The Final Steps to MPI 4.0

Martin Schulz

Technische Universität München

Chair of the MPI Forum

+ the entire MPI Forum

SC 2020 BoF

November 2020

The MPI Forum Drives MPI

Standardization body for MPI
- Discusses additions and new directions
- Oversees the correctness and quality of the standard
- Represents MPI to the community

Organization consists of chair, secretary, editor, convener,
and member organizations

Open membership
- Any organization is welcome to participate
- Consists of working groups and the actual MPI forum (plenary)
- ~~Physical~~ meetings 4 times each year (3 in the US, one with EuroMPI/Asia/USA)
  - Working groups meet between forum meetings (via phone)
  - Plenary/full forum work is done mostly at the ~~physical~~ meetings
- Voting rights depend on attendance
  - An organization has to be present two out of the last three meetings (incl. the current one) to be eligible to vote

The Bulk of Work is in the Working Groups

Collective Communication, Topology, Communicators, Groups
- Torsten Hoefler, Andrew Lumsdaine and Anthony Skjellum

Fault Tolerance
- Wesley Bland, Aurélien Bouteiller and Rich Graham

HW Topologies
- Guillaume Mercier

Hybrid and Accelerator Programming
- Pavan Balaji and Jim Dinan

Large Count
- Jeff Hammond and Anthony Skjellum

Persistence
- Anthony Skjellum

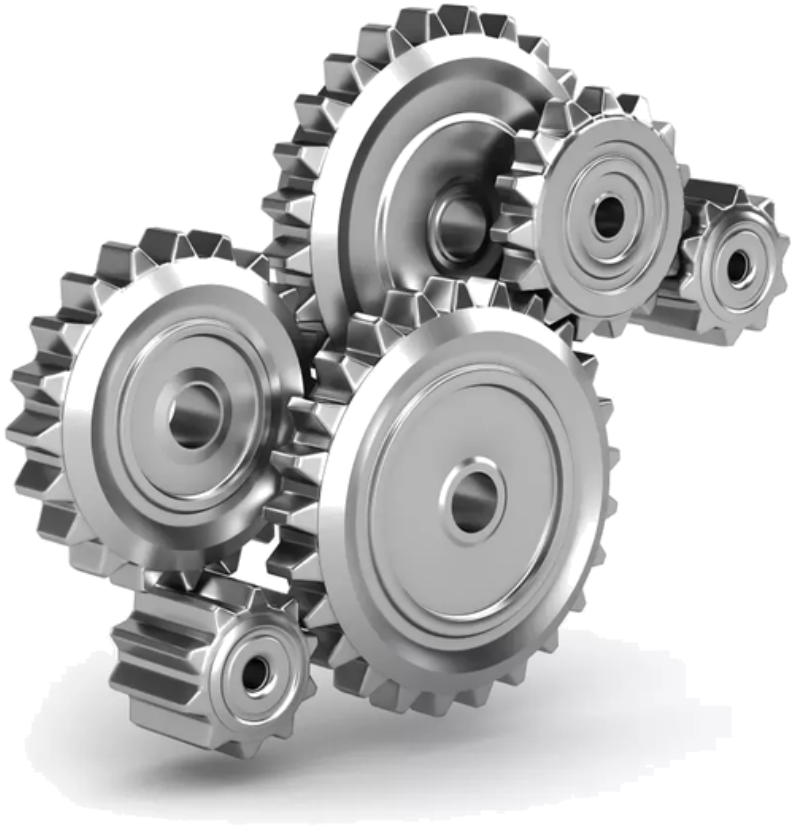
Point to Point Communication
- Rich Graham and Dan Holmes

Remote Memory Access
- Bill Gropp and Rajeev Thakur

Semantic Terms
- Rolf Rabenseifner and Purushotham Bangalore

Sessions
- Dan Holmes

Tools
- Kathryn Mohror and Marc-Andre Hermanns



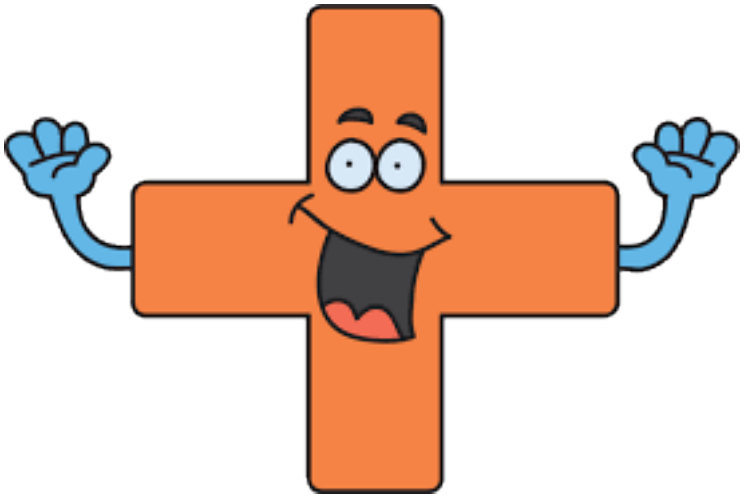
The Status of MPI

MPI 3.0 ratified in September 2012
- Major new functions

MPI 3.1 ratified in June 2015
- Minor updates and additions

Fully adopted in all major MPIs

MPI 4.0 work coming to an end
- Release Candidate for SC
- Is available at http://www.mpi-forum.org/



Major additions for MPI 4.0
- Solution for “Big Count” operations
- Persistent Collectives
- Partitioned Communication
- Topology Solutions
- New init options via MPI Sessions
- Simple fault handling to enable fault tolerance solutions
- New tool interface for events

Big Count aka. Embiggenment

Problem: in current interface “count” arguments are “int”

- Limits communication volumes to 32bit x Datatype
- Significant number of applications need more
- Initial datatype “trick” no longer sufficient

Solutions discussed include:

- Just changing “int” arguments to “MPI_Count” arguments → ☹ ☹ ☹
- Polymorphic bindings → ☹ ☹
- Duplication of interfaces: with int and with MPI_Count (“_c” suffix) → ☹

Changes required

- Update of the general type rules for bindings
- Verification of all bindings, which led to errata tickets
- Addition of many new routines with “_c”

Status: voted into MPI 4.0 / minor errata pending

Persistent Collectives

Following the basic ideas of persistent point to point

- One-time initialization to pass all arguments, which returns a request
- Use of this request to start communication
- Completion using Test/Wait
- Reuse request to restart the operation as often as one wants

Available for all MPI collective communication operations (and barriers)

Why?

- Specify repeated operations
- Ability to lock down resources and to cache execution plan
- Performance optimization after (small) 1x cost
- Allows for continuous plan optimization

Status: voted into MPI 4.0

Partitioned Communication

Core idea – efficient highly concurrent communication

- Built on the concept of persistent operations
- Send buffers are split into partitions
  - Fill each partition and mark it as ready (good for threads/GPUs)
- Receive buffers are split into partitions
  - Individual notifications for each arriving partition
- Enables partial data transfers

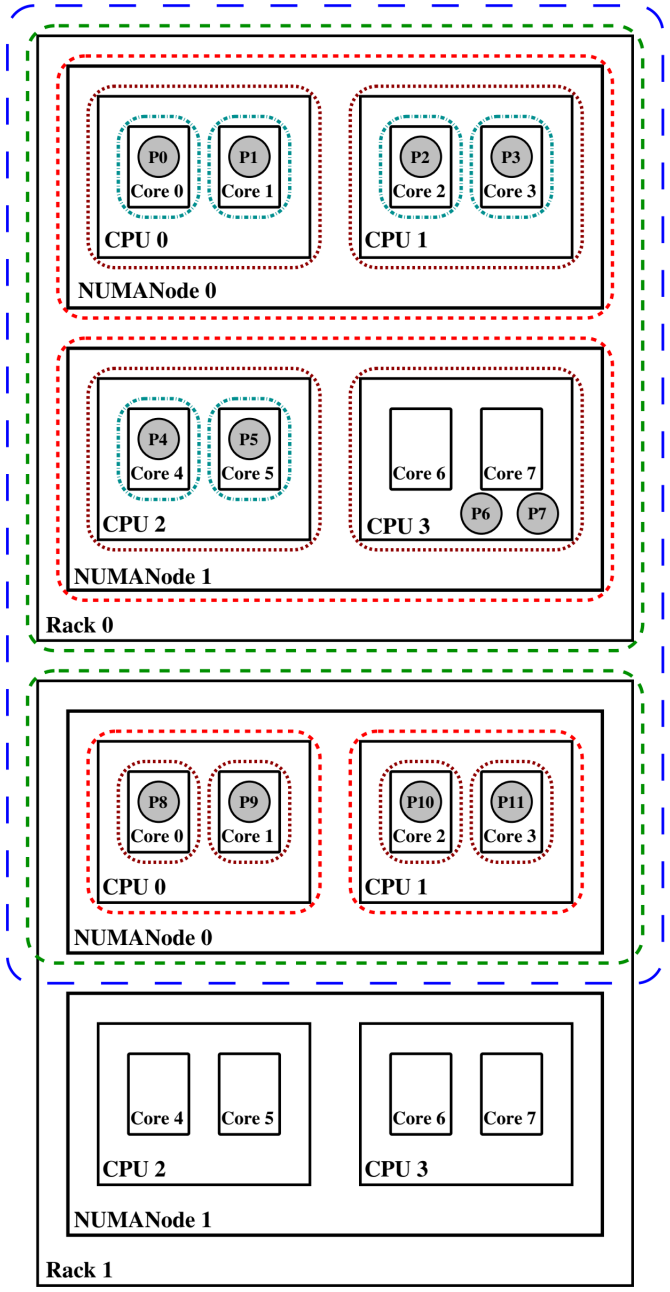
Notifications - on send and receive side – are light-weight

- May be driven from accelerators
- May need additional synchronization to trigger message transfer safely

So far only simple point-to-point options, more to come - GPU optimizations

Status: voted into MPI 4.0 (as a new chapter)

New Ways to Adapt to Hardware Topologies



New systems are hierarchical

- Mapping of processes to resources is critical
- Need topology-aware communicators

Feature Based on MPI_COMM_SPLIT_TYPE

- Introduces 2 new split type values
- Guided Mode
  - Info key to specify hardware level
- Unguided Mode
  - Start from the input communicator (e.g., MPI_COMM_WORLD)
  - Step-wise go to lower/deeper levels
  - Iterative until leaf is reached

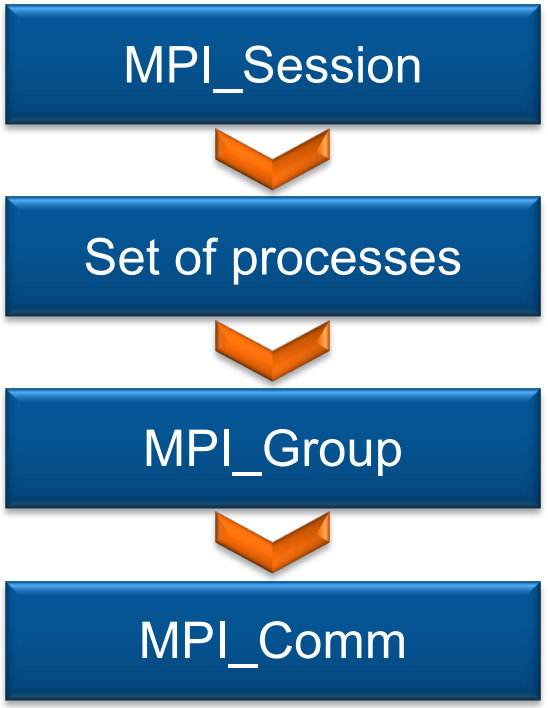
- Query function missing

Status: voted into MPI 4.0

A New Way to Use MPI: MPI Sessions

Basic scheme

1. Get local access to the MPI library
   Get a Session Handle
2. Query the underlying run-time system
   Get a “set” of processes
3. Determine the processes you want
   Create an MPI_Group
4. Create a communicator with just those processes
   Create an MPI_Comm



MPI Session’s intended goals

- No more implicit MPI_COMM_WORLD
- Enable runtime information to flow into MPI
- Creation of communicators without parent communicators
- Re-initialization of MPI
- Resource isolation
- Many future uses … - more later

Status: voted into MPI 4.0

Improved Error Handling

Goal: allow applications to limit impact of failures to avoid terminations

- Specify that MPI_SUCCESS indicates only the result of the operation, not the state of the MPI library.
- Localize error impact of some MPI operations. (e.g. MPI_ALLOC_MEM will now raise an error on COMM_SELF, not COMM_WORLD)
- Specify that MPI should avoid fatal errors when the user doesn't use MPI_ERRORS_ARE_FATAL
- New MPI Error Handler - MPI_ERRORS_ABORT
- Allow the user to specify the default error handler at `mpiexec` time.

What can you do with this?

- Point to Point communication with sockets-like error handling
- Enables manager/worker and other non-traditional types of applications
- Enterprise applications that want to move from sockets to MPI can do so.

Status: voted into MPI 4.0

MPI_T Events: Callback-driven event information

Motivation

- PMPI does not provide access to MPI internal state information
- MPI_T performance variables only show aggregated information

New interface to query available runtime event types

- Follows the MPI_T variable approach
- No specific event types mandated
- Event structure can be inferred at runtime

Register callback functions to be called by the MPI runtime

- Runtime may defer callback invocation (tool can query event time)
- Runtime may reduce restrictions on callback functions per invocation
- Callback can query event information individually or copy data en bloc

Status: voted into MPI 4.0

Other Additions

Assertions for message traffic to guide optimization
- Can state that an application doesn't use wildcards
- Enables traffic optimizations
- Great opportunities for implementations to optimize

Remove info key propagation on communicator duplication
- New function: MPI_Comm_idup_with_info
- Better control over properties attached to communicator

Clarification of what it means to query the info object attached to an MPI object

Deprecation of send cancel
- Long overdue ☺

Small fixes to the MPI Tools Information Interface

Access to MPI Info before MPI initialization (needed for Sessions, MPI_T, FT, …)

The Final Steps to MPI 4.0

Release Candidate available
- MPI Forum Website
- Feedback from the wider community wanted !!!

Final ratification process
- MPI-Forum in December: Validation and First vote
- MPI-Forum in February: Second vote and final ratification

Major open source MPIs are already moving towards MPI 4.0 support
- Open MPI
  - Already added MPIX versions of split types for hardware topology and added support for MPI_ERRORS_ABORT
  - Actively working on MPI_T events, ULFM and sessions support
- MPICH
  - MPICH will soon start a new major release series: MPICH 4.0, slated for Fall 2021
  - Preview releases will likely be made available earlier as major features are added
  - Research prototypes for some features (like comm info assertions) already exist
- MVAPICH
  - MVAPICH is following the MPICH releases and should be up to date with the MPICH releases
  - The team will be working on optimized solutions for MPI 4.0 features (persistent collectives, partitioned communication, hardware topology support, MPI_T callback support) to be released in subsequent months
  - Following the initial support, there will be a GA version with optimized solutions for MPI 4.0 features