Lecture 37: New Features of MPI-3

William D Gropp

www.cs.illinois.edu/~wgropp
Thanks To

• Ewing Lusk
• Pavan Balaji
• Rajeev Thakur
Overview of New Features in MPI-3

• Major new features
  ♦ Nonblocking collectives
  ♦ Neighborhood collectives
  ♦ Improved one-sided communication interface
  ♦ Tools interface
  ♦ Fortran 2008 bindings

• Other new features
  ♦ Matching Probe and Recv for thread-safe probe and receive
  ♦ Noncollective communicator creation function
  ♦ “const” correct C bindings
  ♦ Comm_split_type function
  ♦ Nonblocking Comm_dup
  ♦ Type_create_hindexed_block function

• C++ bindings removed
• Previously deprecated functions removed
Nonblocking Collectives

- Nonblocking versions of all collective communication functions have been added
  - MPI_Ibcast, MPI_Ireduce, MPI_Iallreduce, etc.
  - There is even a nonblocking barrier, MPI_Ibarrier

- They return an MPI_Request object, similar to nonblocking point-to-point operations

- The user must call MPI_Test/MPI_Wait or their variants to complete the operation

- Multiple nonblocking collectives may be outstanding, but they must be called in the same order on all processes
Neighborhood Collectives

- New functions MPI_Neighbor_allgather, MPI_Neighbor_alltoall, and their variants define collective operations among a process and its neighbors.
- Neighbors are defined by an MPI Cartesian or graph virtual process topology that must be previously set.
- These functions are useful, for example, in stencil computations that require nearest-neighbor exchanges.
- They also represent sparse all-to-many communication concisely, which is essential when running on many thousands of processes.
  - Do not require passing long vector arguments as in MPI_Alltoallv.
Improved Remote Memory Access Interface

- Substantial extensions to the MPI-2 RMA interface (MPI_Put, MPI_Get)
- New window creation routines:
  - MPI_Win_allocate: MPI allocates the memory associated with the window (instead of the user passing allocated memory)
  - MPI_Win_create_dynamic: Creates a window without memory attached. User can dynamically attach and detach memory to/from the window by calling MPI_Win_attach and MPI_Win_detach
  - MPI_Win_allocate_shared: Creates a window of shared memory (within a node) that can be accessed simultaneously by direct load/store accesses as well as RMA ops
- New atomic read-modify-write operations
  - MPI_Get_accumulate
  - MPI_Fetch_and_op (simplified version of Get_accumulate)
  - MPI_Compare_and_swap
Improved RMA Interface contd.

- A new “unified memory model” in addition to the existing memory model, which is now called “separate memory model”
- The user can query (via `MPI_Win_get_attr`) whether the implementation supports a unified memory model (e.g., on a cache-coherent system), and if so, the memory consistency semantics that the user must follow are greatly simplified.
- New versions of `put`, `get`, and `accumulate` that return an `MPI_Request` object (`MPI_Rput`, `MPI_Rget`, …)
- User can use any of the `MPI_Test/Wait` functions to check for local completion, without having to wait until the next RMA sync call
Tools Interface

- Beyond the PMPI profiling interface
- An extensive interface to allow tools (debuggers, performance analyzers, etc.) to portably extract information about MPI processes
- Enables the setting of various control variables within an MPI implementation, such as algorithmic cutoff parameters
  - e.g., eager v/s rendezvous thresholds
  - Switching between different algorithms for a collective communication operation
- Provides portable access to performance variables that can provide insight into internal performance information of the MPI implementation
  - e.g., length of unexpected message queue
- Note that each implementation defines its own performance and control variables; MPI does not define them
Fortran 2008 Bindings

- An additional set of bindings for the latest Fortran specification
- Supports full and better quality argument checking with individual handles
- Support for choice arguments, similar to (void *) in C
- Enables passing array subsections to nonblocking functions
- Optional ierr argument
- Fixes many other issues with the old Fortran 90 bindings
Miscellaneous Features

• Other new features
  ♦ Matching Probe andRecv for thread-safe probe and receive
  ♦ Noncollective communicator creation function
  ♦ “const” correct C bindings
  ♦ Comm_split_type function
  ♦ Nonblocking Comm_dup
  ♦ Type_create_hindexed_block function
• C++ bindings removed
• Previously deprecated functions removed
What did not make it into MPI-3

- Some evolving proposals did not make it into MPI-3
  - e.g., fault tolerance and improved support for hybrid programming
- This was because the Forum felt the proposals were not ready for inclusion in MPI-3
- These topics may be included in a future version of MPI

- Current activities of the MPI Forum (for MPI 3.x and MPI 4) can be tracked at [http://meetings.mpi-forum.org/](http://meetings.mpi-forum.org/)
- The full standard and other materials can be found at [http://mpi-forum.org](http://mpi-forum.org)