Other FT Tickets

FTWG Plenary
March 2015 MPI Forum Meeting

#324
Clarify MPI_ERRORS_ARE_FATAL scope of abort

- Previous text said: "*The handler, when called, causes the program to abort on all executing processes. This has the same effect as if MPI_ABORT was called by the process that invoked the handler.*"

- If a communicator's error handler is MPI_ERRORS_ARE_FATAL, when triggered, it is equivalent to MPI_Abort(comm).

- Important for handling sub-communicators and dynamic processing errors correctly

- Previous contention was about whether this would cause situations where the application was unable to do anything after a failure.

- If MPI_Send does not return an error and **then** a process fails, there is no mechanism to return an error.

    - This is invalid because once MPI_Send returns SUCCESS, the error is reported on some other operation.

- Similarly, if the user splits up MPI_COMM_WORLD and never uses it again, the error handler for MPI_COMM_WORLD should not be kicked.

#472
Clarify behavior of multiple error handlers in a single call

- The behavior of MPI_(Wait/Test)(all/some) is unclear if multiple requests need to return an error.

  - Which error handlers are called? How many times?

- We're proposing saying that at most one error handler will be called at most once and that at most one request will be marked with an error class (other than MPI_ERR_PENDING).

- The status.MPI_ERROR value will be in one of 3 states:

  - MPI_SUCCESS if the request has completed successfully

  - MPI_ERR_PENDING if the request has not completed yet

  - Some error class if the request has completed with a failure