Fault Tolerant Working Group Activities Summary

MPI Fault Tolerance Working Group

Aurelien Bouteiller and Wesley Bland

June 2013 MPI Forum Meeting

INNOVATIVE COMPUTING LABORATORY
THE UNIVERSITY of TENNESSEE

Session outline

- Fault Tolerance
  - Status and Directions for ULFM (ticket #323)
  - Status for older/other tickets
  - Discussion: future directions, other ideas?

- And now for something completely different
  - Systematic function nomenclature
  - More non-blocking interfaces

User Level Failure Mitigation

- Goal: Survive **Process failures**
  - Network errors, I/O backend errors, switch errors, etc are not covered (but their effect has been considered)
- **Restore communication capability** after failures
  - **FT activities only in response to user API calls:** Minimal impact on performance, easy to implement, yet flexible enough
  - *Not a fault tolerance model*, there are too many to standardize a particular one
  - Tailored, elaborate **recovery models are to be provided by helper libraries** that **benefit from this standardized, portable low level interface** (most useful models could be standardized as well, based on their own merits, as a convenience to users)

MPI report errors, user repairs

- **Operation centric**: defines resulting errors and state of operations interrupted by failures
  - what is important is that the **operation could not meet its spec**.
  - State and **specification of MPI unchanged after an error** is raised (more operations possible, simply raising more errors if necessary)
  - **New API** provided to restore application **global consistency** (and resolve potential deadlocks resulting from failures): MPI_Comm_agree, MPI_*_revoke
  - RMA and I/O chapters covered with similar semantics (advanced WIP)
- **User Decides**: APIs for user to **restore only the necessary communication objects**
  - MPI_Comm_shrink (if needed)
  - MPI-Dynamics is used to replace missing processors (if needed)

Resources

- Main Website: http://fault-tolerance.org
- Forum Tickets #323, 325, 326, 327, 336
  - https://svn.mpi-forum.org/trac/mpi-forum-web/ticket/323
- Mailing list: ulfm+subscribe@googlegroups.com
- Papers:
  - Post-failure recovery of MPI communication capability: Design and rationale: to appear in Int. J. of HPC and Applications (Sage)
  - An Evaluation of User-Level Failure Mitigation in MPI: http://link.springer.com/article/10.1007%2Fs00607-013-0331-3#

Implementation

- Prototype release: http://fault-tolerance.org/ulfm/downloads/
- Based off Open MPI 1.7
- Covers Communicators only
  - Fully functional (can recover)
  - Performance overhead w/o failure is negligible
  - Some recovery algorithms are naïve, but it still performs good! (see paper, or try it yourself ☺)

Ongoing, what's next?

- Communicators: text fine tuning, no semantic changes
- RMA: major upgrade
  - Jim Dinan made a full pass on the RMA section of the FT chapter
  - Checked **compliance with new MPI-3 RMA**
  - Clarified state of **non-targeted memory**
  - Reduced ambitions on continued use of damaged windows
  - **Simplified the error behavior of Win_lock** to make the implementation less intrusive
- I/O: unchanged as of now
  - It works, no problem found
  - It is a big hammer… Mohamed would like to be able to continue some operations on the file after a failure
  - Next: investigate if we can make I/O more convenient yet still robust and low cost
- Goal: presentation to the forum in december

Review and status of opened FT tickets

MPI Fault Tolerance Working Group

Aurelien Bouteiller and Wesley Bland

June 2013 MPI Forum Meeting

INNOVATIVE COMPUTING LABORATORY
THE UNIVERSITY of TENNESSEE

Executive Summary

- Continue
  - #323, 325, 326, 327, 336: **ULFM**
  - #282: Consistent **error nomenclature**
  - #324: Explicit the effect of fatal errors and the **scope of abort**)
- Reconsider
  - #306: mpiexec, **dead rank0 return code** (too complex now)
  - #277: **MPI_Add_timer**(req) (Fabrice wants to think more on it)
- Close
  - #307: **Return code for MPI_Init** (inconsistent behavior with error handlers, superseded by #323)
  - #292: **MPI_Comm_kill** (no clear use case)

Ticket #323, 325, 326, 327

- **What: ULFM semantic additions**
- *Spans:* A new chapter, minor alterations to section 2 and section 8
- *Status*: text completed, implementation completed
  - *326 (RMA) is under rewrite*
  - *Rest of the text unchanged*
- *Plan*: **Continue**
- *Who*: Aurelien, Wesley

Ticket #336

- **What: a mechanism for an application to query the MPI implementation to determine if it provides the FT features in ULFM (#323)** (some of the semantics can be provided only as non-functional stubs)**.**
- *Spans:* In Section 8.1.2 (Environmental Inquiries) we add one predefined attribute key
- *Status*: text completed, implementation not completed
- *Plan*: **Continue**
- *Who*: Josh/Aurelien

Ticket #282

- ***What*: Consistent Use of the words Error, Exception, Failure, Incorrect program, etc.**
- *Spans:* Small scale changes but everywhere, semantic neutral
- *Status*: Text change not completed
- *Plan*: **Continue**
- *Who*: Darius ?

Ticket #324

- **What: Clarify the meaning of fatal errors (which is to call MPI_Abort on the comm, or on self if no comm available)**
- *Spans: 2 sentences in section 2, a small paragraph in section 8*
- *Status*: text completed, no implementation
- *Plan*: **continue**
- *Discussion:* The ticket itself contains an interesting discussion on an unrelated issue (what error handler is called on MPI_Free'd requests), maybe open a separate ticket on this issue
- *Who*: Josh

Ticket #306

- **What: Specify what code is returned from mpiexec when rank 0 is dead**
- *Spans:* Adds a paragraph in advice to implementors on Finalize Section 8.8
- *Status*: text completed, no implementation
- *Plan*: **continue/rework**
- *Why:* The proposed definition is complicated
- *Who*: Josh

Ticket #307

- **What: Clarifies the return code of MPI_Init when this function is not successful**
- *Spans:* adds a paragraph to Section 8.7 Startup, just after the paragraph following MPI_INIT
- *Status*: text completed, no implementation
- *Plan*: **withdraw**
- *Why:* Ticket #323 supersedes this definition. Behavior of error handlers cannot be consistent at this stage
- *Who*: Josh

Ticket #292

- **What: add MPI_Comm_kill(comm, rank, info)**
- *Spans:* Adds a function
- *Status*: text completed, no implementation
- *Plan*: **withdraw/close**
- *Why:* Forum didn't like in May 2011, functionality doesn't seem to have clear, important use cases
- *Who*: Josh

Ticket #277

- **What: add MPI_Add_timer(MPI_Request req)**
- *Spans:* Adds a function, changes to Wait, waitall etc.
- *Status*: text completed, no implementation
- *Plan*: **uncertain**
- *Why:* The WG does not wish to pursue at the moment, but Fab wants to keep thinking about this.
- *Who*: Fab