Update on ULFM

Fault Tolerance Working Group

MPI Forum, San Jose CA

December, 2014

Outline

- Proposal Reorg
- RMA Updates
- Dynamic Processing
- Papers
- Implementations / Testing

Reorganization

- Reorganizing ticket to split out RMA & Files
- Important Reasons
  - RMA section is still under enough flux to not be ready in the immediate future
  - Communicator section is pretty much ready to go and hasn't changed in a while (pending fixes in dynamic processing)
- Practical Reasons
  - Every time we need to make a minor change in the RMA section, we re-read the entire ticket
  - Cut down on reading times

RMA Updates

- Mostly updated section to account for locally shared memory
- Clarify state of data after failures
- Clarify state of memory when freeing window
- Purpose is to stabilize communication, not data
  - Just like communicators
  - Data can be protected using external mechanisms
    - SCR, GVR, FTI, etc.

Data Status

- After a failure, all data in a window becomes undefined
  - Implementation might do better
- **Data also becomes undefined in overlapping windows without failure**
  - Includes locally shared memory in other processes

Memory Reuse

- There are times where it may not be possible to reuse memory after a failure
  - Some networks can protect your data from late arriving messages
  - Some networks cannot
- It may not be possible to reuse memory exposed to a failure after freeing the window
  - MPI will provide an attribute in the window to let the user know if the memory can be reused

Dynamic Processing

- Reworking this section
- Need to ensure there is a way to "validate" a new communicator
- May require stronger synchronization semantics for SPAWN, CONNECT/ACCEPT, etc.

Papers/Projects

- Programming Models
  - Falanx
- Resilience Libraries
  - Fenix (SC), Message Logging (ICA3PP), LFLR (EuroMPI/ASIA)
- Applications
  - PDE Solvers (IPDPS Workshops), Multi-Level Monte Carlo (PARCO)
- Evaluation / Discussions
  - Lots and lots

Implementations

- MPICH
  - Experimental support added in v3.2a2
- Open MPI
  - Available in branch from UTK
  - Improved algorithms added recently
  - Working on bringing up to date with master branch
- Simulator
  - Developed by Christian Englemann and Thomas Naughton
  - Evaluates performance at large scale

Testing Repository

- Holds common collection of tests
- Used to validate all ULFM implementations
- Currently housed at ORNL, but will be moved to Github soon