Scalable Vector Collectives

Torsten Hoefler and Christian Siebert

On behalf of the Collectives working group

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Current Scalability Problems

- Vector collectives need to specify displacements and offsets at each rank
  - Well-known scalability problem
  - Memory consumption and time grows linearly in P
- All but Alltoall{v,w} can be fixed by distributing the arguments
  - Alltoall{v,w} needs P^2 parameters, all others only P (can be distributed)

Simple Proposal

- Distribute parameters
- New distributed versions:
  - MPI_Gathervd()
  - MPI_Scattervd()
  - MPI_Allgathervd()
  - MPI_Reduce_scattervd() (!)
- Each process specifies count and displacement for its local contribution only

Proposal

- See proposal document

Issues

- Are displacements in bytes or relative to datatypes
  - Relative to which datatype? At the root or at the specifying process?
- Do we need displacements at all?
  - Would be a departure from current model!
  - Do we have a use-case for displs?

Discussion

- Let’s assume k is the number of non-zero ranks in the call. A scalable algorithm would require that k = O(log P)
  - Why not have functions (Jesper’s comment):
  - MPI_Collate(scount, stype, rranks[], rcounts[], rtype, root, comm)
  - MPI_Allcollate(scount, stype, rranks[], rcounts[], rtype, comm)
  - MPI_Allcollateall(sranks[], scounts[], stype, rranks[], rcounts[], rtype, comm)

Issues with Collates

- Rank lists are replicated at each process!
  - Huge memory overhead, problematic for not-so-sparse (k=\Omega(log P)) communications
- Optimizations seem harder than for distributed vector collectives
  - Trees are still possible though
- Maybe we want both?
  - Comments?

Thanks!

Please review ticket 264!

Questions?