INNOVATIVE COMPUTING LABORATORY

ULFM RESILIENCE EXTENSIONS FOR MPI

User Level Failure Mitigation is a set of MPI interface extensions enabling Message Passing programs to restore MPI communication capabilities affected by process failures. It supports rebuilding communicators, RMA windows and I/O Files.

FLEXIBILITY

- No particular recovery model is imposed or favored. Instead, a set of versatile APIs is included that provides support for different recovery styles (checkpoint, ABFT, iterative, Master-Worker, etc.).
- Application directs the recovery, it pays only for the level of protection it needs.
- Recovery can be restricted to a subgroup, preserving scalability and easing the composition of libraries.

PERFORMANCE

- Protective actions are outside of critical MPI routines.
- MPI implementors can uphold communication, collective, one-sided and I/O management algorithms unmodified.
- Encourages programs to be reactive to failures, cost manifests only at recovery.

PRODUCTIVITY

- Backward compatible with legacy, fragile applications.
- Simple and familiar concepts to repair MPI.
- Portability guaranteed by standardization.
- Provides key MPI concepts to enable FT support from library, runtime and language extensions.

The ULFM specification is a crucial infrastructure that will enable the deployment of advanced, production quality fault tolerant techniques; it is a versatile solution to improve the efficiency of novel and established techniques.

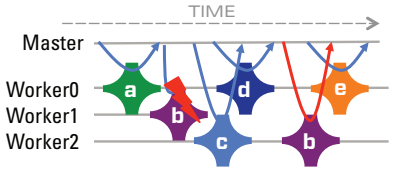
Coordinated Checkpoint/Restart, Automatic, Compiler Assisted, User-driven Checkpointing, etc.

In-place restart (i.e., without disposing of non-failed processes) accelerates recovery, permits in-memory checkpoint

(FTI, SCR, FMI)

Naturally Fault Tolerant Applications, Master-Worker, Domain Decomposition, Containment Domain, etc.

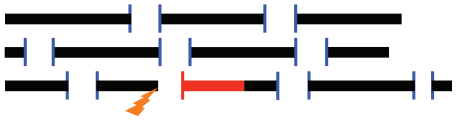
Application continues a simple communication pattern, ignoring failures



ULFM MPI Specification

Uncoordinated Checkpoint/Restart, Transactional FT, Migration, Replication, etc.

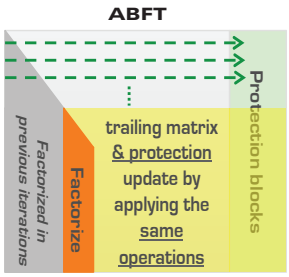
ULFM makes these approaches portable across MPI implementations



Algorithm Fault Tolerance

ULFM allows for the deployment of ultra-scalable, algorithm specific FT techniques.

(QR, LU factorizations, CG decomposition, etc.)



ICL UT INNOVATIVE COMPUTING LABORATORY THE UNIVERSITY of TENNESSEE

SPONSORED BY RIKEN U.S. DEPARTMENT OF ENERGY National Science Foundation

CITR CENTER FOR INFORMATION TECHNOLOGY RESEARCH

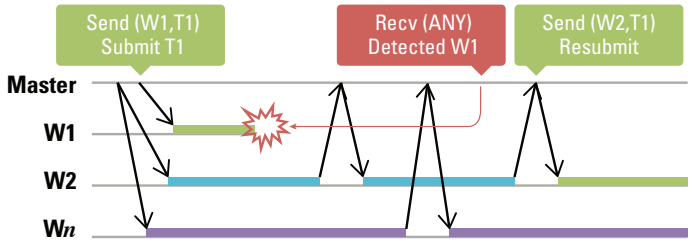
FIND OUT MORE AT http://fault-tolerance.org/

Resilience Extensions for MPI: ULFM

ULFM provides targeted interfaces to empower recovery strategies with adequate options to restore communication capabilities and global consistency, at the necessary levels only.

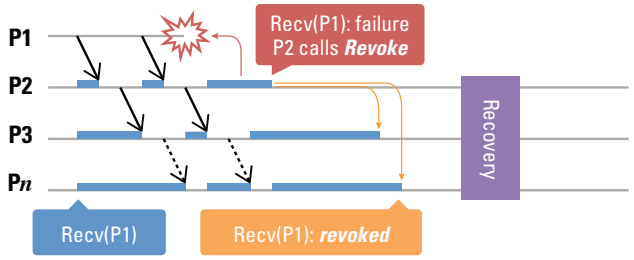
CONTINUE ACROSS ERRORS

In ULFM, failures do not alter the state of MPI communicators. Point-to-point operations can continue undisturbed between non-faulty processes. ULFM imposes no recovery cost on simple communication patterns that can proceed despite failures.



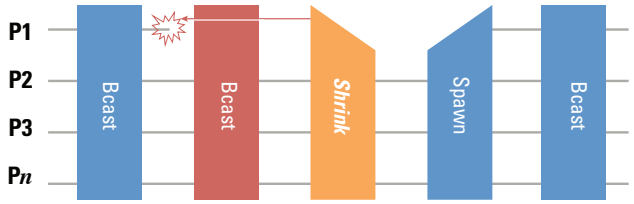
EXCEPTIONS IN CONTAINED DOMAINS

Consistent reporting of failures would add an unacceptable performance penalty. In ULFM, errors are raised only at ranks where an operation is disrupted; other ranks may still complete their operations. A process can use MPI_[Comm,Win,File]_revoke to propagate an error notification on the entire group, and could, for example, interrupt other ranks to join a coordinated recovery.

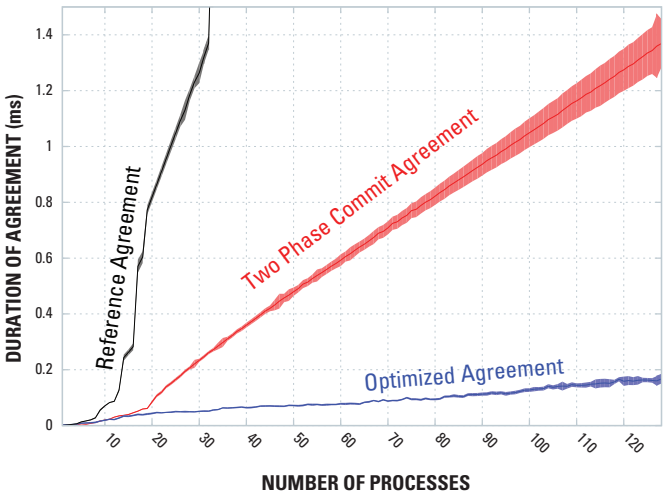


FULL-CAPABILITY RECOVERY

Allowing collective operations to operate on damaged MPI objects (Communicators, RMA windows or Files) would incur unacceptable overhead. The MPI_Comm_shrink routine builds a replacement communicator, excluding failed processes, which can be used to resume collective communications, spawn replacement processes, and rebuild RMA Windows and Files.

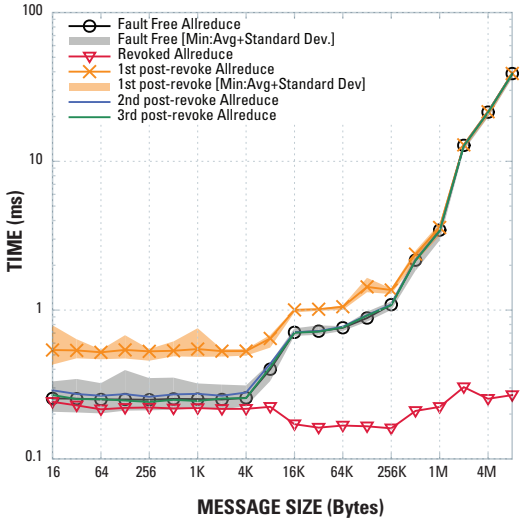


OPEN MPI ULFM IMPLEMENTATION PERFORMANCE



Novel MPI_COMM_AGREE algorithms developed at UTK have significantly better time-to-completion than the state-of-the-art two-phase commit agreement, and remains scalable with the increase in size of the underlying communicator.

Revoke Time and Perturbation in Allreduce (np=128, IB20G)



Time for a Revoke notification to propagate to all participants during AllReduce. As Revoke is asynchronous some perturbation on the subsequent communications is expected. The figure shows that this perturbation lasts for approximatively a small AllReduce, and then the communications return to nominal.