Mercury Discussion

Jerome Soumagne
HDF5 is a data model, library, and file format for storing and managing data. It supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data. HDF5 is portable and is extensible, allowing applications to evolve in their use of HDF5.

Virtual object layer provides the user with the HDF5 data model and API, but allows different underlying storage mechanisms:

- HDF5 (MDS & Remote plugins)
- LANL (PLFS plugin)
- Intel (IOD/DAOS-M plugin)
- Others
Mercury Layers

Level 3:
Generate code

Level 2:
Encode/decode function arguments

Level 1:
Send/recv RPC (buffer level)
Execute user RPC callback

Level 0:
Network Abstraction Layer

HG Macros

HG + HG Proc

HG Core

HG Bulk

NA

BMI

MPI

CCI

...
Recent Changes / Progress Model

- Separation of layers
  - NA less visible to users
  - NA_Initialize() / HG_Init_na()

- Moved to callback model (HG_Progress / HG_Trigger)
  - Originally based on request model (wait with timeout)
  - Request emulation API

- Unified client/server API
Main goal: recover from faults
- Non-fault cancellation may not always be supported

Which operations?
- RPC cancellation (HG_Forward and HG_Respond)
- Bulk cancellation
- Lookup cancellation?

Operation canceled = callback pushed to completion queue with canceled state when NA ops canceled

Cancellation is local only

Testing phase / CCI / Design and documentation online
• Lookup = non-blocking remote address resolution
• May involve connection or not depending on plugin
  • BMI has deferred connection
  • CCI calls connect

• Pondering idea for providing an infrastructure so that plugins can follow that model and have lookup ≠ connect
• Goal: Improve intranode communication for all plugins
• Re-use existing technology (shm/CMA)
• Transparently enabled (but can also be disabled)
• Progress on multiple NA plugins
  • Expose file descriptor for NA plugins that support it and use poll() on them
  • Otherwise may just have to busy spin
  • Latency issue?
Other improvements

• RMA: Reduce overhead from memory registration and bulk descriptor exchange
  • Bulk streaming
• Batching RPC calls
• One-way RPCs
• Make NA plugins allocate / expose memory pointers used for short messaging
• Documentation / website / github / wiki
  - http://mercury-hpc.github.io
  - https://github.com/mercury-hpc

• Testing infrastructure: travis + Cdash
  - https://travis-ci.org/mercury-hpc/mercury (cron job + continuous build)
  - https://cdash.hdfgroup.org/index.php?project=MERCURY (results from tests)

• Pull requests (easiest way to get stuff tested)

• Release milestones
  - 0.9.0 – Callback model and cancellation
  - 1.0.0 – Shared memory, etc