User Level Failure Mitigation is a set of MPI extensions to report errors, provide interfaces to stabilize the distributed state, and restore the communication capabilities in applications affected by process failures. Relevant communicators, RMA windows and I/O files can be reconstructed online, without restarting the application, as required by the user recovery strategy.

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 capability of restoring communication after a fault is a crucial infrastructure to support the design and deployment of production-grade recovery strategies. Multiple applications and programming frameworks are already taking advantage of ULFM constructs to deliver varied fault tolerance strategies, from run-through algorithms that continue without rejuvenating the lost processes, to methods that restore the lost processes and their dataset, either from checkpoints, or from checkpoint-free forward recovery techniques.

ULFM IN FAULT TOLERANT FRAMEWORKS AND APPLICATIONS
FEATURED APPLICATION: S3D AND FENIX

S3D is a highly parallel method-of-lines solver for PDEs and is used to perform first-principles-based direct numerical simulations of turbulent combustion. Fenix is a framework implemented above ULFM that enables online and transparent recovery from process, node, blade, and cabinet failures for parallel applications in an efficient and scalable manner. An implementation of S3D over Fenix using ULFM demonstrates order of magnitude performance improvements compared to the state of the art.

FRAMEWORKS USING ULFM
LFLR, FENIX, FTLA, Falanx, X10, CoArray Fortran

RESEARCH CENTERS USING ULFM
SNL, ANL, ORNL, UUIC, Rutgers, UTK, INRIA (FR), ETH Zurich (CH), ANU (AU), Queens U (CA), NUDT (China), Brunel U. (UK), Kyonggi U. (SK), Riken (JP), U. Tokyo (JP), AIST JP

TUTORIAL
“Fault Tolerance for HPC: Theory and Practice”
Monday Nov. 14th 8:30am - 5pm, Room 250-F

PAPER
“Failure Detection and Propagation in HPC systems”
Tuesday Nov. 15th, Room 359-D, Best Paper Finalists

Image courtesy of the authors: M.Gamell, D.Katz, H.Kolla, J.Chen, S.Klasky, and M.Parashar, “Exploring automatic, online failure recovery for scientific applications at extreme scales”, In Proceedings of SC’14
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_shrink to rebuild an entire communicator, excluding failed processes, which can be used to resume collective communications, spawn replacement processes, and rebuild RMA Windows and Files.

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.

ULFM-1.1 RELEASED
Taking into account user’s feedback, the ULFM implementation has been improved, increasing its scalability and reliability, and reducing the overheads of all fault-tolerance operations.

FUNCTIONALITY COVERAGE
Support for non-blocking version of the agreement MPI_Comm_Iagree
Compliance with the latest ULFM specification draft
Full support of all resilient concepts on intercommunicators

PERFORMANCE IMPROVEMENT
New logarithmic algorithm to perform agreement
New algorithm to perform communicator revocation (see our paper at EuroMPI’15, “Plan B: Interruption of Ongoing MPI Operations to Support Failure Recovery”)
Faster algorithm for Context ID allocation, allowing a better scalability of communicators creation and recovery

RELIABILITY IMPROVEMENT
Improved support of basic network layer (TCP, Shared Memory)
Added support for High-Performance Networks (Open IB, uGNI)
Tuned collective module enabled by default, exhibiting performance similar to default Open MPI
Runtime integration (PBS/ALPS)