Argonne NATIONAL LABORATORY

MPI Fault Tolerance (User Level Failure Mitigation)

Fault Tolerance Working Group

MPI Forum BoF

Wesley Bland

wbland@anl.gov

U.S. DEPARTMENT OF ENERGY

Goal

- Allow wide range of fault tolerance techniques by adding a minimal set of FT functions to the MPI Standard
- Not necessarily designed to be simplest to use, but feature complete
- Is designed to encourage libraries to sit on top of ULFM and provide more user-friendly semantics

What are we trying to solve?

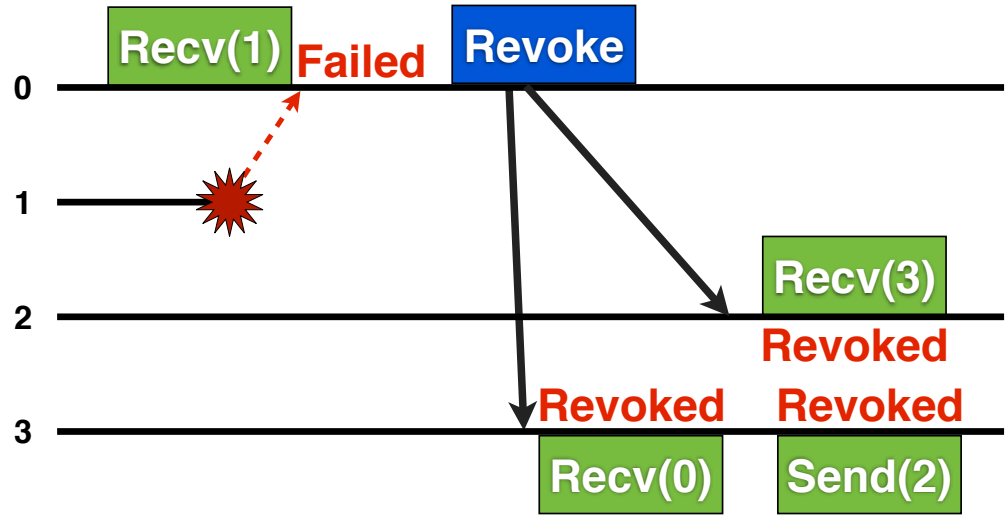
- Failure Notification
- Failure Propagation
- Failure Recovery
  - Point-to-point
  - Wildcard
  - Communicator

How do we propose to solve those issues?

- Failure Notification
  - Error codes
  - New API for getting group of failed processes
- Failure Propagation
  - Local notification
  - New API for notifying other processes
- Failure Recovery
  - Point-to-point
    - Nothing required
  - Wildcard
    - New API to re-enable MPI_ANY_SOURCE
  - Communicator
    - New API to create communicator without failed processes

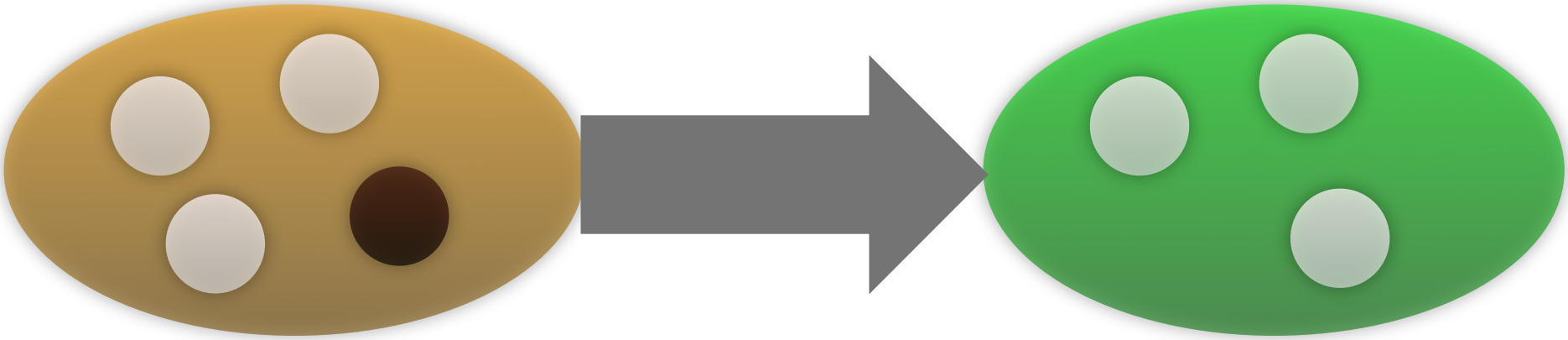
MPI_REVOKE

- MPI_Comm_revoke(MPI_Comm comm)
  - Disables further (non-local) use of comm
  - Local function
  - Global effect (over *comm*)
  - Other processes receive MPI_ERR_REVOKED



MPI_SHRINK

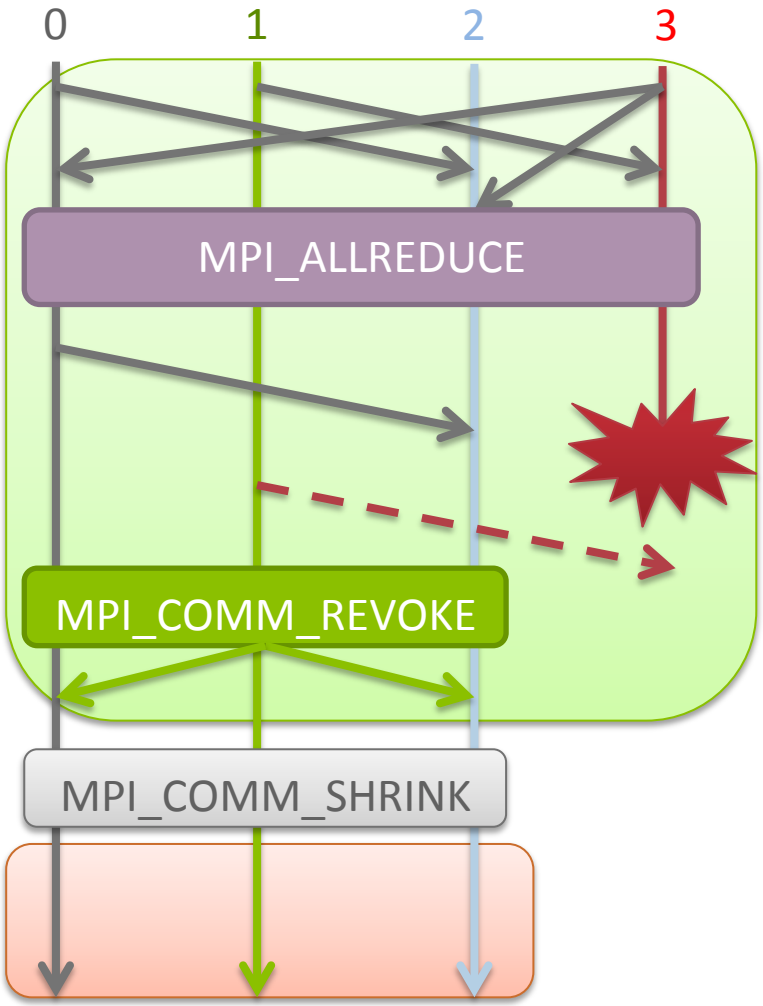
- int MPI_Comm_shrink(MPI_Comm comm, MPI_Comm* newcomm)
- Creates a new communicator with all of the processes from the original communicator
  - Excludes failed processes
- Old communicator is still around and usable (if not revoked)



MPI_COMM_AGREE

- MPI_Comm_agree(MPI_Comm comm, int *flag)
  - Performs fault tolerant agreement algorithm over the boolean *flag*
  - Ignores all failed processes
  - **Propagates the MPI_ERR_REVOKED error code if it was called before/during the agreement**
    - Meaning that all processes will receive MPI_ERR_REVOKED if it was called early enough rather than getting a correct agreement
  - Useful for determining success/failure of previous operations where agreement is necessary
  - Expensive and should be used sparingly

Recovery with Revoke/Shrink ABFT Example



- ABFT Style application
- Iterations with reductions
- After failure, revoke communicator
  - Remaining processes shrink to form new communicator
- Continue with fewer processes after repairing data
  - Still possible (not hard) to bring in a replacement process
  - Rebuild communicators with the failed process

Implementations

- MPICH
  - Will be released as part of MPICH 3.2
  - Most features available now in MPICH 3.2a2
  - www.mpich.org
- Open MPI
  - Available as a branch developed at The University of Tennessee
  - www.fault-tolerance.org

Beyond ULFM

- Reinit (LLNL)
  - Provide model that targets BSP applications
  - MPI rolls back to a safe state (such as MPI_Init)
  - All user communicators are destroyed
  - MPI automatically does everything itself (detection, recovery, notification)
- FA-MPI (Auburn)
  - Try/catch semantics
  - Transactional model
  - Timeouts used for error detection