

# MPI Communicator Assertions

Jim Dinan

Feb. 2015

# Big Picture

- Goal: Allow the application to provide assertions about its behavior
- MPI runtime can optimize based on assertions
- Examples:
  - No send cancel
  - No wildcards
  - No message ordering

# Why Not Info Keys?

- 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, which would break backward compatibility
  - E.g. if a library is passed a communicator with restricted semantics and duplicates it

# Two New Proposals

1. New API to apply a set of assertions to a communicator
2. Use MPI\_T CVars to change configuration parameters for a communicator

*Early concepts – feedback requested*

# Communicator Assertions

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

- Collective call to set/get assertions on *comm*
  - Set has undefined behavior if ops pending on *comm*
- Assert properties of application behavior on *comm*
  - Can be used to restrict MPI semantics
  - Assertions are not propagated
  - MPI library can ignore them, but application cannot
- MPI\_Assert is a dictionary (i.e. Info object)
  - Could duplication Info API for MPI\_Assert
  - Could also have MPI\_Info\_to/from\_assert()

# MPI\_T CVars

- MPI\_T control variables could 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