Status Report and Discussion
MPI Forum
Torsten Hoefler
Indiana University
Oct. 21st 2008
Chicago, IL, USA
1) Nonblocking Collectives Proposal Draft
2) Sparse/Topological Collective Operations
3) MPI_IN_PLACE, collective or not?
4) Persistent Collective Operations
5) Items from the Floor
High-level Decisions

Decisions made during last telecon

(Based on Straw-Votes during the Sept. Forum):

- Calls for everything (we didn't define what is useful yet)
- **No** mixing of blocking and nonblocking collectives
- Usage of MPI_Requests for request objects
- We allow multiple outstanding requests
  (implementations don't have to execute them simultaneously!)
- Ordering is global for all collectives (more later)
- Prefix: I_ (for immediate)
Mixing/Matching/Nesting

- Mixing of blocking/nonblocking colls must fail (prevent portability issues)
- No tags, matching is defined by issue-order
- Matching is defined globally for all collectives (no difference between different colls – see examples)
Example 1 - correct

Process 1
MPI_Ibarrier(req)
MPI_Bcast()
MPI_Wait(req)

Process 2
MPI_Ibarrier(req)
MPI_Bcast()
MPI_Wait(req)
Example 2 – incorrect – false matching

Process 1
MPI_Ibarrier(req)
MPI_Bcast()
MPI_Wait(req)

Process 2
MPI_Bcast()
MPI_Ibarrier(req)
MPI_Wait(req)
Example 3 - correct

Process 1

MPI_Ibarrier(req)
MPI_Send()
MPI_Wait(req)

Process 2

MPI_Irecv(req[0])
MPI_Ibarrier(req[1])
MPI_Waitall(req, 2)
Example 4 - correct

Process 1
MPI_Ibarrier(req)
MPI_Wait(req)
MPI_Send()

Process 2
MPI_Ibarrier(req)
MPI_Recv()
MPI_Wait(req)
Example 5 - correct

Process 1
MPI_Ibcast(req[0])  MPI_Ibcast(req[0])
MPI_Ibcast(req[1])  MPI_Ibcast(req[1])
MPI_Waitall(req, 2)  MPI_Waitall(req, 2)

Process 2
Other Issues

- Maximum number of outstanding requests
  - Might be limited by the hardware
  - Do we want to provide a query function
  - Number might be comm-specific
  - Do we want to enforce a minimum? Like 32768 tags for point-to-point messages.
Proposal Draft

- How do we handle comments to the proposal?
- It's in PDF format right now
- We want it in MPI style
- I volunteer to edit it
- Send me anything (marked up and scanned, change descriptions) – please no big files over ML
Examples in Proposal

• Which examples do we want to put in the draft?
  • All of them?
  • An application example (parallel compression or FFT?)
  • Also wrong examples?
Better wording for "matching"

- "Matching" is not really defined
- "At the same time" isn't correct
- Say something like "in logical order" (sounds weird)
- Any ideas?
How do we proceed?

• What do we do with the proposal?
  • Finish changes to draft until a week before next telecon
  • Discuss it at telecon
  • Read it at next forum?
Sparse/Topological Collectives

- Application examples:
  - Cart: CFD, regular stencil computations, Poisson solver
  - Graph: AMR, Sparse matrix operations, Parallel Graph
- Do we know applications or programmers to collaborate with?
  - Try implementations
  - Understand issues better?
  - Any contacts?
- We have TDDFT/Octopus already at medium scale
Sparse/Topological Alltoall

MPI_Sparse_alltoall(  sendbuf,
            sendcount,
            sendtype,
            [sendneighbors],
            recvbuf,
            recvcount,
            recvtype,
            [recvneighbors],
            [topo]comm)

- MPI_IN_PLACE?
- Really Alltoall? It's more like an [neighbor] exchange?
Sparse/Topological Alltoallv

MPI_Sparse_alltoallv( sendbuf,
    sendcounts,
    senddispls,
    sendtype,
    [sendneighbors],
    recvbuf,
    recvcounts,
    recvdispls,
    recvtype,
    [recvneighbors],
    [topo]comm)

• MPI_IN_PLACE? (probably not)

• Really Alltoallv? It's more like an [neighbor] exchangev?
Sparse/Topological Reduce

\[
\text{MPI\_Sparse\_reduce}( \text{sendbuf}, \\
\text{sendcount}, \\
\text{sendtype}, \\
[\text{sendneighbors}], \\
\text{recvbuf}, \\
\text{recvcount}, \\
\text{recvtype}, \\
[\text{recvneighbors}], \\
\text{op}, \\
[\text{topo}]\text{comm})
\]

- MPI\_IN\_PLACE? (probably not)
MPI_Sparse_reducev( sendbuf, sendcount, sendtype, [sendneighbors], recvbuf, recvcount, recvtype, [recvneighbors], op, [topo]comm)

- MPI_IN_PLACE?
Sparse/Topological Issues

- Do we want special operations for cartesian grids?
  - Shift operation
  - Neighbor communication with bigger stencils
- Groups or Topocomms (again)
  - Dublin: 11/2/13 for topocolls and 2/8/11 for groups (y/n/a)
- Do calls have to be collective on the communicator
  - Yes: would allow forwarding
  - No: would allow more asynchronism and more flexible programming models
Should MPI_IN_PLACE be collective?

- Picked up from MPI-2.2 discussions!
- MPI_Allreduce requires MPI_IN_PLACE to be collective
  - Why?
  - Should the same apply to Reduce_scatter
  - What about other collectives (Alltoall)?
Persistent Collectives/Issues

MPI_Startall()?

- another pro for tags
- in which order do similarly tagged colls match?
- Not defined in the point-to-point case
  - Do we want to do the same again?
- match in "array-order" or make the operation illegal?
Persistent Collectives/Issues

- Do we want to consider changing arguments of a persistent collective?
- Was this discussed earlier (MPI-2.0)?
- For example change local buffers or communication patterns
Persistent Collectives/Issues

• We need more research
• Use-cases could be:
  • Optimization of *v operations
  • Explicit cache for registered memory
    • Anything else?
• Find applications/algorithms that benefit
  • Should be many out there!
Collective Plans/Schedules

- can we find a better name?
- act as expert interface for advanced users or ...
- ... compilation target
- → Christian (I'll have a different interface)
More Comments/​Input?

Any items from the floor?
General comments to the WG?
Directional decisions?
Telecons are very educational/productive :)
Come and join!