The state of this presentation is undefined

Josh Hursey
Oak Ridge National Lab.

8.3 Error Handling

- Current Standard states that:

  After an error is detected, the state of MPI is undefined. That is, using a user-defined error handler, or MPI_ERRORS_RETURN, does not necessarily allow the user to continue to use MPI after an error is detected. The purpose of these error handlers is to allow a user to issue user-defined error messages and to take actions unrelated to MPI (such as flushing I/O buffers) before a program exits. An MPI implementation is free to allow MPI to continue after an error but is not required to do so.

  Advice to implementors. A good quality implementation will, to the greatest possible extent, circumscribe the impact of an error, so that normal processing can continue after an error handler was invoked. The implementation documentation will provide information on the possible effect of each class of errors.

Interface Portability

- “Warning” error classes, such as argument errors:
  - MPI_ERR_COUNT
  - MPI_ERR_TAG
  - MPI_ERR_RANK
- Such warnings are critical in the default case (MPI_ERRORS_ABORT) (except file open…)
- If the application wants to handle these it sets MPI_ERRORS_RETURN, but does not know if the MPI library is usable or not.

A New Error Class (Ch. 8.5)

- MPI_ERR_CANNOT_CONTINUE
  - The user requested MPI_ERRORS_RETURN (or custom)
  - The MPI library is no longer able to function properly due to a previous error.
  - The MPI library will return this value on all subsequent MPI function calls.

What does this fix?

- Portable interface behavior for applications.
- Allows applications to use MPI_ERRORS_RETURN and have well defined semantics for if it can continue to use the MPI implementation.
- A base MPI implementation will return the original error, but set a flag to ***lock out*** all other functions (similar to the current initialized checks).
  - A better version will allow MPI to continue being used for warning cases, for example.
- Backwards compatible ☺

Discussion