Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

MPI Forum Meeting
Collectives and Topology WG
December 2012

Torsten Hoefler

On behalf of the working group

AGENDA

- Neighborhood reductions
  - Addition of skipped functionality
  - Use-cases were found
- Scalable vector collectives
  - Two interfaces proposed, delayed
- A user-proposal
  - Unknown-length collectives
- Discussion
  - Misc. items

Neighborhood Reductions

- Were proposed for MPI-3.0
  - Removed by Forum request due to missing use-case
- A *lot* of user-feedback about missing functionality
  - Jed Brown, Shirley Moore, Heike Jagode, others
  - Clear use-cases have been identified
    - See collwg mailinglist

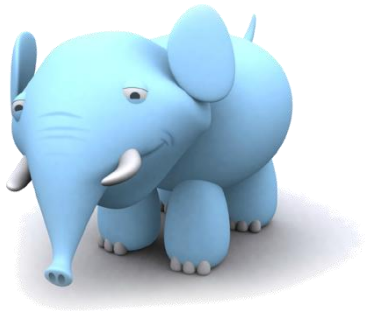
Rejuvenating the Discussion

- Blocking interface
  - topol.pdf page 33ff
- Nonblocking interface
  - topol.pdf page 41ff

- Reading & discussion …

Scalable Vector collectives

- State of the art:
  - Varying amounts of data from the processes
  - Integral part of the standard since MPI-1.0:
    - MPI_GATHERV,
    - MPI_SCATTERV,
    - MPI_ALLGATHERV,
    - MPI_ALLTOALLV,
    - MPI_ALLTOALLW (added in MPI-2.0),
    - and MPI_REDUCE_SCATTER (no ‘v’ suffix)



Scalability Problems

- Need to specify counts and displacements for each process at each process
  - Memory and time grow linearly in P
  - ➔ Well-known scalability problem
- Problems are getting worse with ever increasing system sizes
- Memory needs will eventually prevent the use of current vector collectives (cf. “Exascale”)

Are vector collectives used?

- Yes, even in libraries such as
  - ParMETIS (Parallel Graph Partitioning)
  - PBGL (Parallel Boost Graph Library)
  - PETSc (Parallel Algorithms to solve PDEs)
  - PSBLAS (Parallel Sparse Linear Algebra)
  - LibTopoMap (Topology Mapping Library)

➔ Many applications use them!

Importance of Scalable Variants

- Vector collectives are used (irregular apps)
- They work on today's machines
- It is primarily not about performance
- Current vector collectives will not work on future systems ≥ “Exascale”!
- Add-on: Scalable variants will perform more efficiently in sparse scenarios

Simple Proposal (part I)

- Four new distributed interfaces:
  - MPI_Gatherdv()
  - MPI_Scatterdv()
  - MPI_Allgatherdv()
  - MPI_Reduce_scatterdv()
- Each process specifies only the parameters for its local contribution

  e.g., int recvcounts[p] ➔ int recvcount

Simple Proposal (part II)

- Two new alltoall distributed interfaces:
  - MPI_Alltoalldv()
  - MPI_Alltoalldw()
- Each process specifies only the parameters for non-zero neighbors
  - Sparse representation (cf. topologies)
  - Still pair-wise specification assuming symmetric knowledge
- … more coming later

PROPOSAL

Ticket #264

("Scalable Variants of Vector Collectives")

See coll.pdf

User Input

- After discussion with Jeremiah Willcock
- Alltoallwl (don't discuss names!)
  - Semantics: essentially DSDE
  - Irregular communication
  - Every process knows its targets but not the sources
  - Several protocols exist (Hoefler, Moody …)

[1] PPoPP'12: Scalable Communication Protocols for Dynamic Sparse Data Exchange

Discussion

- Any other items for collective or topologies?
- Any visions for the next “big” MPI?

- Anything else?