Lecture 4: Advanced MPI

MPI 4 standard: https://www.mpi-forum.org/docs/mpi-4.0/mpi40- report.pdf
MPI 3 (version 3.1) standard: https://www.mpi-forum.org/docs/mpi3.1/mpi31-report.pdf
OpenMPI documentation: https://www.open-mpi.org

Collective MPI calls

- Which is the most characteristic property of a collective call?
- Experiment with ignoring it! (e.g. MPI/Coll1.c)

Need to be called by all processes (ranks) in the communicator!

- Sketch (e.g., on paper) the data placement before and after the MPI_Gatherv call in MPI/Coll1.c!

- Replace MPI_Gather[v] by MPI_Allgather[v]!
  What else needs to be changed?

  - Remove
    if (rank==root)

- Collective computations (mainly reductions):

  Add a reduction by MPI_Reduce of your choice to MPI/Coll_1.c
  and check the result!
  Try also MPI_Allreduce!

- Creating new communicators

  Run and understand communicator splitting in MPI/Split_1.c!

  Try a meaningful modification of the key argument of MPI_Comm_split!

  Also: MPI/Comm_1.c, MPI/Split_2.c

- User-defined datatypes in MPI

  Run and understand MPI/Data_1.c!

  Modify count, blocklength and stride of MPI_Type_vector!

- What is a general datatype in MPI?

- sequence of basic datatypes (“type signature”)
  +
- sequence of integer (byte) displacements

- One-sided communication

  Clarify:

  origin vs. target process

  data source vs. data destination

  active vs. passive synchronization

- origin vs. target: active vs. passive process

- Put etc.: data source on origin, data destination on target
  Get etc.: data source on target, data destination on origin

- active sync: both origin and target involved

  passive sync: only origin

- One-sided communication:

  Run and understand MPI/One_sided_1.c !

  Also: MPI/One_sided_2.c, MPI/One_sided_3.c