Open MPI State of the Union XI
Community Meeting SC18

Jeff Squyres

George Bosilca

Edgar Gabriel

Josh Ladd

Interactive / Online / SC thingy

- Online question topic submission: Linklings
- BOF feedback form

https://www.open-mpi.org/sc18/


EuroMPI'19
September 11-13 2019
Zurich, Switzerland
https://eurompi19.inf.ethz.ch
Important dates:
Submission server opens: January 14th, 2019
Full paper submission: April 15th, 2019 (AOE)
Notification: July 1st, 2019
Camera-ready: August 5th, 2019

Open MPI versioning

Quick review

Open MPI versioning

- Open MPI uses “A.B.C” version number triple
- Each number has a specific meaning:

A This number changes when backwards compatibility breaks

B This number changes when new features are added

C This number changes for all other releases

Definition

- Open MPI vY is ***backwards compatible*** with Open MPI vX (where Y>X) if:
  - Users can compile a correct MPI / OSHMEM program with vX
  - Run it with the same CLI options and MCA parameters using vX or vY
  - The job executes correctly

What does that encompass?

- “Backwards compatibility” covers several areas:
  - Binary compatibility, specifically the MPI / OSHMEM API ABI
  - MPI / OSHMEM run time system
  - `mpirun` / `oshrun` CLI options
  - MCA parameter names / values / meanings

What does that <u>*not*</u> encompass?

- Open MPI only supports running exactly the same version of the runtime and MPI / OSHMEM libraries in a single job
  - If you mix-n-match vX and vY in a single job…

Version Roadmaps

v2.1.x *(End of Life)*

v3.0.x (Prior stable)

- Release managers
  - Brian Barrett, AWS
  - Howard Pritchard,
    Los Alamos National Lab


- Current release: v3.0.3
  - October 29, 2018
  - v3.0.4 expected Q1’19
- Maintenance mode
  - No new features for life of series
- Major features
  - MPI_THREAD_MULTIPLE support by default

v3.1.x (Prior stable)

- Release managers
  - Brian Barrett, AWS
  - Jeff Squyres, Cisco

- Current release: v3.1.3
  - October 29, 2018
  - v3.1.4 expected Q1'19
- Maintenance mode
  - No new features for life of series
- Many usability features over 3.0.x

v4.0.0 just released!

v4.0.x (Current stable)

- Release managers
  - Howard Pritchard,
    Los Alamos National Lab
  - Geoff Paulsen, IBM

- Lots of bug fixes and performance improvements
- Big changes:
  1. Removed MPI-1 APIs not prototyped in mpi.h by default
  2. IB support now via UCX
  3. ABI compatible with 3.x
  4. MPIR usage deprecated

PSA: Stop using MPI-1 removed APIs!

- MPI_ADDRESS
- MPI_ERRHANDLER_CREATE
- MPI_ERRHANDLER_GET
- MPI_ERRHANDLER_SET
- MPI_TYPE_EXTENT
- MPI_TYPE_HINDEXED
- MPI_TYPE_HVECTOR
- MPI_TYPE_STRUCT
- MPI_TYPE_LB
- MPI_TYPE_UB
- MPI_UB
- MPI_LB
- MPI_COMBINER_HINDEXED_INTEGER
- MPI_COMBINER_HVECTOR_INTEGER
- MPI_COMBINER_STRUCT_INTEGER
- MPI_HANDLER_FUNCTION

- All of these were:
  - Deprecated in MPI-2.0 in 1996
  - Removed in MPI-3.0 in 2012
- *All have easy replacements*
  - See “Removed MPI constructs” FAQ category
    - open-mpi.org/faq/
  - Includes code samples showing how to update your code

PSA: Stop using MPI-1 removed APIs!

- MPI_ADDRESS
- MPI_ERRHANDLER_CREATE
- MPI_ERRHANDLER_GET
- MPI_ERRHANDLER_SET
- MPI_TYPE_EXTENT
- MPI_TYPE_HINDEXED
- MPI_TYPE_HVECTOR
- MPI_TYPE_STRUCT
- MPI_TYPE_LB
- MPI_TYPE_UB
- MPI_UB
- MPI_LB
- MPI_COMBINER_HINDEXED_INTEGER
- MPI_COMBINER_HVECTOR_INTEGER
- MPI_COMBINER_STRUCT_INTEGER
- MPI_HANDLER_FUNCTION

- **NOT PROTOYPED IN v4.0.x** `mpi.h` **BY DEFAULT**
  - Applications using these removed symbols will fail to compile
  - The symbols are in libmpi, however (so ABI is preserved)
- Can use `--enable-mpi1-compatibility` to restore the removed `mpi.h` prototypes
  - This CLI option, prototypes, and symbols will exist for all v4.0.x releases
  - *…but may disappear in a future Open MPI release*

InfiniBand support → UCX PML

- OpenUCX (openucx.org) is now the preferred method for InfiniBand support
  - You may need to download/install OpenUCX before installing Open MPI
- By default, the openib BTL will refuse to run on IB devices
  - Unless manually enabled by setting the MCA param `btl_openib_allow_ib` to 1

v4.0.x: RoCE / iWARP → openib BTL

- RoCE and iWARP devices still default to the openib BTL
  - Can force the use of the UCX PML for RoCE/iWARP:
  - `mpirun --mca pml ucx --mca osc ucx …`
- RoCE and iWARP will likely default to UCX in a future release

REMINDER

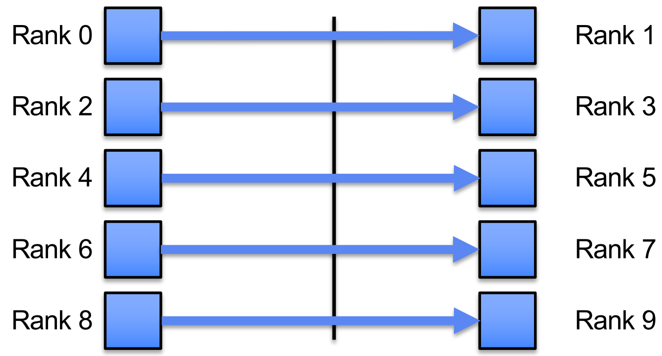
Deprecation notice: MPIR

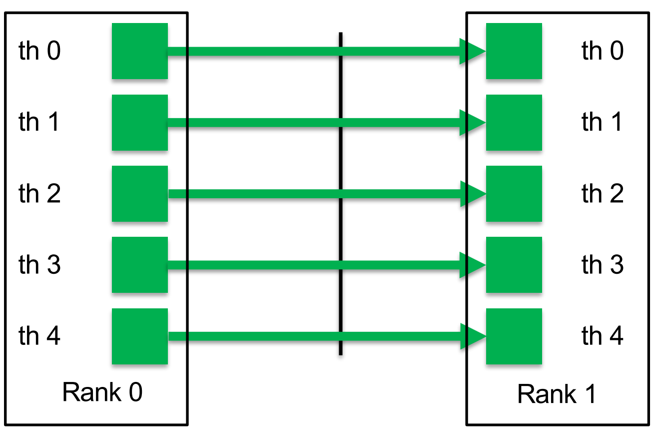
- MPIR interface is used internally to launch / attach tools and debuggers
- The maintainer for Open MPI's MPIR is retiring!
- Initially announced at SC'17 BOF:
  - Unless someone else takes over, this is the plan:
    - Deprecation notice in NEWS in early CY2018
    - User runtime warnings in mid/late CY2018 (v4.0.0)
    - Removal in CY2019

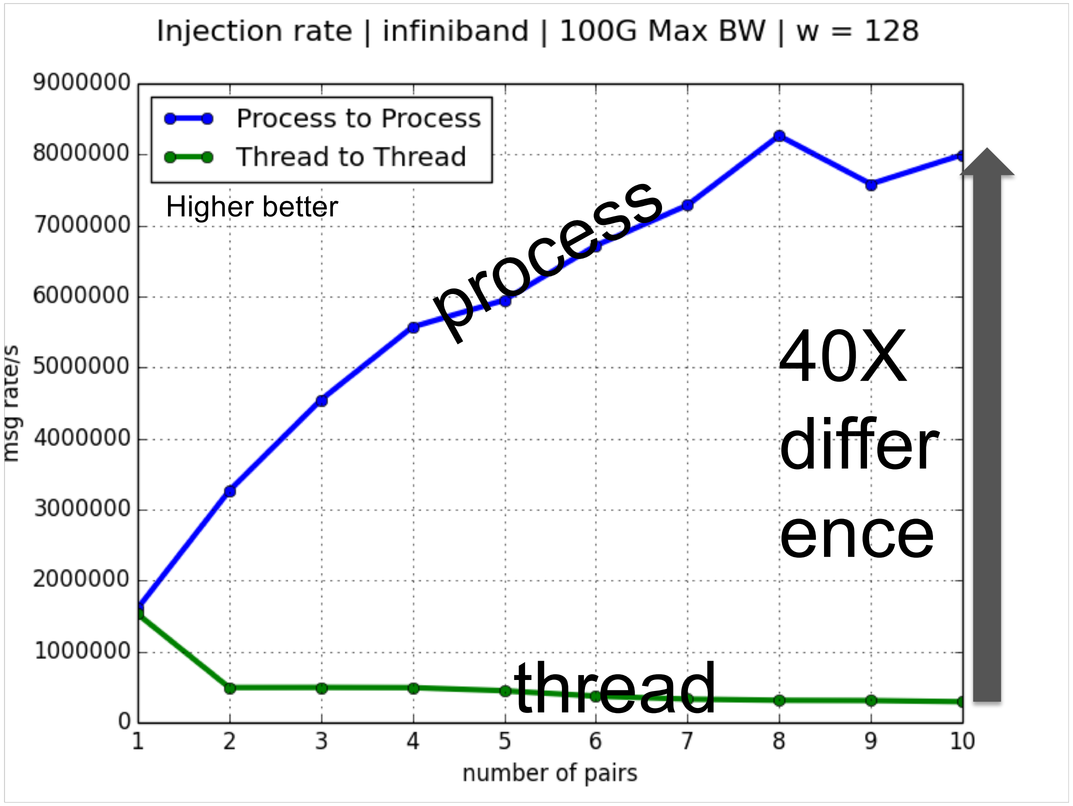
Threading, Collectives, Tools, Resilience

George Bosilca
University of Tennessee

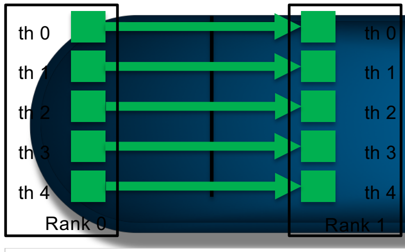
Threading support

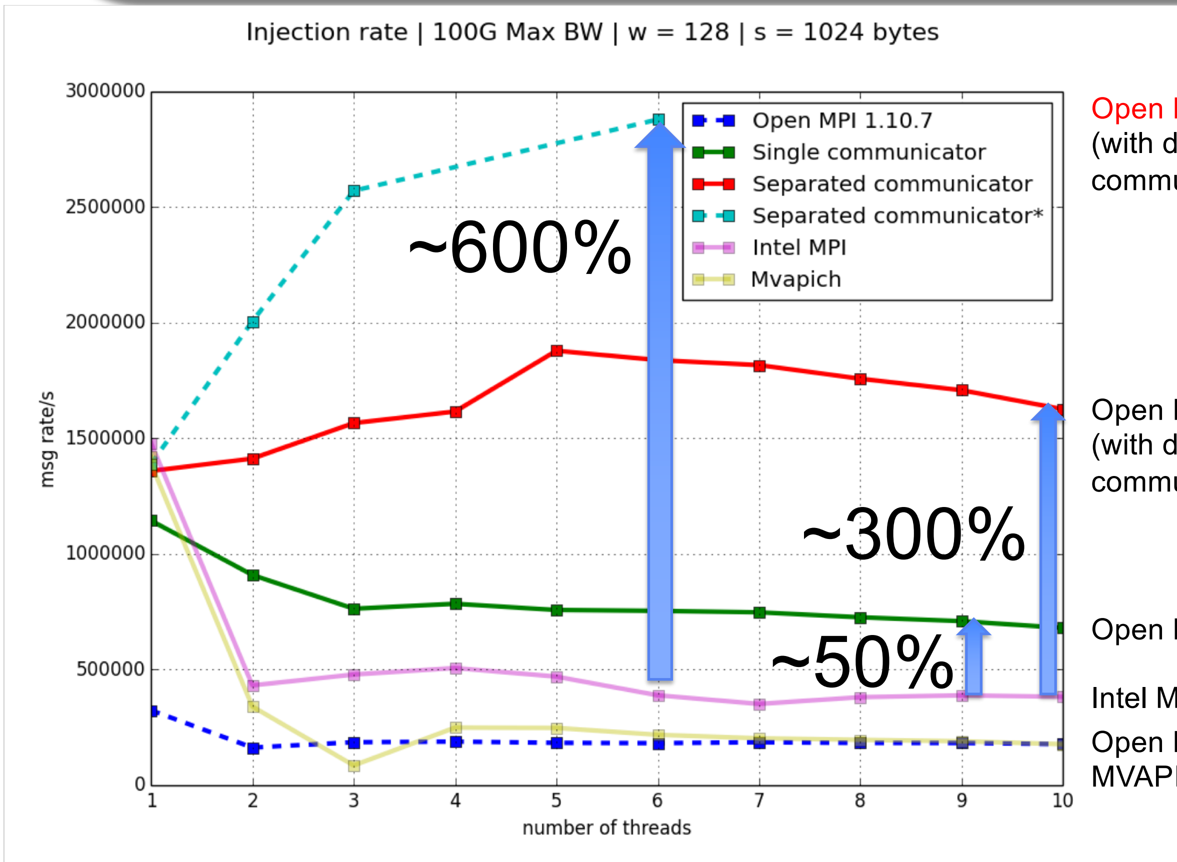






Threading support





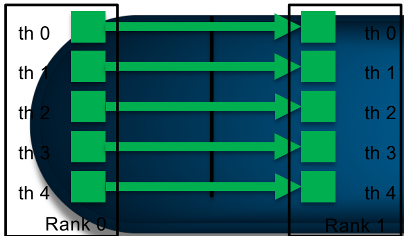
Open MPI 4.1
(with different
communicators)

Open MPI 4.0
(with different
communicators)

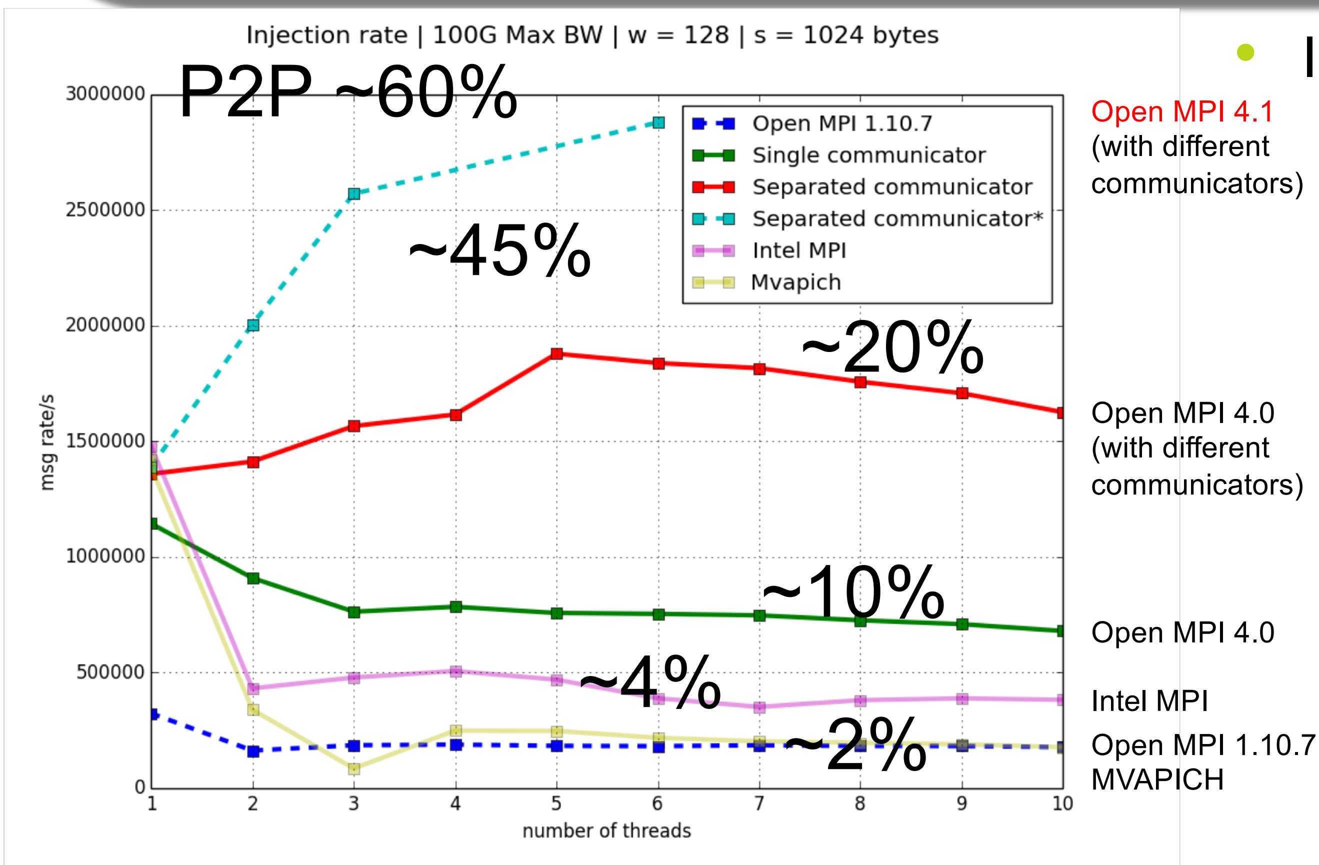
Open MPI 4.0

Intel MPI

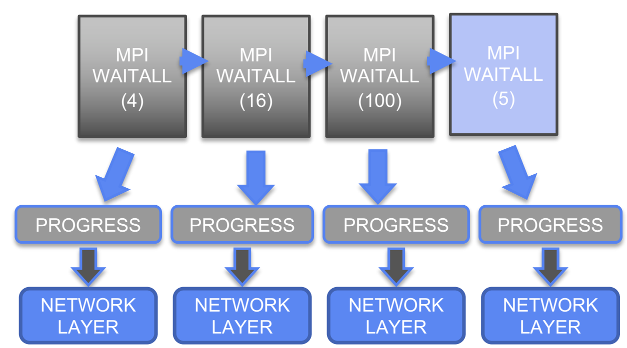
Open MPI 1.10.7
MVAPICH



Threading support

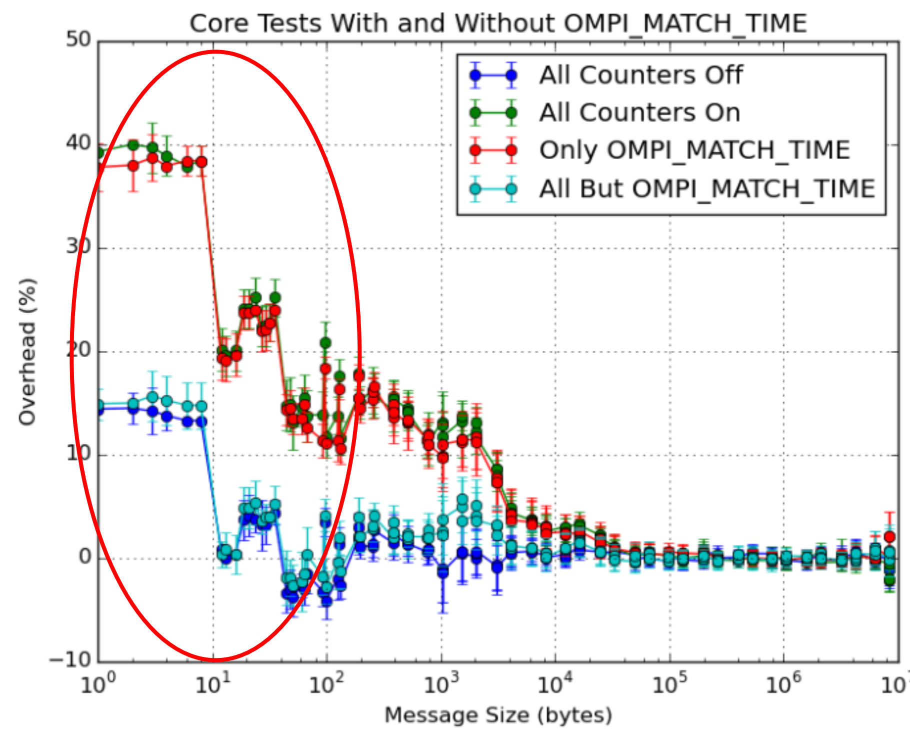


- Improvements:
  - Synchronization primitive
  - Unrestricted progress (protections done at the lowest level)
  - Credit management
  - Requests memory management
  - Out-of-sequence management (limited bypass)

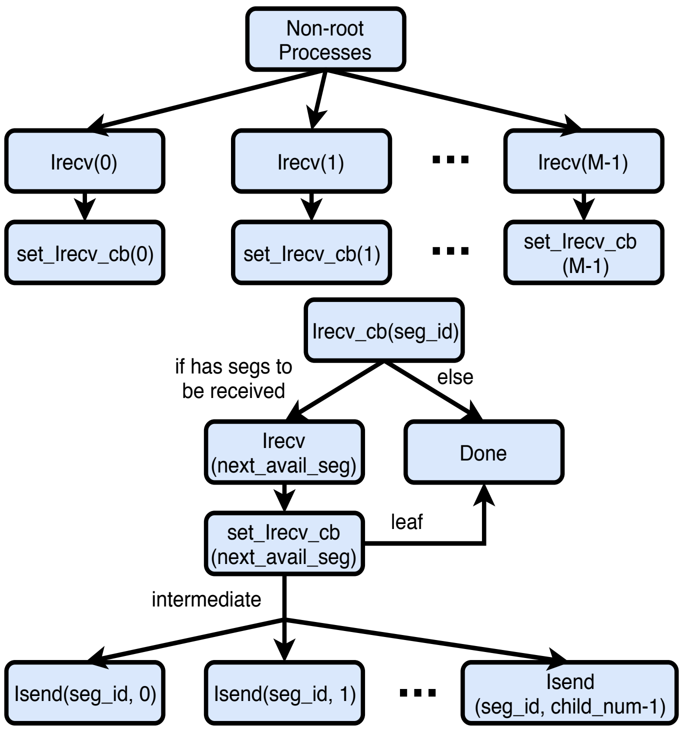


SPC: Software Performance Counters

- Similar to PAPI counters but exposing internal information not available through other means
  - Out-of-sequence messages, time to match, number of unexpected, instant bandwidth, collective bins
- Can be accessed via MPI_T, PAPI SDE, or shared file via PMIx plugins



Collective Communication

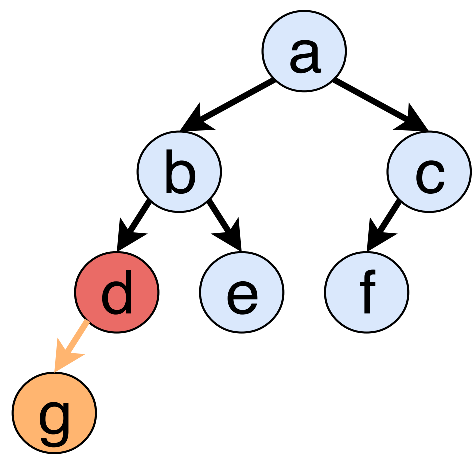


Data Dependency:

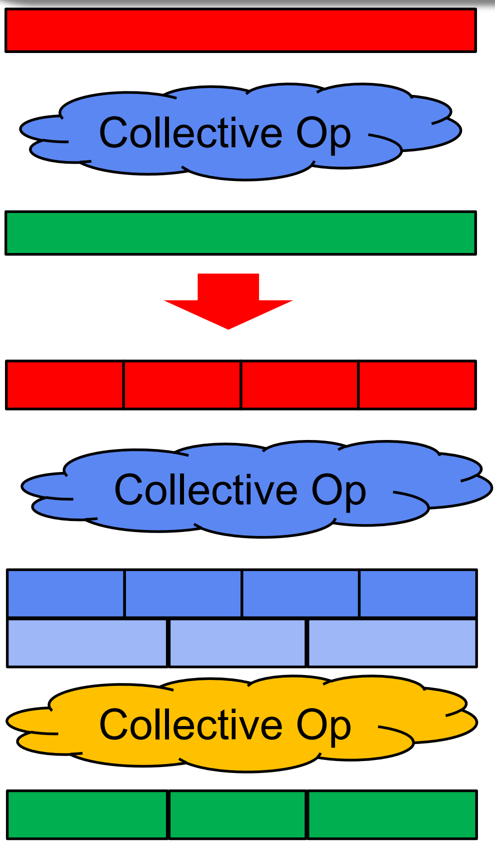
- same as previous implementation.

Synchronization Dependency:

- Segment independence
  - Rebalance
  - Decouple receiving of next segment and sending of current segment
- Child independence
  - Decouple the data transfer from different children



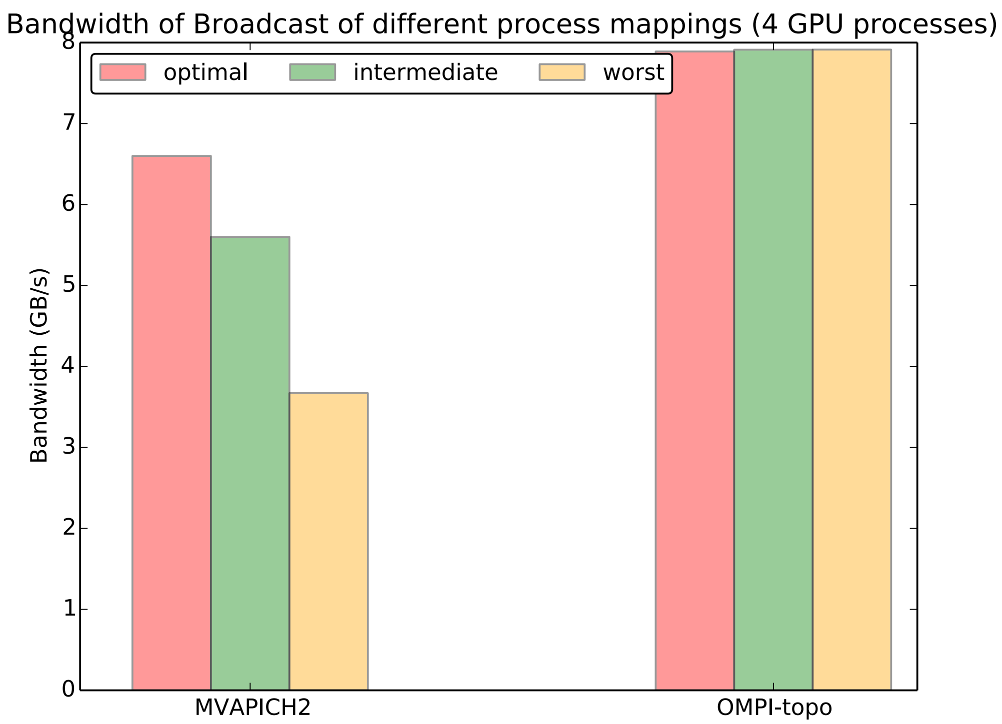
Collective communications



- Dataflow collective: different algorithms compose naturally (using a dynamic granularity for the pipelining fragments)
- Architecture aware: Each level reshape tuned collective to account for architecture capabilities
- The algorithm automatically adapts to network conditions
- Resistant to system noise

Collective Communication

Process location
Noise Reduction
Shared Memory
Hybrid Architecture



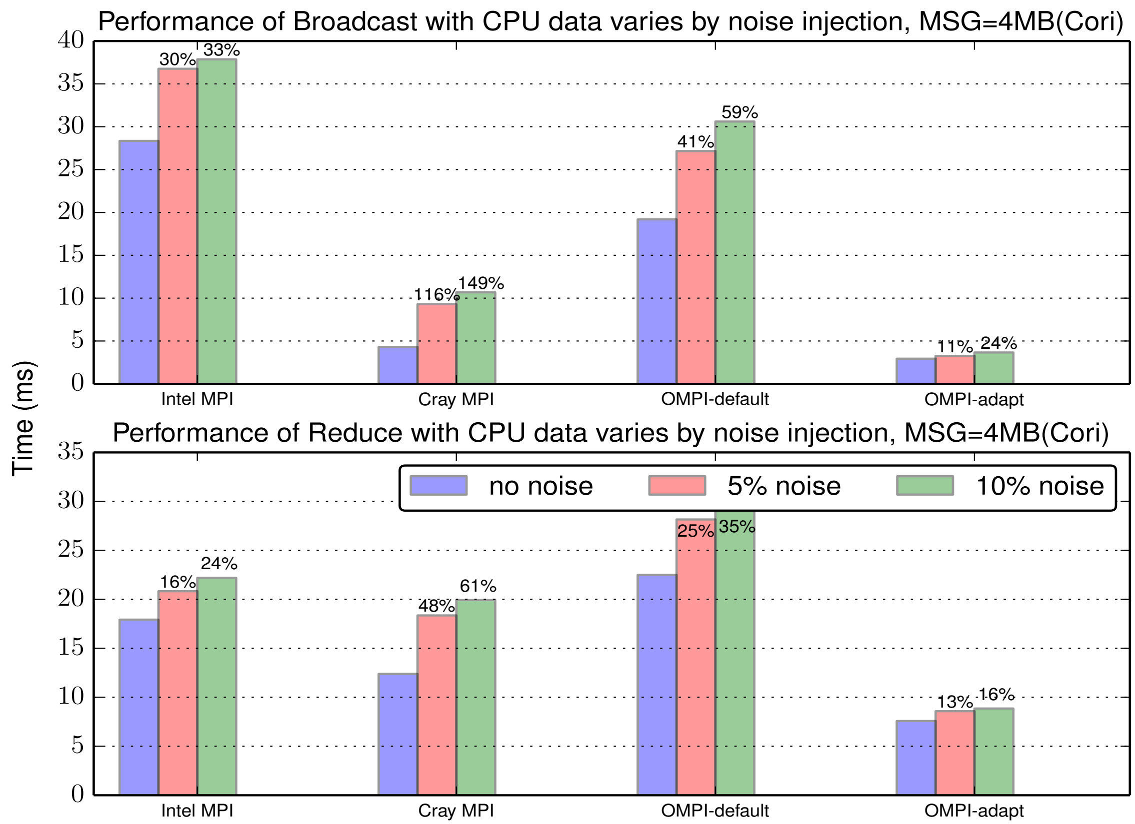
Collective Communication

Process location

Noise Reduction

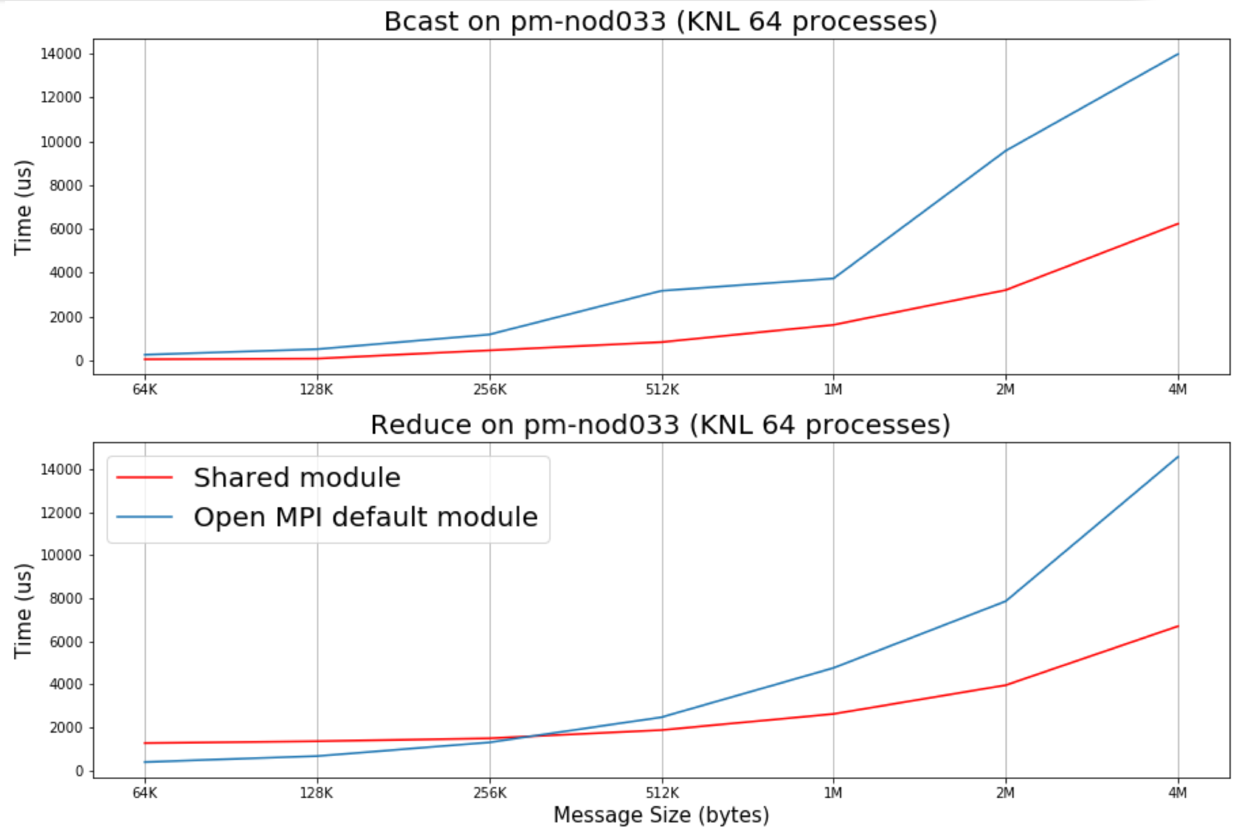
Shared Memory

Hybrid Architecture



Collective Communication

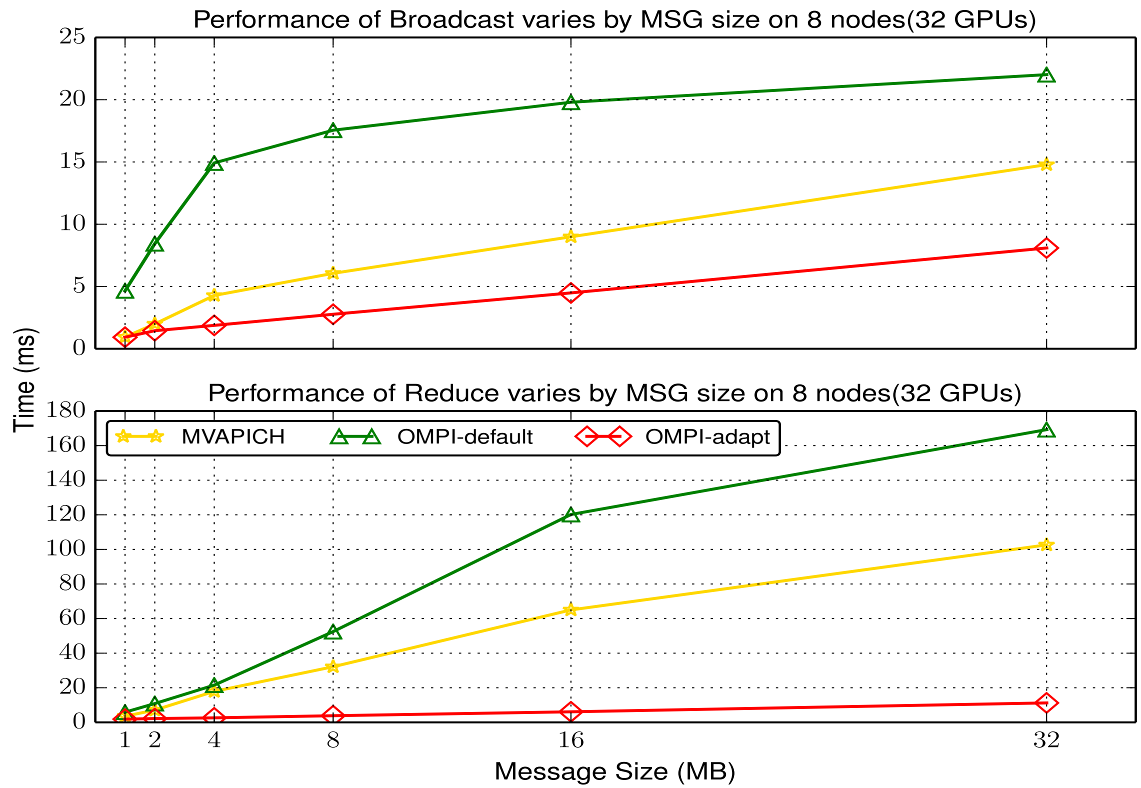
Process location
Noise Reduction
Shared Memory
Hybrid Architecture



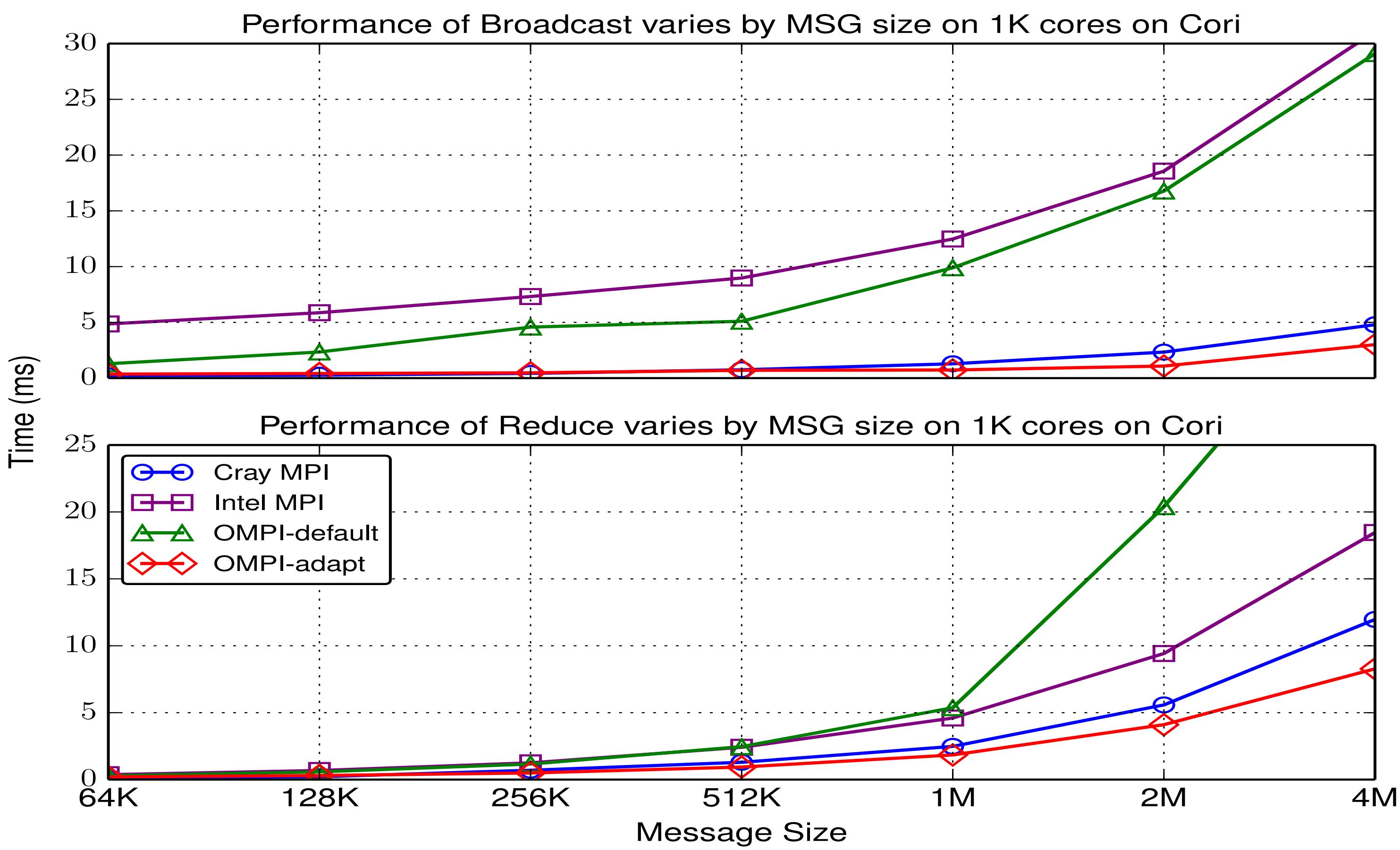
Collective Communication

Process location
Noise Reduction
Shared Memory
Hybrid Architecture

PSG Cluster:
4*K40/node
FDR IB

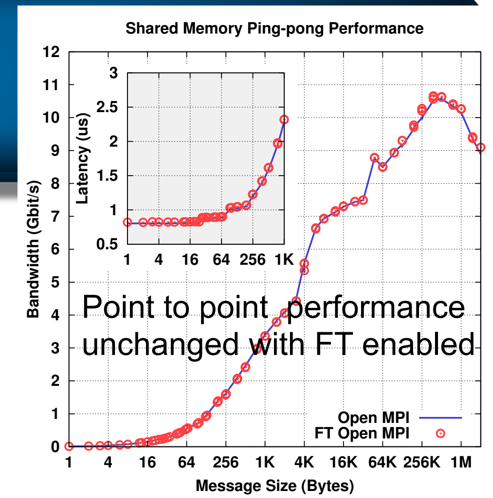


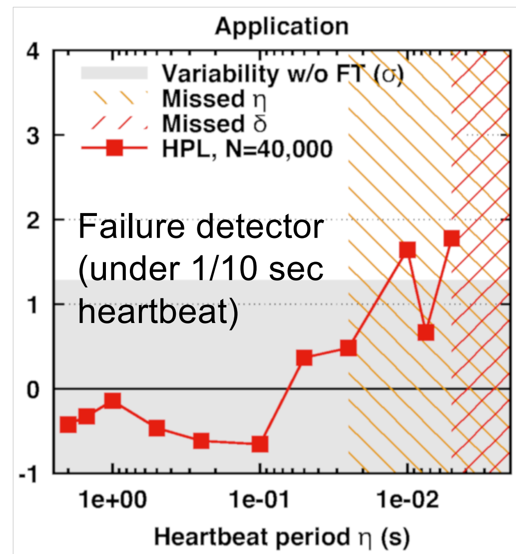
Collective Communication



Resilience - User Level Failure Mitigation (ULFM)

- Move the underlying resilient mechanisms outside ULFM/OMPI
  - Failure detector and reliable broadcast in PMIx
  - Used in OMPI ULFM and SUNY OpenSHMEM
- ULFM 2.1 released
  - Based on OMPI master (will remain in sync)
  - Transition to integrate ULFM in OMPI master
- Scalable fault tolerant algorithms demonstrated in practice for revoke, agreement, and failure detection (SC’14, EuroMPI’15, SC’15, SC’16)





OMPIO

Edgar Gabriel
University of Houston

OMPIO

- Highly modular architecture for parallel I/O
- Key features:
  - Tightly integrated with the Open MPI architecture (frameworks/modules, derived datatype handling, progress engine, etc.)
  - Support for multiple collective I/O algorithms
  - Automatic adjustments of number of aggregators
  - Multiple mechanisms available for shared file pointer operations

This work is funded in part by NSF grant SI2-SSI 1663887.

OMPIO

- New features:
  - Multi-threading support (Open MPI 3.1.0)
  - Better support for NFS file systems (Open MPI 3.1.1)

  - Support for CUDA GPU buffers (Open MPI 4.0.0)
  - New collective I/O component: vulcan (Open MPI 4.0.0)
  - Revamp of shared file pointer operations (Open MPI 4.0.0)
  - Support for more MPI I/O hints (Open MPI 4.0.0)

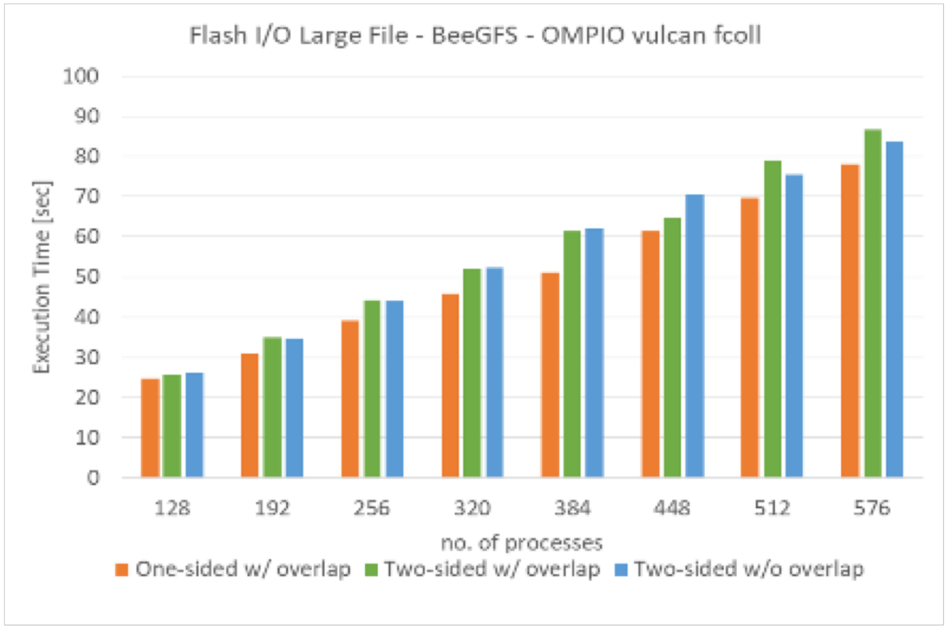
OMPIO file systems

- Generic Unix FS (XFS, EXT4)
- BeeGFS
- Lustre

- PVFS2/OrangeFS
- NFS

vulcan collective I/O component

- Features:
  - Overlaps two internal iterations of the algorithm
  - Uses asynchronous I/O (if available)
  - Communication based on two-sided (current release) and one-sided operations (upcoming release)



  - No data sieving

Mellanox Update

Joshua Ladd

UCX in Open MPI

- UCX PML replaces OpenIB BTL as the out-of-the-box network substrate for Infiniband fabrics in v4.x.
  - UCX GitHub – https://github.com/openucx/ucx
  - UCX now available in most Linux distros, will be in-box in the near future
- UCX transparently supports high-performance RDMA offloads:
  - Scalable reliable connections with DC transport (ConnectIB and higher)
  - MPI hardware tag matching offload (ConnectX-5 and higher)
  - Adaptive routing and out-of-order data placement (ConnectX-5 and higher)
  - GPU direct RDMA
- New in 2018:
  - Full GPU support on Nvidia (CUDA TL), and AMD (ROCM TL) GPUs.
  - Hardware-offloaded bitwise atomics, for OpenSHMEM v1.4.
  - Support for non-blocking memory registration.
  - Emulation layer for RMA/atomics over older hardware, shared memory, and TCP.
  - UCX OSC with multithreaded optimizations.
  - Multi-rail and HDR support.
  - Small message optimization with ConnectX-5 MEMIC.
  - Malloc hooks using binary instrumentation (BISTRO.)

HCOLL in Open MPI

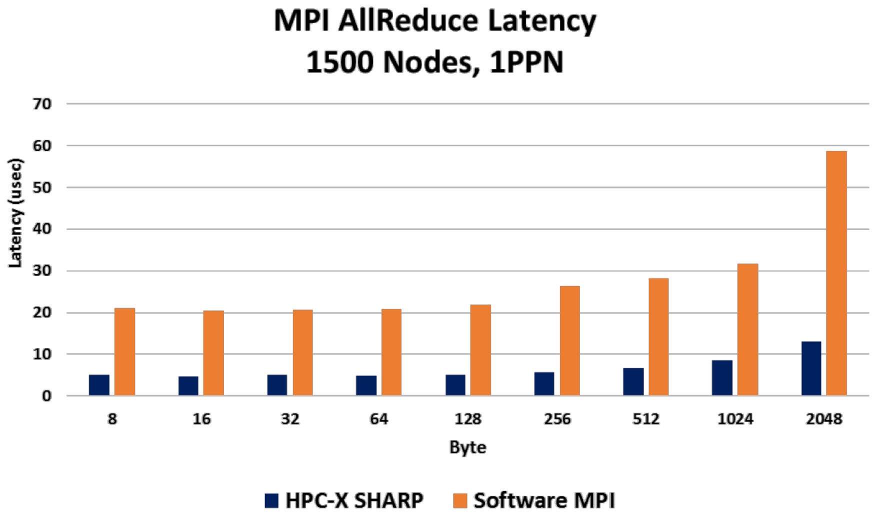
- Designed for exascale systems, now targeting Machine Learning frameworks.
- Deployed in production on Summit and Sierra
  - SHARP based allreduce, barrier
  - Multicast based broadcast
  - Highly optimized shared memory collectives
  - Optimized multithreaded

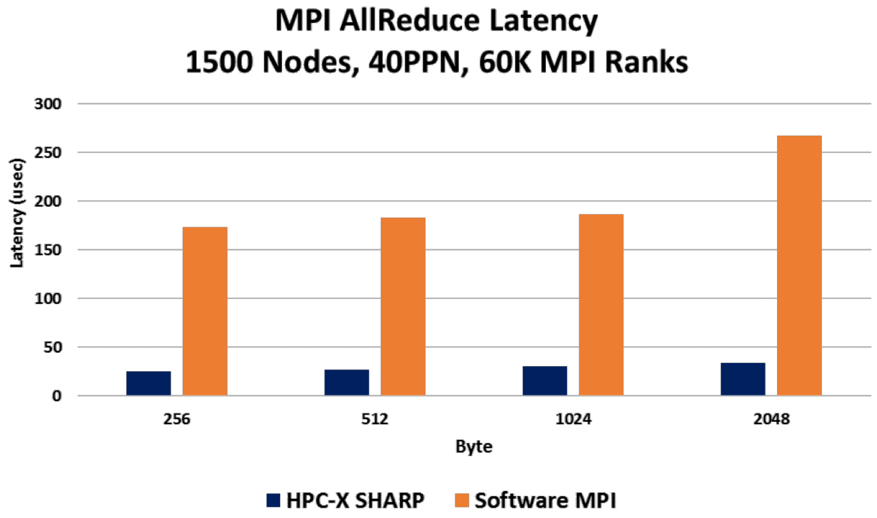
- Features targeting Machine Learning Workloads
  - Collectives over GPU Memory
    - SHARP small data reductions.
    - SHARP large data reduction(HDR ConnectX-6 / Quantum switch.)
    - Streaming reliable multicast for large data broadcast over GPU buffers.
    - UCX/GPU memory scatter-reduce-allgather algorithm for large data reductions.
    - Hierarchical GPU collectives.
  - Support for FP16 on Nvidia GPUs
    - Reductions on the GPU device
    - Reductions in the Switch.

OpenSHMEM v1.4 in Open MPI

- Available starting in Open MPI v4.0.0
- Contains many new features, allowing users to manage much more flexibility in communication and computation of OpenSHMEM programs
  - New feature list (specification 1.4, Annex G)
    - Communication management routines (context object)
    - Thread safety support
    - Sync routines
    - Test routines
    - Calloc routine for symmetric objects
    - Bitwise atomic operations

SHARP AllReduce Performance Advantages
1500 Nodes, 60K MPI Ranks, Dragonfly+ Topology

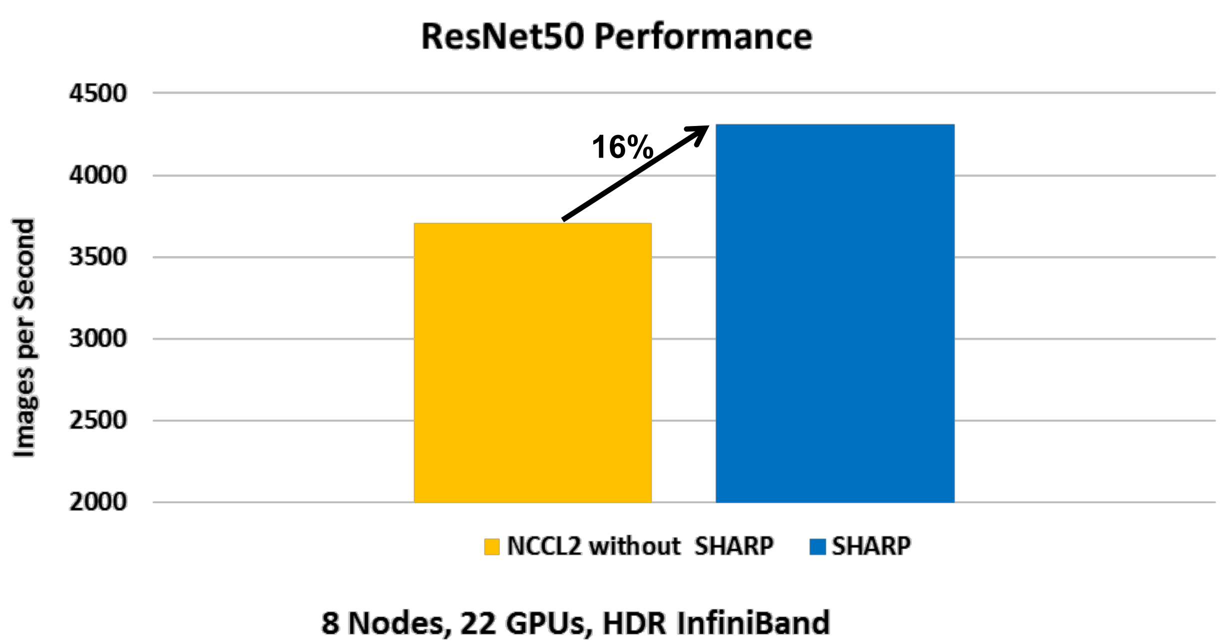




SHARP Enables Highest Performance

SHARP Performance Advantage for AI

- SHARP provides 16% Performance Increase for deep learning, initial results
- TensorFlow with Horovod running ResNet50 benchmark, HDR InfiniBand (ConnectX-6, Quantum)

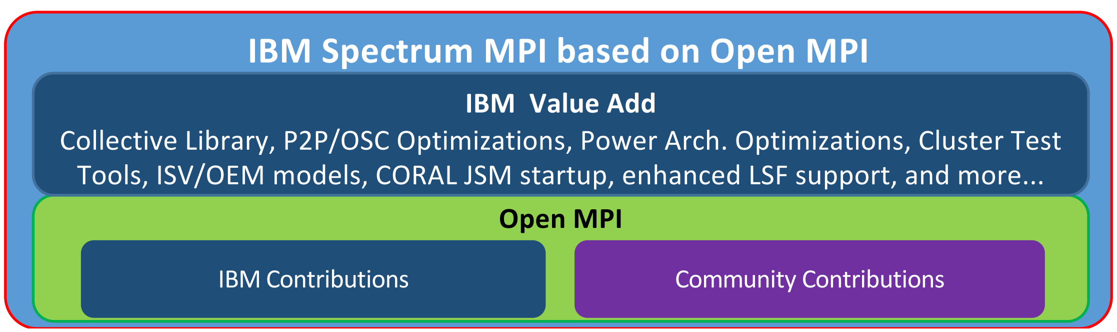


IBM Spectrum MPI

IBM Spectrum MPI

IBM Spectrum MPI

- IBM Spectrum MPI is a pre-built, pre-packaged version of community Open MPI plus IBM value add components.
- Spectrum MPI is based on Open MPI release branches
  - SMPI 10.1.0 based on OMPI v2.0.x branch
  - SMPI 10.2.0 based on OMPI v3.0.x branch
- Supporting scalable application performance on a variety of HPC systems including ORNL's Summit and LLNL's Sierra systems.
  - Improvements in MPI point-to-point, collective, and one-sided performance at all scales



Summary of Key Features

- Improved usability via command line options and packaging of tools
  - Interconnect selection (**`-tcp`**, **`-ibv`**, **`-pami`**), network selection (**`-netaddr rank:10.10.1.0/24`**)
  - Display table of interconnects used by your application
  - Supports multiple PMPI based tools both pre-packaged (e.g., Jumpshot by using **`-entry mpe`**) & user defined libraries (**`-entry mpe,mylib`**)
  - **`$MPI_ROOT`** mechanism to quickly switch between different SMPI versions
  - Single install for multiple compilers (GNU, XL, PGI)

```
$ mpirun -np 4 -prot -TCP ./hello
Host 0 [node01] ranks 0 - 1
Host 1 [node02] ranks 2 - 3

 host | 0    1
======|===========
    0 : shm  tcp
    1 : tcp  shm

Connection summary:
  on-host:  all connections are shm
  off-host: all connections are tcp

  0/  4) [node01] 61808 Hello, world!
  1/  4) [node01] 61809 Hello, world!
  2/  4) [node02] 10697 Hello, world!
  3/  4) [node02] 10698 Hello, world!
```

- Performance optimizations
  - Shared memory optimizations for POWER9 and PAMI cross memory attach
  - PAMI point-to-point and one-sided components support async. progress, hardware tag matching, on-demand paging, hardware data gather/scatter, dynamic tasking, POWER9 tunneled atomics, IB hardware atomics
  - CUDA IPC and GPU Direct support for Power Systems
  - libcollectives library of IBM tuned collective operations with the ability to automatically chooses ‘best’ algorithm at runtime based on a variety of criteria.

IBM **Spectrum MPI**

ARM Update

Arm Update

- Open MPI works on Arm!
  - https://developer.arm.com/products/software-development-tools/hpc/resources/porting-and-tuning/building-openmpi-with-arm-compiler
  - https://developer.arm.com/products/software-development-tools/hpc/resources/porting-and-tuning/building-openmpi-with-openucx

Arm Update

- Active collaboration between LANL and Arm to enable CI and MTT testing on Arm
  - Arm CI machines with InfiniBand hosted at HPCAC
  - Arm CI/MTT machines hosted at LANL

AWS & Open MPI

Brian Barrett & Raghu Raja

TCP Transport

- Improving network configuration support
  - Multiple IPs per network device (in master)
  - Differing number of interfaces
  - Complex routing configurations
- Multiple TCP connections between ranks
  - The `btl_tcp_links` MCA parameter had been around for many releases, but had bit-rotted
  - Works for simple cases, expanding in future

CI Testing

- "Pull Request Build Checker" running in AWS
- 23 test builds for every PR, all but 3 run in AWS
- Working on accelerating test execution time

PMIx in OMPI

Ralph H. Castain
Joshua Hursey

Current State

- PMIx plays integral role today
  - Embedded for “simple install”
    - OMPI 4.x → PMIx 3.x
    - OMPI 3.x → PMIx 2.x
  - Symbol-shifted to avoid conflicts with application-level bindings
- ORTE
  - Primary RTE for unmanaged environments

Future Directions

- PMIx as “First Class Citizen”
  - Direct use of PMIx functions
  - No more symbol shifting
- ORTE → PRRTE (pronounced: purr-tay)
  - Reduce RTE “cost” by sharing it with PMIx
  - PMIx-based tools

PMIx BoF: Wed, 5:45-6:15pm
Focus: Application-level examples!
(https://pmix.org)

Fujitsu Limited

FUJITSU

MPI for the Post-K Computer

- Post-K MPI based on Open MPI
  - Work on A64FX (Armv8.2-A + SVE) and TofuD
  - Plan to use Open MPI v4.0 and PMIx v2.1
- Contribution to Open MPI from post-K MPI
  - Persistent collectives [*see next page*]
  - Datatype for half-precision floating point [*early 2019*]
  - Thread parallelization of pack/unpack [*early 2019*]

The post-K computer is underdevelopment by RIKEN and Fujitsu

Persistent Collectives in MPI-4.0

(or MPI-3.2)

- Persistent collectives are in Open MPI 4.0.x
- Overlap computation & communication and reduce communication initialization cost
- Use *MPIX_* prefix because standardization is not complete
- Performance is similar to nonblocking collectives

```
MPIX_Bcast_init(
  buf, count, ..., &req);
for (...) {
  MPI_Start(&req);
  // ... your computation
  MPI_Wait(&req, &stat);
}
MPI_Request_free(&req);
```

See *man MPIX_Barrier_init* for details

Bull Open MPI

Guillaume Papauré

Atos | montblanc-project.eu | @MontBlanc_EU

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement n° 671697

Performance oriented MPI

- Application performance
  - Hierarchical collectives optimizations (work done with University Tennessee Knoxville [UTK])
  - Tuned for Bull eXascale Interconnect (BXI)
    - portals4 offload: tag matching, rendezvous, non blocking collectives
    - Tera-1000: 8256 nodes, 11.9 Pflops, 14th TOP500 (June 2018)
- MPI+X
  - Bull Hybrid Communication Optimizer

    (currently Open MPI+OpenMP; other runtimes planned)
  - One sided notifications support in Open MPI OSC

Bull
atos technologies

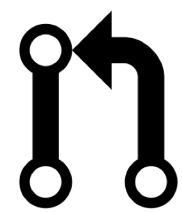
User oriented MPI

- Ease of use
  - Hybrid MPI+OpenMP mpirun options
  - User parameters profiles
  - Collectives numerical reproducibility (work done with UTK)
- Fits with increasing HPC heterogeneity
  - ARM, x86_64, GPU+affinity
  - gcc, Intel compiler, ARM compiler
  - Supports all the way up through MPI_THREAD_MULTIPLE

Wrap up

Where do we need help?

- Code
  - Any bug that bothers you
  - Any feature that you can add
- ***User documentation***
- Testing (CI, nightly)
- Usability
- Release engineering

We ❤️ 

Come join us!