MPICH

Performance & Portability

Argonne NATIONAL LABORATORY

UChicago ▶ Argonne LLC

MPICH is a high-performance and widely portable implementation of the Message Passing Interface (MPI) standard.

- MPICH-derived implementations are **used exclusively on nine of the world's top 10 supercomputers** (June 2014 ranking), including the world's fastest supercomputer: Tianhe-2
- Proudly funded by DOE for 22 years
- R&D 100 award (2005)

Goals of MPICH

- Provide an MPI implementation that **efficiently supports different platforms** including commodity clusters (desktops, shared-memory systems, multicores), high-speed networks (10 Gigabit Ethernet, InfiniBand) and proprietary high-end computing systems (IBM Blue Gene, Cray)
- **Enable cutting-edge research** in MPI through an easy-to-extend modular framework for other derived implementations.

POC: Pavan Balaji <balaji@anl.gov>, Rajeev Thakur <thakur@anl.gov>

MPICH Application Binary Interface (ABI) Compatibility Initiative

(http://www.mpich.org/abi/)

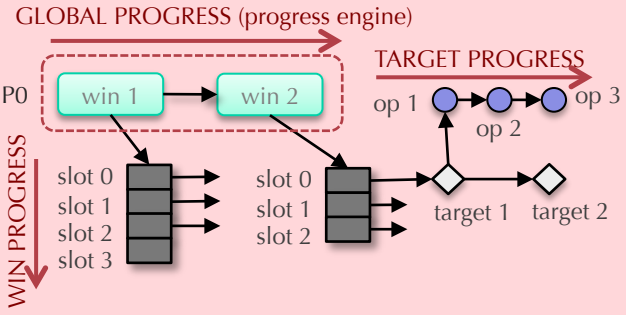
- ABI compatibility allows programs to conform to the same set of runtime conventions.
- Aim of the initiative: all parties to agree on a schedule for necessary ABI changes, leading to a **more stable release cycle and fewer surprises for developers.**
- Several notable MPICH-derived MPI implementations have begun a collaboration with the explicit goal of maintaining ABI compatibility between their implementations. The collaborators and package release dates include:
  - MPICH v3.1.3 (October 2014)
  - IBM MPI v2.1 (December 2014)
  - Intel MPI v5.0 (June 2014)
  - Cray MPT v7.0.0 (June 2014)

POC: Ken Raffenetti <raffenet@mcs.anl.gov>

Coming in MPICH 3.2

Improved RMA Infrastructure

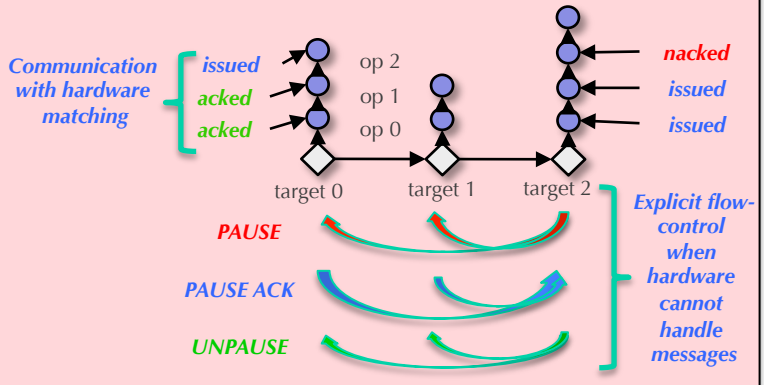
The RMA infrastructure has been rewritten from the ground up for MPICH 3.2, providing better scalability and performance (up to 50%). MPICH now better takes advantage of existing hardware to accelerate data movement and uses software as a fallback.



POC: Pavan Balaji <balaji@anl.gov>

Support for Portals4

Support for Portals4 networks in MPICH has undergone significant improvements in MPICH 3.2. Improvements include support for large messages, support for MPI one-sided operations, recovery from flow control events, and increased stability.



POC: Ken Raffenetti <raffenet@mcs.anl.gov>

Other New Features

- ✓ Fortran 2008 Bindings
- ✓ Improved Support for Intel MIC
- ✓ MXM / HCOLL Support
- ✓ New Fault Tolerance API
- ✓ Support for Upcoming MPI 3.1 Standard (Including Nonblocking Collective I/O APIs)

MPICH Partners

Cray, IBM, Intel, Lenovo, Mellanox, Microsoft, NUDT, Ohio State University, University of Tokyo, and many others…

Visit us at http://www.mpich.org

Come join us at the MPICH BoF! Tues 5:30 in Room 386-7

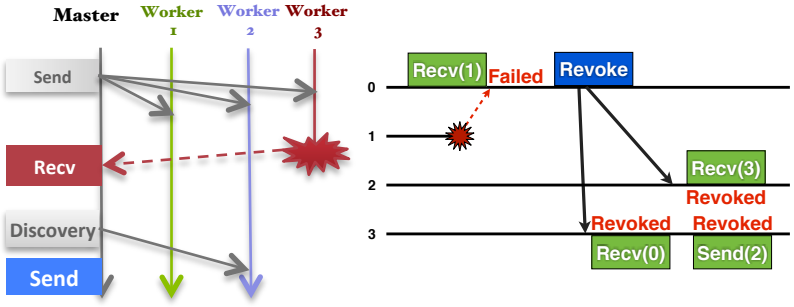
MPICH

Enabling Cutting-edge Research

Argonne NATIONAL LABORATORY — UChicago Argonne LLC

ULFM: User Level Failure Mitigation

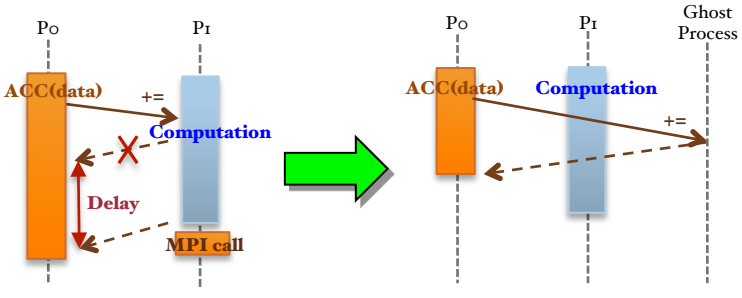
ULFM is the fault tolerance proposal for the MPI 4.0 Standard. It includes new APIs to handle process failures within an application. MPICH provides an implementation of the proposed specification as an experimental feature for users and vendors. It enables new forms of fault tolerance to be built into applications and libraries looking to prepare for exascale resilience. *(Coming in MPICH-3.2)*



POC: Wesley Bland <wbland@anl.gov>

Casper: Process-based Async. Progress for MPI RMA

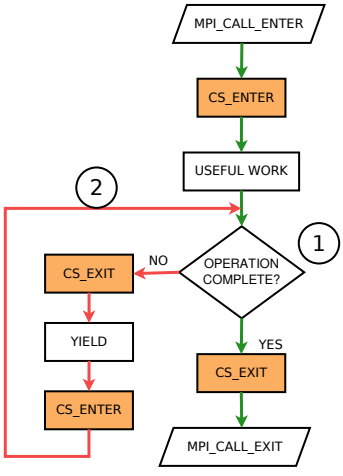
MPI RMA communication is not always truly one-sided. On RDMA supported platforms (e.g., IB, Cray), some ops such as noncontiguous accumulates still have to be done in software (they need MPI calls to make progress), resulting in long delays if the target process is busy computing outside the MPI stack. Casper dedicates arbitrary number of ghost processes on multi- and many-core architectures. It utilizes MPI-3 shared memory windows to map memory from multiple user processes into the address space of ghost processes.



POC: Antonio Peña <apenya@anl.gov>

MPI + Threads: Runtime Contention and Remedies

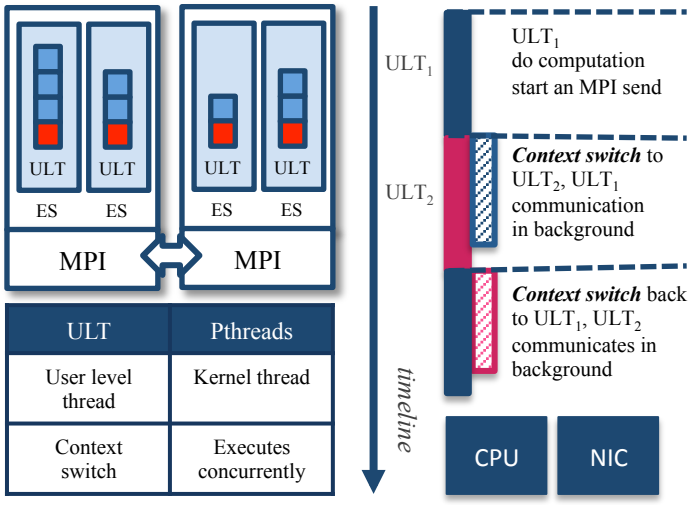
General methods for thread-safety, e.g. mutexes, incur resource monopolization and do not adapt well to the workload of an MPI runtime. By eliminating resource monopolization using fair scheduling, threads progress was much improved. We further refine the scheduling to promote threads likely to yield more useful work. This is achieved by giving higher priority to threads in the main execution path (1) and lowering the priority of those polling for progress (2). This allows to feed the runtime with more work and to reduce wasted resource acquisitions. *(to appear in PPoPP'15)*



POC: Huiwei Lu <huiweilu@anl.gov>

MPICH and User-level Threads

In this work, we build a hybrid runtime that integrates user-level threads (ULTs), such as Argobots and qthreads, with MPICH. ULTs are used as the fundamental unit of computation and communication, tightly integrated with the scheduler of MPICH. MPICH with ULTs can provide more opportunities for computation and communication overlapping.



POC: Huiwei Lu <huiweilu@anl.gov>, Sangmin Seo <sseo@anl.gov>

MPICH Related Events at SC'14

Papers	Mon / 04:10pm - 04:40pm / 286-7 / *Simplifying the Recovery Model of User-Level Failure Mitigation* Wed / 10:30am - 11:00am / 393-4-5 / *Nonblocking Epochs in MPI One-Sided Communication* (Best Paper Finalist) Wed / 11:30am - 12:00pm / 393-4-5 / *MC-Checker: Detecting Memory Consistency Errors in MPI One-Sided Applications*
Poster	Tue / 05:15pm - 07:00pm / Lobby / *Using Global View Resilience (GVR) to add Resilience to Exascale Applications* (Best Poster Finalist)
BoFs	Tue / 05:30pm - 07:00pm / 386-7 / *MPICH: A High-Performance Open-Source MPI Implementation* Wed / 05:30pm - 07:00pm / 293 / *The Message Passing Interface : MPI 3.1 and Plans for MPI 4.0*
Tutorials	Mon / 08:30am - 05:00pm / 389 / *Advanced MPI Programming,* by Pavan Balaji, William Gropp, Torsten Hoefler, Rajeev Thakur Mon / 08:30am - 05:00pm / 386-7 / *Parallel I/O In Practice,* by Robert J. Latham, Robert Ross, Brent Welch, Katie Antypas
Demos	Tue / 04:20pm - 05:00pm / UTK/NICS Booth #2925 / *Argo Runtime for Massive Concurrency* Wed / 11:00am - 01:00pm / DOE Booth #1939 / *ARGO: An Exascale Operating System and Runtime*

MPICH mailing lists: Questions: discuss@mpich.org; Announcements: announce@mpich.org