31, 1995, symmetry support defined by Dec 31, 1996, or one year later?

Better all-to-all communication (e.g. transpose)

Lloyd Lewins proposal added to chapter.

Plans for Vienna meeting:

- In-place proposal(s) complete (Fleischmann et. al.)
- intercommunicator collectives (blocking) complete
- schedule for their inclusion in MPI-2
- Updated proposal content on
  - non-blocking
  - add "cancel" discussion
  - better all-to-all family of operations
  - revised tentative schedule for completing entire chapter.

4.5 Report from Miscellaneous

Report from Rusty Lusk.

The "Miscellaneous" chapter is beginning to take shape along the lines of several identifiable subtopics, and specific individuals are taking responsibility for the subtopics.

1. Standardizing MPI program Startup (Rusty Lusk). Possibilities:
   - do nothing
   - Standardize mpirun (e.g. `mpirun -np 64 myprog`)

    - Standardize program arguments (e.g. `myprog -mpi 'np 64'`)

   The general consensus favored an attempt to standardize at least a part of mpirun. Several versions already exist, and we will study them to see what they have in common.

2. Interlanguage issues (Marc Snir). Discussion centered on the following topics:

    - MPI_INIT (args). Either C init or Fortran init should do all necessary initialization. C init will not access argc argv
    - Objects (multiple proposals).

      Opaque objects. Each language has functions to convert handles to integers. Status cannot be passed. same opaque object in all languages.
    - Attributes. An attribute should be "big" enough for both fortran and C. If sizeof(Integer) ¡ sizeof(void*) the fortran assigns and reads least significant bits.

      Datatype has same meaning (specifies same data layout and same type signature.) C can send Fortran types, and vice versa.

      Constants and predefined attributes have same value in C and Fortran.

      MPI_Bottom cannot be used for initialization in C.
    - Requests
    - Communication

      C to F communication:

      C can use a datatypes defined F and vice-versa.

      C can construct an F datatype.

      F can construct a C datatype.
    - Addresses

3. Fortran character strings (Heidi Poxon). There was discussion of the problems caused for character strings by the MPI-1 decision to allow multiple types for buffer addresses. Most compilers seem able to deal with this in one way or another, but we are not sure. Three ways to fix this were discussed.

4. More robust arguments (Rusty Lusk). There are a number of places where the standard could specifically allow arguments that ordinarily might be considered invalid, but should be allowed. (Passing NULL as `status`, for example, when the status will not be checked.) A proposal will be forthcoming.

5. More general types (Lloyd Lewins)
   - `MPI_CONTIGUOUS_STRUCT` was discussed, which would specifically allow "holes" in the receive buffer to be overwritten.
6. We decided not to pursue a submission on a new type of error-handling behavior because it could be managed with user-defined error handlers.
7. We decided not to pursue a submission on communicator formation that did not require participation by all members.

4.6 Report from External

Report by Steve Huss-Lederman.

4.6.1 Generalized Requests

- Use Communicate & Call functions
  - Use MPI_REQUEST_MARK_COMPLETE to finish
- Open Issues
  - Status fields: (also applicable to isend status)
    * Erroneous to check status except for error field
    * Fields other that status unchanged.
  - Cancel: still needs work: allow?

4.6.2 Interesting ideas but performance concerns

- Communicator ID
- Event callback

We will remove these for now, possibly reinstate if can solve performance problems or implement with profiling library

4.6.3 Datatype Accessing:

- include ability to convert to string
  - User provides string storage: if compiler limits length in Fortran then can't get all info.
- Two proposals to traverse datatype tree
  - low-level

  - high-level

  We will see if the high-level can be implemented with the low-level. May want both for different users.

4.6.4 Datatype compacting

- Add function to tell true extent of datatype
- See if we can do what we want with this.

4.6.5 Cacheing on handles

- Not on MPI_REQUEST or MPI_STATUS
- Must look at others new handles to consider

4.6.6 Debuggers

Need more discussion to find out what is needed for debuggers.

4.7 Report from Bindings

Report from Jeff Squyres.

4.7.1 Fortran-90

Fortran-90 has several new features that need discussion to determine how they could be used in the MPI spec:

- dynamic memory (welcome to 1920)
- derived data types (not quite structs)
- array sections (disallow some operations
- strong typing (too strong)
- modules (useful)
- pointers (that are not addresses)

4.7.2 C++

C++ binding alternatives:

- C only
- C with reference semantics
- conservative C++ with objects (like MPI++)
- class library (like OOMPI)

Results of subcommittee discussions:

- Conservative C++ is the favorite choice. Will there be interoperability problems?
- endorse class library as an annex
- Is there interest in language-independent
  - objects
  - class library

  (to guide Eiffel, Smalltalk, ADA, OOCOBOL)?

4.8 Report from Real-time

Report from Tony Skjellum.

Existing proposals/material constituting “chapter”.

- time-driven proposal (Arkady)
- priority messages proposal (MSU group)
- Overview of issues writeup (MSU group)
- The *empty* virtual channel proposal (TBD)

Time Driven:

- Reaching closure on final proposal for this area.
- MPI to address
  - synchronized clocks
  - key functionality for scheduling messages
  - “Middleware” exemplars will demonstrate
  - Experiments to follow

Predictable MPI Performance needed.

Priority messages

- priority per communicator (per message)

Other entities to be considered: processes, threads, ...

- messages inherit communicator priority
- communicators inherit priority via dups
- mapping mechanism
- still need to ask about network preemption
- do applications need to control/impact packetization "sizes"?

Issues (religious arguments)
Three major models: time, priority, event are related
Criticality (e.g. super-important messages) is another "approach" to real time.
Areas to be pursued:

- Timeouts for messages
- Lossy protocols, MPI streams
- Instrumentation issues
- Others noted above
- Virtual channels

Where Chapter will b e by Supercomputing '95:

- It will be an integrated document :-)
- committee promises to iterate on
  - time driven
  - priority

  proposals.

...by Vienna ...

- Virtual channel proposal
- some early experimental experience?
- Examine a number of real-time OS's
- Try out ideas on real-time hardware.

...by next USA meeting ...

- tentative schedule for time-driven.

5 Attendees

Name	Institution	email
Robert Babb	U of Denver	babb@cs.du.edu
Joel Clark	Intel	joel@ssd.intel.com
Raja Daoud	Convex	raja@convex.com
Doug Doerfler	Sandia	dwdoerf@sandia.gov
Anne C. Elster	Schlumberger	elster@slb.com
Stephen Fleischman	Convex	sfleisch@convex.com
Hubertus Franke	IBM	frankeh@watson.ibm.com
Al Geist	Oak Ridge	gst@ornl.gov
Kei Harada	Hitachi	haradeke@soft.hitachi.co.jp
Leslie Hart	NOAA/FSL	hart@fsl.noaa.gov
Tom Henderson	NOAA/FSL	hender@fsl.noaa.gov
Hans-Christian Hoppe	PALLAS	hch@pallas.de
Steve Huss-Lederman	CCS	lederman@super.org
Arkady Kanevsky	Mitre	arkady@mitre.org
Koichi Konishi	NEC	konishi@research.nj.nec.com
Susan Kraus	NEC	kraus@hstc.necsyl.com
Mark Law	Sanders	mlaw@sanders.com
Lloyd Lewins	Hughes	llewins@msmail4.hac.com
Ziyang Lu	Miss State	lu@cs.msstate.edu
Andrew Lumsdaine	Notre Dame	Andrew.Lumsdaine@nd.edu
Rusty Lusk	Argonne	lusk@mcs.anl.gov
Thom McMahon	Miss State	thom@erc.msstate.edu
Nick Nevin	OSC	nevin@alex.osc.edu
Jarek Nieplocha	PNL	j-nieplocah@pnl.gov
Paul Pierce	Intel	prp@ssd.intel.com
Heidi Poxon	Cray	hlg@cray.com
Arindam Saha	Miss State	saha@ee.msstate.edu
Eric Salo	SGI	salo@sgi.com
Bill Saphir	NASA/Ames	wcs@nas.nasa.gov
Fred Shirley	Sanders	fshirley@sanders.com
Lance Shuler	Sandia	shuler@cs.sandia.gov
Tony Skjellum	Miss State	tony@Aurora.CS.MsState.Edu
Marc Snir	IBM	snir@watson.ibm.com
Jeff Squyres	Notre Dame	jsquyres@lsc.cse.nd.edu
Greg Tensa	Intel	gregt@ssd.intel.com
MaryDell Tholburn	Los Alamos	marydell@lanl.gov