Advancing the “Persistent” Working Group

MPI-4

Puri Bangalore (puri@uab.edu)
Tony Skjellum (skjellum@auburn.edu)

June 3, 2014

Persistence Goals (Updated)

- Cross-cutting use of “planned transfers”
  - Point to point
  - Collective (intercomm and intracomm)
  - Generalized requests
- Allow program to express
  - Locality, static use of resources
  - Connectivity of sends/receives
- Improve speed and scalability for regular, static communication (temporal locality)
- Cut middleware overheads (a lot) ~ Price of Portability too high for data parallel programs = $N_{½}$ like “vendor level” for repeated transfers

Original Requirements

- The minimum number of instructions in critical path to launch a send for "Nth" time after "setup"
- Arguments frozen after first time
  - All
  - Most (except starting address, length)
- Want the "Start" to track right to "kick DMA" or "kick transfer offload engine"… O(1) OS bypass / application bypass instruction
- Makes regular programs finer grain in O/H & Latency
- Provides its own communication space/order/matching (outside of *parent* communicator)
- Use all the features you can use in an MPI_Isend; simple datatypes may be faster in practice
- Easy changes to existing MPI data parallel programs

MPI-4 Requirements

- Work primarily within the MPI API as given in MPI-2-4
  - Overload tags in point-to-point
  - FIFO
  - No wildcards
  - No truncation error info after short receives
- Utilize per-communicator controls to effect special semantics
  - This is the differentiated service we talked about before
- Achieve multiple outstanding (slack) transfers
- Uses overloading of current API
- Adds "MPI_Comm_dup_differentiated"
- Optionally adds
  - MPI_Comm_create_differentiated
  - MPI_Comm_init_differentiated

Concepts review

- Drew analogies from MPI/RT
- Point-to-point Channels vs half-channels
- Slack-1 “bind a send” to “a receive”
- Goal: cut the # of instructions in critical path
- Reuses “Send_init” and “Recv_init”
- Open issues
  - Multiple slack channels
  - Rebinding when some arguments change

Concepts review

- For all non-blocking collective, we can define "persistent non-blocking collective"
- The approach in MPI/RT was to have collective channels as well as point to point
- Again, the goal is to allow for static resource management (memory, network, etc), algorithm selection in advance, and single-mode performance of operations

Differentiated Service

- Each communicator is a separate “MPI fabric” or independently ordered overlay network that is all-to-all
- Allow communicators to offer differentiated service
  -
  -
  -
  -
  -
  -
  - QoS concepts could be added as well
  - Subset of MPI that is usable
- Goal: allow each communicator fabric to operate at optimal performance and scalability and even to make error/fault choices to support FT where appropriate
- Use MPI_Comm_dup and MPI_Comm_create as prototypes for generating such functions

May 2011 Document

- We have a 12-page document from 2011 that we held back while MPI-3 was finishing
- It contains basic proposals
- We'd like to refine these and update the May 2011 document after this working group meeting today
- We'd like to flesh out any concerns not yet fully understood
- So, first a couple of slides, then we go through the document, and capture action items and ideas and next steps!

June 14 Document

- We’ve revised the proposals and added some!
- We’ll go over the document shortly!

Q&A