MPI Forum - Overview

LEADERSHIP
COMPUTING FACILITY
NATIONAL CENTER FOR COMPUTATIONAL SCIENCES

presented by

Richard L. Graham

Chairman

Oak Ridge National Laboratory
U.S. Department of Energy

Outline

- Goal
- Forum Structure
- Meeting Schedule
- Scope
- Voting Rules

Goal

To produce new versions of the MPI standard that better serves the needs of the parallel computing user community

Structure

- Chairman and Convener: Rich Graham
- Secretary: Jeff Squyres
- Steering committee:
  Jack Dongarra
  Al Geist
  Rich Graham
  Bill Gropp
  Andrew Lumsdaine
  Rusty Lusk
  Rolf Rabenseifner

Face-To-Face Meetings

- Jan 14-16, 2008
- March 10-12, 2008
- April 28-30
- June 30 - July 2, 2008 - Menlo Park, CA
- Sept 3-5, 2008 Dublin, Ireland (Euro PVM/MPI is Sept 7-10, 2008 in Dublin)
- October 20-22, 2008 - Chicago, IL
- December 15-17, 2008 - Menlo Park, CA

Scope of the MPI Form

LEADERSHIP
COMPUTING FACILITY
NATIONAL CENTER FOR COMPUTATIONAL SCIENCES

presented by

Oak Ridge National Laboratory
U.S. Department of Energy

What is MPI ?

- A standard
- A library
- Provides communications primitives (network and file-system)
- Provides some process control capabilities
- O/S agnostic
- H/W agnostic
- Aimed at
  - HPC and ?
  - Clusters
  - SMP’s

Changes to MPI ?

Consistent with MPI’s goals

==>

Developing Three new versions of the Standard - 2.1, 2.2, 3.0

Only ONE version defined at a given point in time

Scope of MPI 2.1, 2.2, and 3.0

LEADERSHIP COMPUTING FACILITY
NATIONAL CENTER FOR COMPUTATIONAL SCIENCES

presented by

Oak Ridge National Laboratory
U.S. Department of Energy

MPI 2.1 - Scope

Clarification to the MPI standards document, resulting in a single document describing the full MPI 2.1 standard. This includes merging of documents, text corrections, and added clarifying text.

MPI 2.2 - Scope

Scope: Small changes to the standard. A small change is defined as one that does not break existing user code - either by interface changes or semantic changes - and does not require large implementation changes.

MPI 3.0 - Scope

Additions to the standard that are needed for better platform and application support. These are to be consistent with MPI being a library providing of parallel process management and data exchange. This includes, but is not limited to, issues associated with scalability (performance and robustness), multi-core support, cluster support, and application support.

Backwards compatibility is maintained - Routines may be deprecated

Current state of MPI 2.1, 2.2, and 3.0

LEADERSHIP
COMPUTING FACILITY
NATIONAL CENTER FOR COMPUTATIONAL SCIENCES

presented by

Oak Ridge National Laboratory
U.S. Department of Energy

MPI 2.1

Scope : Clarification to the MPI standards document, resulting in a single document describing the full MPI 2.1 standard. This includes merging of documents, text corrections, and added clarifying text.

MPI 2.1 Primary Author: Rolf Rabenseifner

- First full draft of combined document is in place
  - Text addressing one topic collocated
  - References to old versions of MPI removed
  - All functions have C, Fortran, and C++ prototypes in the same location
  - Deprecated routines moved to Appendix (phased out over a LONG time)
  - Total of 16 chapters +2 Appendices
- Official reading occurred last week
- Straw vote passed unanimously
- Targeted final approval: Sept 2008, right before Euro PVM/MPI Users Meeting

MPI 2.2

Scope: Small changes to the standard. A small change is defined as one that does not break existing user code - either by interface changes or semantic changes - and does not require large implementation changes.

MPI 2.2 Primary Author: Bill Gropp

- Effort just getting off the way. Short duration - fix “urgent” issues.
- More corrections to the MPI document (superceding MPI 2.1)
- Current suggested changes
  - Allowing concurrent access to user send buffers
  - Interface changes indicating buffer usage (adding the c-”const” keyword)
  - Usage guarantees from the application to the library
  - Allowing more than 2^32 in the send/recv “count” parameter (maybe 3.0)
  - Update data-type support (make sure it supports the current C, C++, and Fortran standards)

MPI 3.0

Scope:Additions to the standard that are needed for better platform and application support. These are to be consistent with MPI being a library providing of parallel process management and data exchange. This includes, but is not limited to, issues associated with scalability (performance and robustness), multi-core support, cluster support, and application support.

Current Working Groups

- Application Binary Interface : Jeff Brown - Los Alamos National Laboratory
- Collective Operations : Andrew Lumsdaine - Indiana University
- Fault Tolerance : Richard Graham - Oak Ridge National Laboratory
- Fortran Bindings : Craig Rasmussen - Los Alamos National Laboratory
- Generalized Requests : George Bosilca - The University of Tennessee
- MPI Sub-Setting : Alexander Supalov - Intel Corporation
- Point-To-Point Communications : Ron Brightwell - Sandia National Laboratory
- Remote Memory Access : Bill Gropp, University of Ilinois Champaign/Urbana - Rajeev Thakur, Argonne National Laboratory

Application Binary Interface

Goal: To define any additional support needed in the MPI standard to enable static and dynamic linkage compatibility across MPI implementations on a target platform for MPI based Applications.

- Aiming at C support first
- Standardizing on the values of pre-defined constants
- Standardizing on the sizes of opaque handles
- Looking at run-time issues

Collective Operations

Goal: Improved support for collective communications.

Current items:

- Non-blocking collectives
- Topology aware collectives

Fault Tolerance

- Goal: To define any additional support needed in the MPI standard to enable implementation of portable Fault Tolerant solutions for MPI based applications.
- Items being discussed
  - Define consistent error response and reporting across the standard
  - Clearly define the failure resonse for current MPI dynamics - master/slave fault tolerance
  - Support for recovery from failed processes
  - Data piggybacking
  - Dynamic communicators
  - Asynchronous dynamic process control

Generalized Requests

- Goal: Redefine the generalized requests interface. A more flexible interface between the user defined requests and the MPI library is required in order to allow the provider of the generalized request to integrate a progress function inside the MPI library. The ultimate goal is to allow the generalized requests progress to be done without a special test on wait function.

MPI Sub-setting

- Goal: To establish a mechanism by which MPI implementations can provide support for a subset of the full MPI standard, maintaining full API and semantic compatibility with the complete MPI standard. This is aimed at allowing optimization opportunities such as for performance or resource foot-print.
    - Functionality sets
    - Mandatory vs. optional sets
    - Sub-setting for performance
    - Sub-setting for memory foot print
    - How does one support niche communities

Point-To-Point Communications

- Goal: To re-examine the MPI peer communication semantics and interface, and consider additions and/or changes needed to better support point-to-point data movement within MPI.
  - Better support in threaded environment for probe/recv
  - Better communications between memory allocator and data-manipulation routines

Remote Memory Access

- Goal: To provide improved support for Remote Memory Access.
  - Read-Modify-Write operations
  - Flexible RMA synchronization
  - Registration of data for one-sided operations

Just getting off the ground

What should be the focus of MPI 3.0 ?

Voting Rules

- There is one vote per organization, which must be present at the meeting when the vote is taken.
- To vote, an organization must have been present at two of the last three meetings.
- Votes are taken twice, at separate meetings. Votes are preceded by a “reading” at an earlier meeting, at which straw votes may be taken.
- Measures pass on simple majority.
- Only items consistent with the charter can be considered.

Bringing Items To A Vote

- Working group is established in meeting N (may be folded into a larger working group)
- Working groups is opened up to all
- Working group brings specific proposal to discussion before the full forum in subsequent meeting
  - Schedule with me at least 4 weeks prior to next face-to-face meeting
  - Provide Chairman and Secretary draft proposal in LaTex format
- Straw vote taken after discussion in the full Forum
- Formal voting process proceeds, if/when working group is ready to bring this for vote.

Committee Rules

- Any one may propose a committee
- Must submit proposal in electronic form to the secretary
- Must be consistent with the Charter and Scope
- Need a minimum of 4 organizations supporting the proposal
- Semantics before API
- Need prototype implementation, with source code, for a given proposed feature. Ideally, this would be in one of the widely used Open Source implementations, such as MPICH and/or Open MPI.

On Line Information

meetings.mpi-forum.org

- Meeting Schedule
- Meeting logistics
- Mailing list signup
- Mail archives
- Wiki pages for each working group

Questions - Comments ?

- Does MPI provide all your current communications and process control needs ?
- What do you like about MPI ?
- What do you dislike about MPI ?
- How can MPI change to help you ?

Want to get involved ?