MPI Communicator Assertions

Jim Dinan

Point-to-Point WG

March 2015 MPI Forum Meeting

Big Picture

- Goal: Allow the application to provide hints about its behavior
  - Hints about behavior should not be propagated
  - MPI library can ignore them, but application cannot
- MPI runtime can optimize based on application's behavior
- Examples:
  - No send cancel
  - No wildcards
  - No message ordering

Info Keys

- Initially we were looking at info keys, however:

- MPI 3.0, Section 6.4.4:
  - "Hints specified via info (see Chapter 9) allow a user to provide information to direct optimization. Providing hints may enable an implementation to deliver increased performance or minimize use of system resources. However, hints do not change the semantics of any MPI interfaces."

- MPI_COMM_DUP also propagates info hints
  - Hints that make certain operations invalid could break libraries
  - E.g. if a library is passed a communicator with no_wildcards set, duplicates it, then uses wildcards on the new communicator

Can Info Keys Restrict MPI?

- No agreement within the Forum
  - Pavan, stop reading email and speak up :)
  - P2P WG was asked to develop alternate proposals
- Several RMA info keys already restrict behavior
  - E.g. no_locks
- Primary issue is propagation
  - No propagation in the RMA interface
  - We could remove info propagation in MPI_Comm_dup without breaking backward compatibility

Two New Assertions Proposals

1. New API to apply assertions to a communicator
2. Use MPI_T CVars to change configuration of a communicator

Early concepts – feedback requested

Communicator Assertions API

```
MPI_Comm_set/get_asserts(MPI_Comm comm,
                         MPI_Assert asserts)
```

- Collective call to set/get assertions on *comm*
  - Set has undefined behavior if there are operations pending on *comm*
- MPI_Assert is a dictionary (like MPI_Info)
  - Could duplicate the MPI_Info API for MPI_Assert
  - Simpler: could add MPI_Info_to/from_assert() conversion routines

MPI_T CVars

- MPI_T control variables are used to change properties and configuration settings of the MPI implementation
  - CVARs are part of the tools interface, added in MPI 3.0
  - Could also be used to change behavior of a communicator
  - Suggested by Martin Schulz

- Challenges:
  - Currently, no predefined cvars (must be queried)
  - Propagation of cvars is not defined
- Pro:
  - Less change to the API

Summary: Three Proposals

1. Use info, remove info propagation in dup()
2. Add MPI_Comm_assert
3. Use CVARs