

**SC23**  
Denver, CO | i am hpc.

# Open MPI State of the Union XV Community Meeting

George Bosilca

Edgar Gabriel

Pritchard, Howard

Tomislav Janjusic





# 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 v $Y$  is backwards compatible with Open MPI v $X$  (where  $Y > X$ ) if:
  - Users can compile a correct MPI / OSHMEM program with v $X$
  - Run it with the same CLI options and MCA parameters using v $X$  or v $Y$
  - 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



# Version Roadmaps

# v4.0.x (Previous stable)

- Release managers
  - Howard Pritchard, Los Alamos National Lab
  - Geoff Paulsen, IBM
- Current release: v4.0.7
- Released Nov 15, 2021



- No further releases are planned

# v4.1.x (Current stable)

- Release managers
  - Brian Barrett, AWS
  - Jeff Squyres, Cisco
- Current release: 4.1.6
  - September 2023







# Open MPI v5.0.0

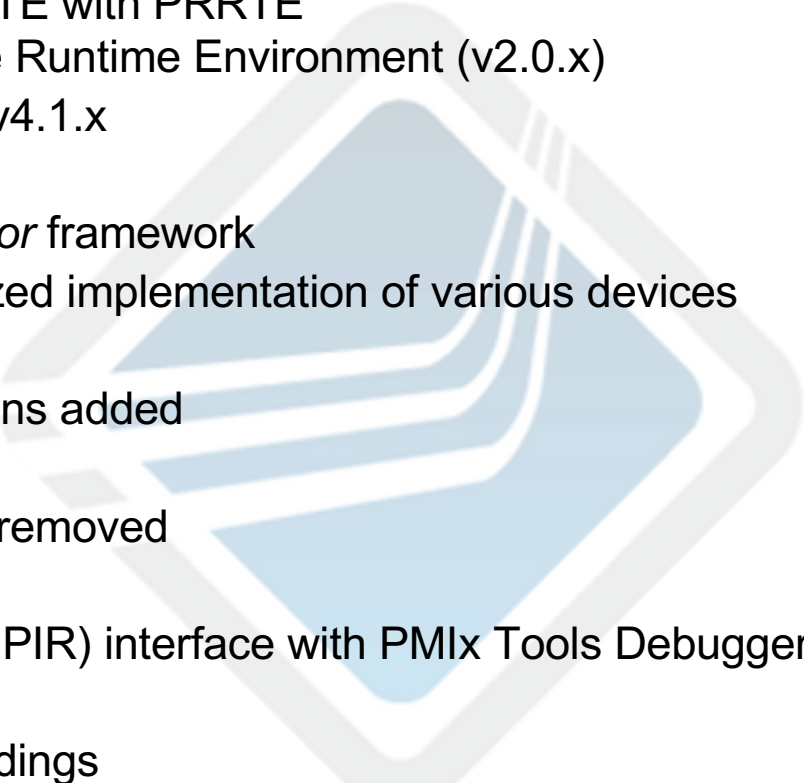
Many improvements are coming!

# v5.0.x

- Release managers
  - Austen Lauria, IBM
  - Tomislav Janjusic, NVIDIA
  - Geoff Paulsen, IBM
- Current release
  - V5.0.0



# Open MPI v5.0.0 Major Changes

- Replaced launcher ORTE with PRRTE
    - PMIx Reference Runtime Environment (v2.0.x)
    - Requires PMIx v4.1.x
  - Added a new *accelerator* framework
    - Allows for modularized implementation of various devices
  - Support for MPI Sessions added
  - OpenIB BTL has been removed
  - Replaced Debugger (MPIR) interface with PMIx Tools Debugger interface
  - Removed MPI C++ bindings
- 

# New and Improved Runtime Environment

- PRRTE is our new launcher!
  - PMIx Reference Runtime Environment (v2.0.x)
  - PMIx standard based
- ORTE effectively forked into [its own repo](#)
- Run time no longer specifically tied to Open MPI
  - Also works with other MPI implementation
  - And in other non-MPI/HPC environments

# What is PRRTE?

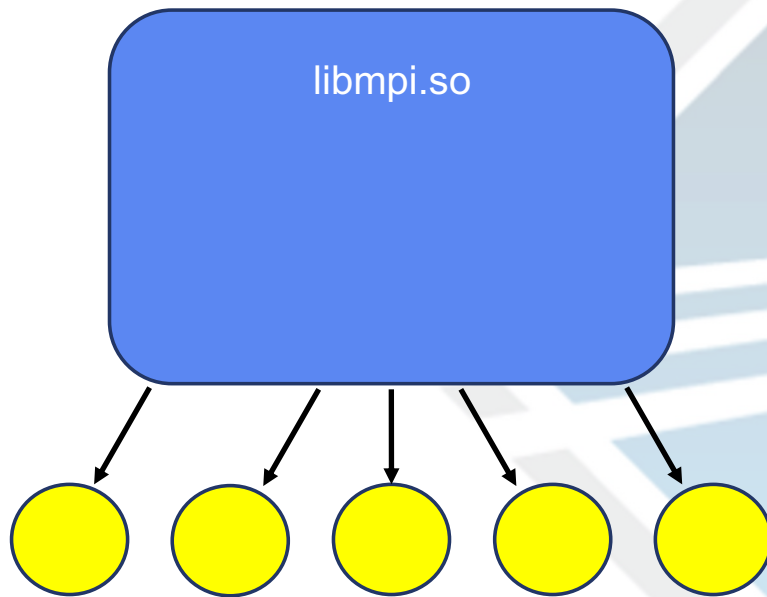
- PMIx Reference Runtime Environment
  - Feature full, scalable, PMIx-Enabled runtime environment
- It is an open source, community supported, runtime implementation
  - Supports one-off jobs via prterun and multiple jobs via persistent daemons
  - Requires OpenPMIx >= 4.1.0
- Evolved from ORTE in Open MPI to standalone project
  - Provides much of the same feature set as ORTE, plus more
  - Some mpirun command line options have changed
  - The changes were documented and addressed via aliases
  - mpirun and mpiexec are now small programs that exec prterun

# What is OpenPMIx

- Is a feature complete implementation of PMIx standard
  - **P**rocess **M**anagement **I**nterface - **E**xcascale
- Provides an implementation to connect PMIx-enabled clients (e.g., Open MPI) with tools (e.g., debuggers), and resource managers (e.g, PRRTE, SLURM, IBM JSM)
  - PMIx also provides event notification (fault tolerant libraries), and process wire-up, including “instant on” where supported
- OpenPMIx is an open source, community supported, scalable implementation
  - OpenPMIx releases tied to corresponding PMIx Standard releases
  - Proving ground for new PMIx Standard additions
  - Used on many large scale HPC systems

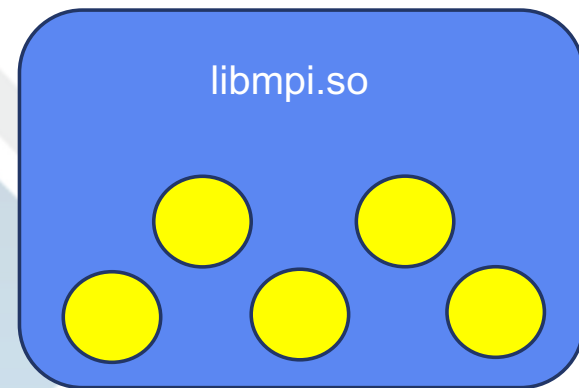
# New v5.0.x defaults

$\leq$  Open MPI v4.1.x



Plugins loaded dynamically at runtime

$\geq$  Open MPI 5.0.0



Plugins located in the library

# Packaging

Package	V4.x	V5.0.x
Hwloc	Prefer external	Prefer external
Libevent	Prefer external	Prefer external
Open PMIx	Prefer external	Prefer external
ROMIO	Internal	Internal
Treematch	Internal	Internal
PRRTE	N/A	Prefer external

- v4.x: OMPI prefers an external library and only allow the use of external libraries newer than internal versions
- v5.0.x: OMPI prefers external packages that meet our version requirements and allow them even if they are older than our internal version



# ABI / Command Line Changes

- Now following GNU CLI conventions
  - Multi-letter tokens are double-dash (e.g., --mca not -mca)
  - Will automatically replace and print a warning if a single-dash is used
- Some MCA params have moved from *ompi\_* to *prte\_* or *pmix\_*
  - Automatically convert MCA params to their proper project
    - Can be specified in default param files, environment, and command line.
  - To see the full list of MCA parameters for each project, run `ompi_info --all`, `prte_info --all`, or `pmix_info --all`
  - For further information please refer to the docs at <https://docs.open-mpi.org/>

# Open MPI Documentation

- New central docs location: <https://docs.open-mpi.org/>
- Combines all previous documentation:
  - README
  - NEWS
  - FAQ
  - Man pages
- These HTML pages are also included in the distribution tarball
  - Suitable for offline / disconnected-from-the-internet viewing
  - Man page files also included in the tarball
- Contains high-level descriptions, detailed examples, man pages (MPI APIs and command line executables), etc.

# New Features

- User Level Fault Mitigation ([ULFM](#))
  - Conforms to the ULFM MPI Standard draft proposal
  - Enabled by default
  - New ULFM test suite in [github.com/open-mpi/ompi-tests-public](https://github.com/open-mpi/ompi-tests-public) repository
- Threading MCA [framework](#)
  - pthreads (default), Argobots, and Qthreads.
  - Specify by configure with “--with-threads=”
- Enable load-linked, store-conditional atomics for [AArch64](#)
  - Up to 40x performance improvement with multi-threaded lifo/fifo test benchmarks

# New Features Cont.

- Added OMPIO GPFS filesystem support.
- Added OMPIO atomicity support.
- `--mca ompi_display_comm 1`
  - Displays a proc-by-proc communication matrix to stdout
- FP16 support via MPIX extensions
- `memory_patcher`
  - Add ability to detect patched memory.
- OpenSHMEM v1.5 compatible
  - Headers present with v1.5 features in development

```
Host 0 [helios017] ranks 0, 4, 8, 12
Host 1 [helios018] ranks 1, 5, 9, 13
Host 2 [helios019] ranks 2, 6, 10, 14
Host 3 [helios020] ranks 3, 7, 11, 15
```

```
host | 0 1 2 3
-----|-----
3/16
0 : sm  uct uct uct
1 : uct sm  uct uct
2 : uct uct sm  uct
3 : uct uct uct sm
```

```
Connection summary: (pml)
on-host: all connections are sm
off-host: all connections are uct
```

# New Features Cont.

- New UCC (Unified Collective Component) from UCX community
  - Configure with `--with-ucc=<DIR>` and increase priority
  - Added `MPI_Scatter()` and `MPI_Isscatter()` collectives
- Added a new Accelerator framework.
  - CUDA-specific code was replaced with generic framework standardizing various device features
  - Allows for modularized device implementation
  - Open MPI build can be shipped with CUDA support enabled without requiring CUDA libraries
- Added CUDA support to [mtl/ofi](#)
  - Requires Libfabric  $\geq$  v1.9
- Hierarchical collective framework (HAN) is the default

# Removed Features

- osc/pt2pt
  - Not maintained, very buggy
  - Replaced with osc/rdma + btl/tcp
- btl/openib
  - IB/RoCE now supported via UCX
- Legacy btl/sm
  - Replaced with btl/vader, which is renamed to btl/sm (shared-memory).
  - Alias “vader” exists for backwards compatibility
- MXM support
  - pml/yalla, opal/atomic/mxm, oshmem/spml/ikrit all removed

# Other Removed Features

- 32bit builds are more limited
  - Now only supported with C11 compliant compilers
- Support for GNU gcc compilers < v4.8.1
- Deprecate PMI support
  - PMI-1 and PMI-2 are no longer supported
  - PMI shim is available [here](#)
- MPI C++ bindings
  - Removed from MPI 3.0 standard (9 years ago)

# More Removed Features

- [fcoll/two\\_phase](#) removed
- The “--am” and “--amca” options are [deprecated](#)
- [patcher/linux](#)
  - Attempted to hook calls by patching dynamic symbol table,
  - however it did not work in all cases
- Checkpoint-Restart (CR) is now completely [removed](#)



# MPIR has been removed

- What is MPIR?
  - MPI Process Acquisition interface
  - Is not an API, but instead "... requires that a tool reads symbol table information and traces the starter process..."
  - Used by debuggers for MPI processes
- Replaced by the **PMIx tool interface**
  - gdb scripts that rely on MPIR\_\* routines will need to be updated
  - Contact your debugger provider for their timeline
  - To use non-PMIx enabled debuggers, see [MPIR-to-PMIx guide](#) for shim layer
- Initially announced at SC'17 BOF
  - Deprecation notice in NEWS in early 2018
  - User runtime warning in v4.0.0 (mid/late 2019)
  - [Finally] Removed in v5.0.0

# MPI-4 support in v5.0.0

- Open MPI v5.0.0 is not MPI-4.0 compliant
  - Officially, it is still MPI **v3.1** compliant
  - However, it does have a number of MPI v4.0 features!
- Support for MPI Sessions
- ERRORS\_ABORT infrastructure
  - New error classes to distinguish between aborting a communicator and aborting an entire application
- Initial error handler implementation
  - The error handler that is set before MPI is initialized and after it is finalized
  - Can be selected from mpiexec/MPI\_Comm\_spawn() info key parameters

# MPI-4 support in v5.0.0 cont.

- Error handling for 'unbound' errors to MPI\_COMM\_SELF
  - Requires that unbound errors trigger the error handler on MPI\_COMM\_SELF instead MPI\_COMM\_WORLD
- Persistent collectives added to the MPI\_ namespace
  - Previously available via the MPIX\_ prefix
- MPI\_Comm\_get\_info(), MPI\_File\_get\_info(), and MPI\_Win\_get\_info() updated to be MPI 4.0 compliant
  - Changes to modified keys are now reflected

# MPI-4 support in v5.0.0 cont.

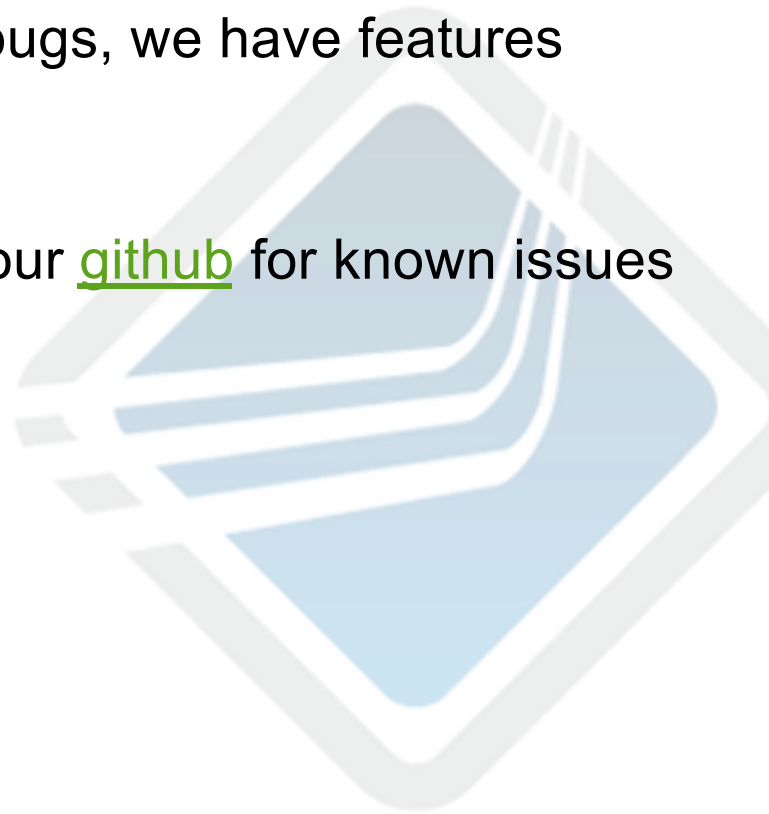
- Partitioned communication implemented
- Support for *mpi\_minimum\_alignment* info key
- MPI\_Info\_get\_string() support
  - Replaces MPI\_Info\_get() and MPI\_Info\_get\_valuelen()
- What is still missing ?
  - Big Count (WIP), MPI Events

# Open MPI v5.0.x Schedule

- OMPI v5.0.x branched from master
  - March 11, 2021
- v5.0.0 Officially released Oct. 27<sup>th</sup> 2023
- Target date for v5.0.1 Dec 15<sup>th</sup> 2023
  - Bug fix release
- Goal is to stabilize v5.0.x
  - Users should start testing and making the switch v5.0.x

# Known issues

- We don't have bugs, we have features
- Please refer to our [github](#) for known issues





# OMPI v5.0.0 NVIDIA Updates

Mamzi Bayatpour, Tomislav Janjusic,  
Joshua Ladd



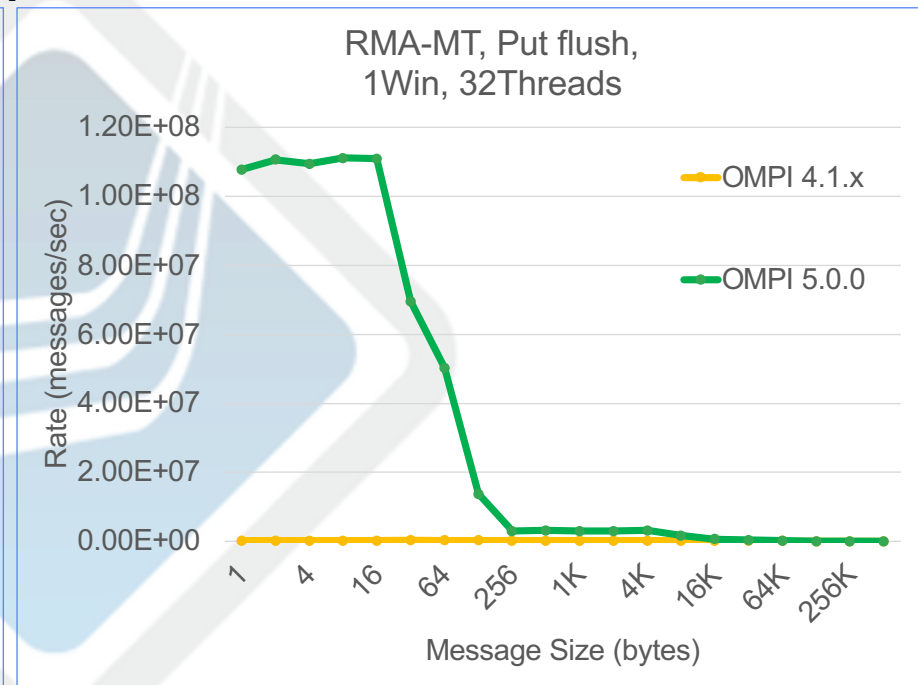
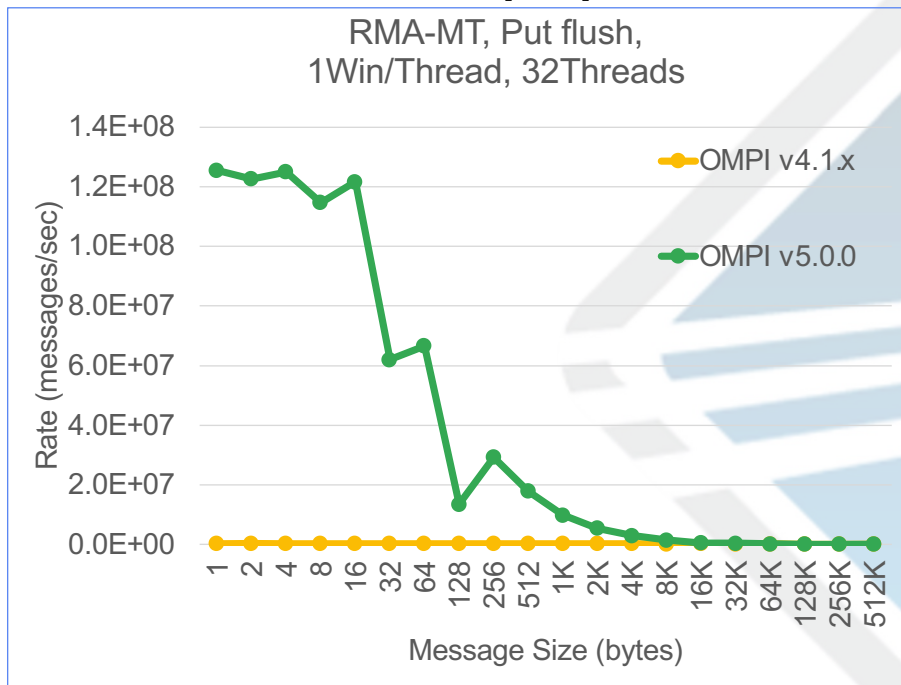
# UCX One-sided Component

- **Effort to deliver state-of-the-art performance and be the best-in-class MPI-RMA implementation**
- **Worker-Pool implementation (targeting multi-threaded performance)**
  - Complete re-design from OSC/UCX generation in v4.1.x.
  - Delivers significantly better performance for multi-threaded applications.
- **Bug fixes and stability improvements**
  - Nearly 100% passing rate using MPICH and RMA-CI test suites.
  - Capable of running real-world applications (NWCHEM) at large scales with improved performance.
- **Resource utilization optimization**
  - Complete re-design delivering savings on network resources.



# OSC/UCX: worker pool

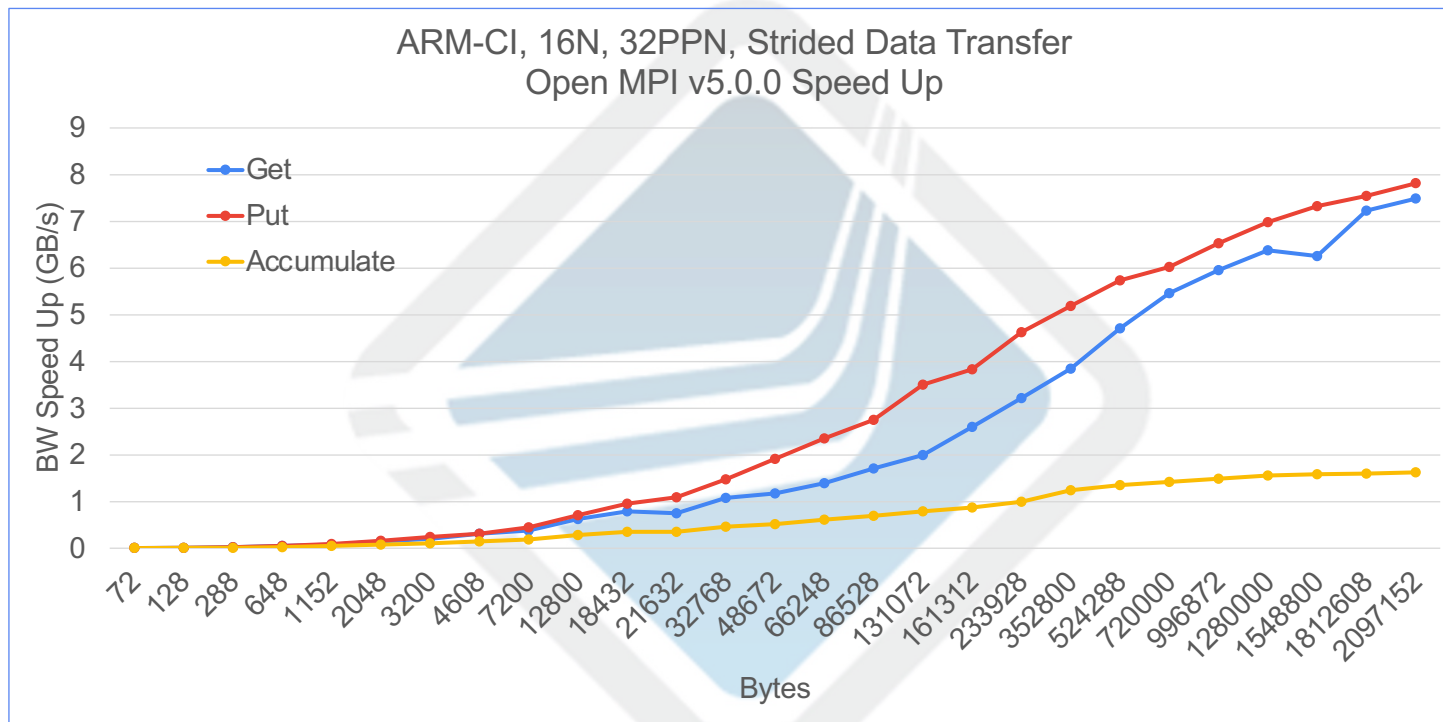
- **OSC thread-multiple performance improvements out-of-the-box in v5.0.0**



# Scaling Open MPI to Exascale

- Non-contiguous datatype support
- Resource utilization improvements
- Asynchronous Progress improvements
- Scaling NWCHEM to 16,384 MPI ranks

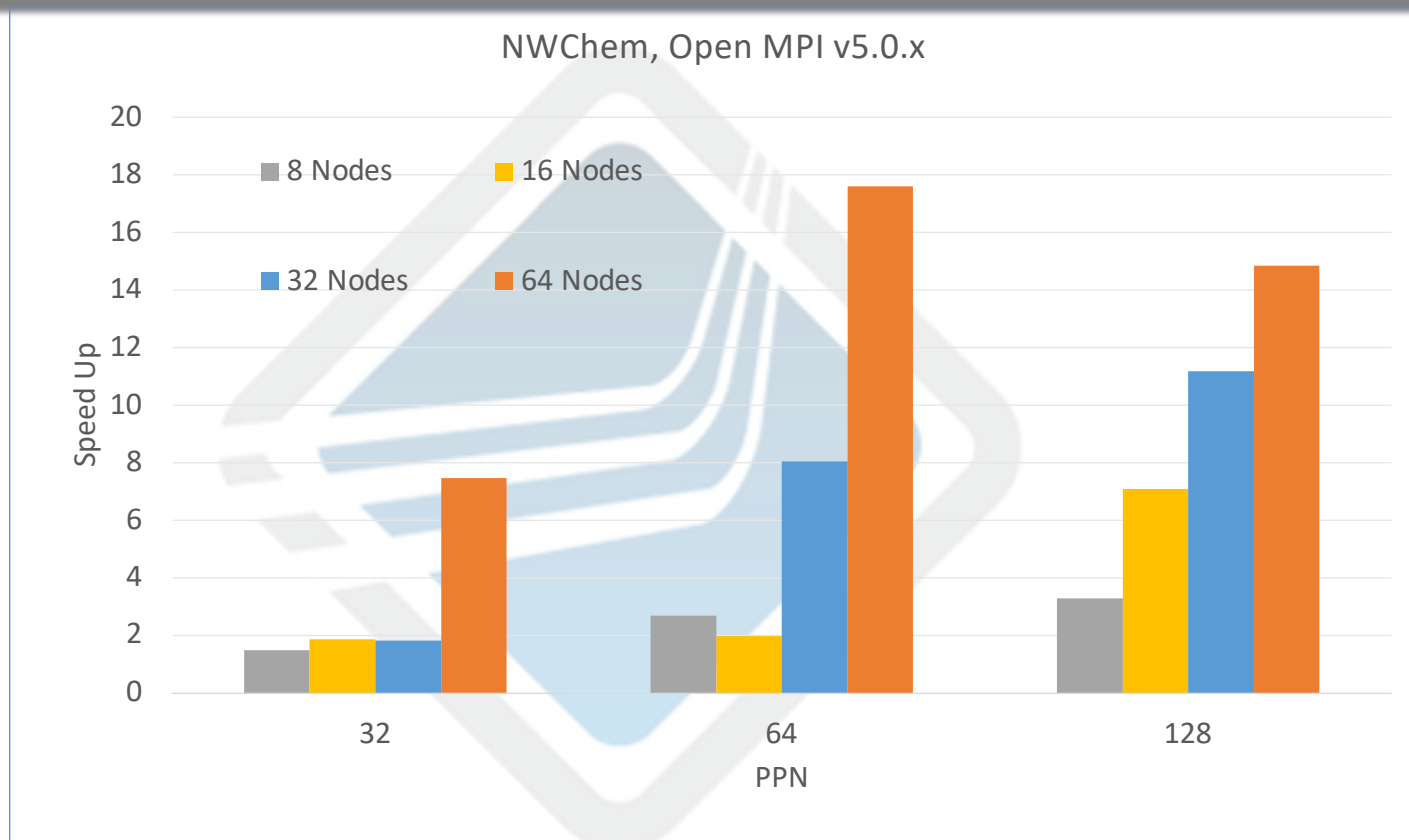
# ARM-CI Strided Data Transfer



# NWCHEM Performance

## Microsoft Azure HPC

- HBv4 VMs
- Nvidia IB NDR 400
- AMD Genoa
- HPC-X v2.15



# Acknowledgements

Our sincere thanks to Microsoft Azure HPC, Jithin Jose and Jie Zhang, for providing us with early access to HBv4, Azure's HPC's state-of-the-art HPC VMs.



# Support for AMD GPUs in Open MPI

**Edgar Gabriel**

**[edgar.gabriel@amd.com](mailto:edgar.gabriel@amd.com)**

**AMD**   
together we advance\_

# Open MPI 4.1.x with AMD ROCm Devices

- MPI jobs can utilize ROCm device memory through components that support AMD GPUs with Open MPI 4.1.x, for example:
  - **UCX** : Point-to-point communication with device memory (pml/ucx) and one-sided operations (osc/ucx)
  - **UCC**: Collective operations with device memory (starting from Open MPI 4.1.4) (coll/ucc)

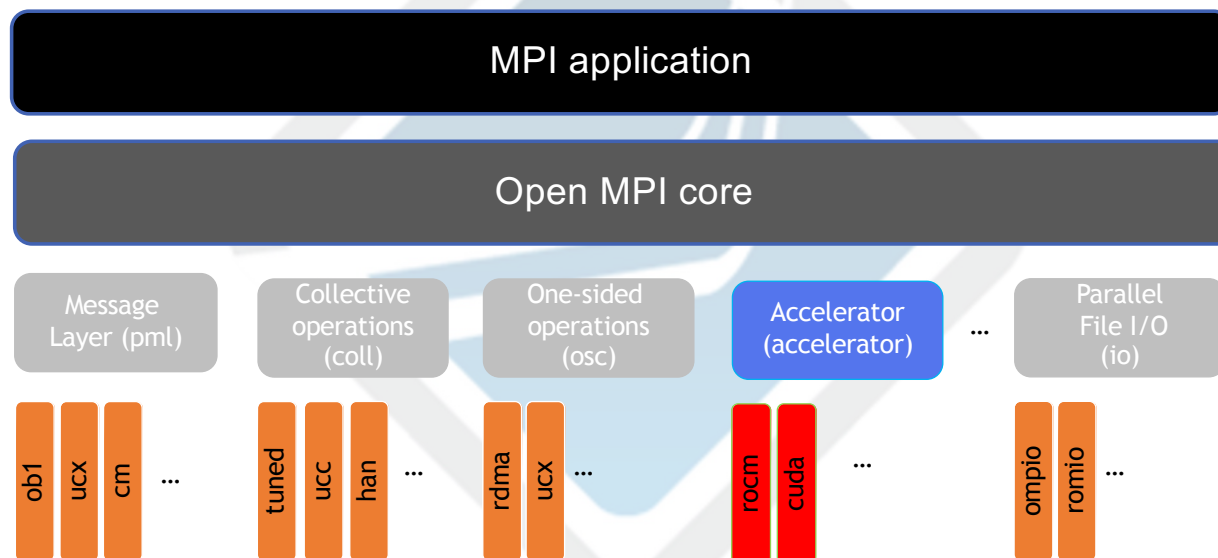
# ROCM Support in Open MPI 5.0

- ROCm support added to Open MPI starting v5.0
  - ROCm memory type detection
  - ROCm device memory allocation
  - Data transfer to/from ROCm device memory
    - Non-contiguous derived datatypes
    - MPI File I/O
  - Enables usage of additional data transfer components (e.g., libfabric, ob1, etc.)
  - Query availability of ROCm support in MPI library (`MPHX_QUERY_ROCM_SUPPORT`)



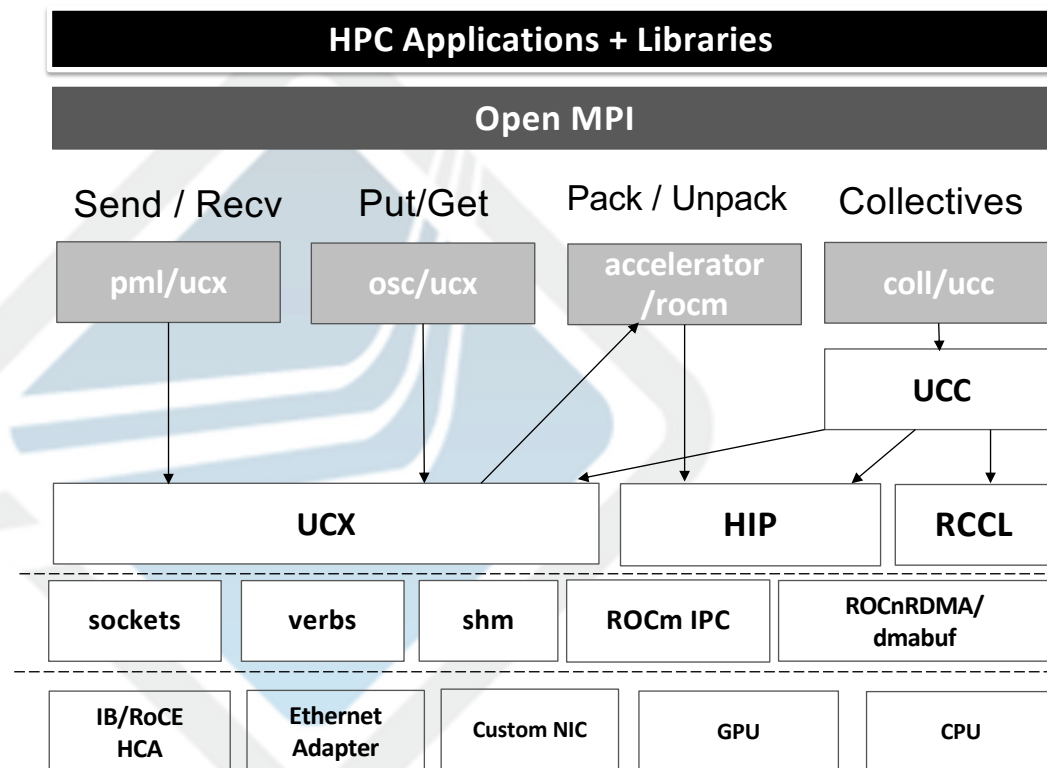
# Accelerator Framework in Open MPI 5.0

- New framework introducing an abstraction layer for GPU support in Open MPI

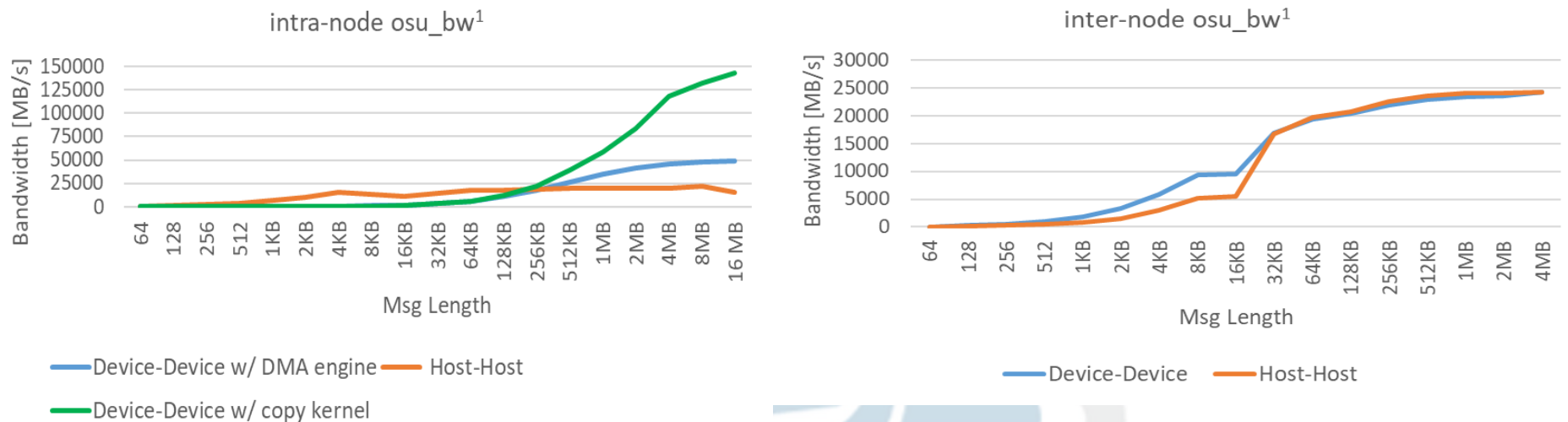


# ROCM Aware Open MPI Software Stack with UCX and UCC

- Recommended software stack for InfiniBand and RoCE networks
- Most stable and best tested configuration



# Performance Results: Point-to-Point



## Key take-aways:

- Intra-node device-to-device data transfers over high-bandwidth InfinityFabric links
- Direct access to ROCm device buffers for inter-node data transfers using ROCnRDMA kernel component for RDMA capable network adapters

## Hardware:

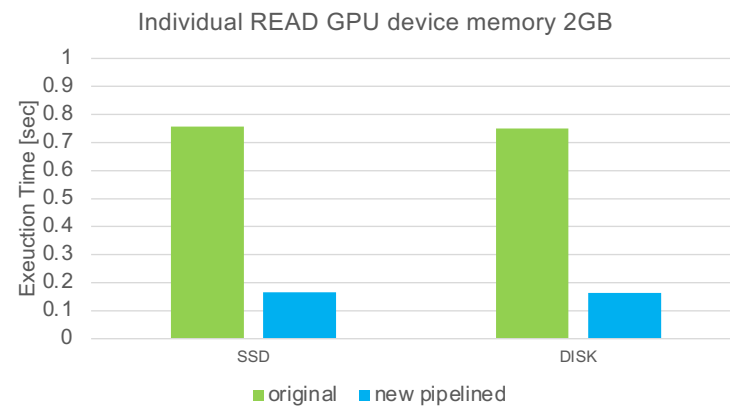
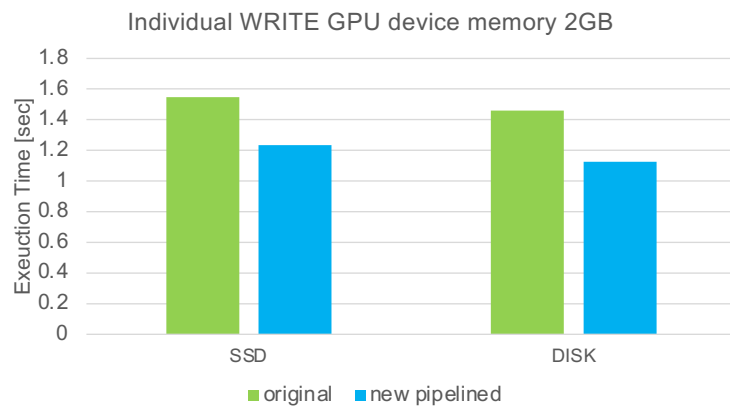
- 8 AMD Instinct MI250 GPUs per node
- 200Gb Broadcom RoCE network interconnect

## Software:

- Open MPI 5.0.0
- UCX 1.15.0
- ROCm 5.7.0

[1] OSU Benchmark Suite <https://mvapich.cse.ohio-state.edu/benchmarks/> (BSD License).

## Performance Results: MPI\_File\_read/write



- Enhanced pipeline protocol to perform file I/O operations directly to/from GPU buffers in the Open MPI 5.0.0

# Other Transports

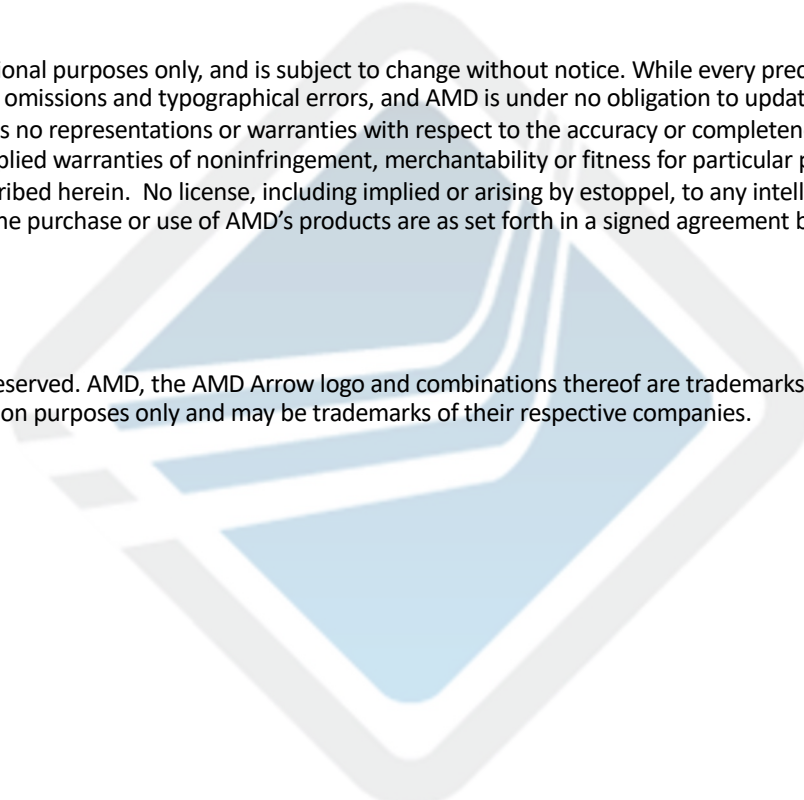
- Open MPI using libfabric (pml/cm with mtl/ofi)
  - Significantly enhanced support for AMD ROCm devices in libfabric 1.19
    - Support for device-to-device IPC transfers
    - Support for asynchronous copy operations
  - Code contributed by Oak Ridge National Laboratory
- Open MPI using pml/ob1
  - Provides support for ROCm device memory through a (host) staging buffer
- More work planned for upcoming Open MPI releases to further enhance ROCm support for libfabric and ob1



## DISCLAIMER

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18

©2023 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.



**AMD** 



# Open MPI Activities at Los Alamos



LA-UR 23-32780




# Los Alamos - Current Work

- Support for Intel Ponte Vecchio Accelerators (ANL focused)
- Open PMIx shared memory based GDS component – available in upcoming v5.0.0 release
- Initial work to support MPI ABI proposal
- Spack package maintenance

# Support for Ponte Vecchio (ZE)

- A ZE component was added to accelerator framework this past quarter as part of ECP work
- Only available in main branch
- Functional (hopefully). Only lightly tested.
- Testing/development hindered by old software on sunspot – SLES15sp3 (so no GPU direct), old CXI version 2.0.2

# MPI ABI effort

- Preliminary investigation
  - Work will include support for MPI\_Count
  - Targeting a post 5.0 release stream
- 

# Open MPI and Spack

- Things should work if you stick to the 4.1.x releases
  - No ROCM support however
- 5.0.0 is presenting challenges
  - Hooks to support ROCM in flight
  - Many changes between the 4.1 release stream and 5.0, so you may need to change your spack variants



# Open MPI on Frontier

Amir Shehata, David Bernholdt, Thomas Naughton (ORNL)  
Howard Pritchard (LANL)

# Open MPI on Slingshot 11

- ORNL & LANL porting/tuning for HPE Cray EX
  - OMPI-X: US DOE Exascale Computing Project
  - Support new Open MPI on new exascale systems
- CXI libfabric provider for Slingshot 11
  - Supports host & device (GPU) buffers
  - CXI not directly support on-node comm.
- Explored potential paths to use CXI
  1. **MTL path** – use libfabric tagged message interface
  2. **BTL path** – use libfabric for byte transfer only, MPI for tag matching & higher level logic



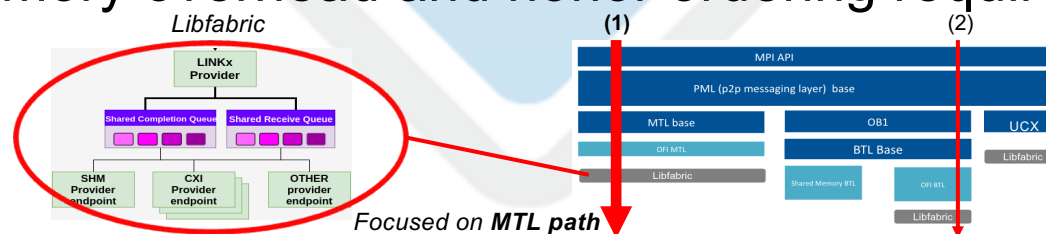
ECP OMPI-X Project



Frontier @ OLCF

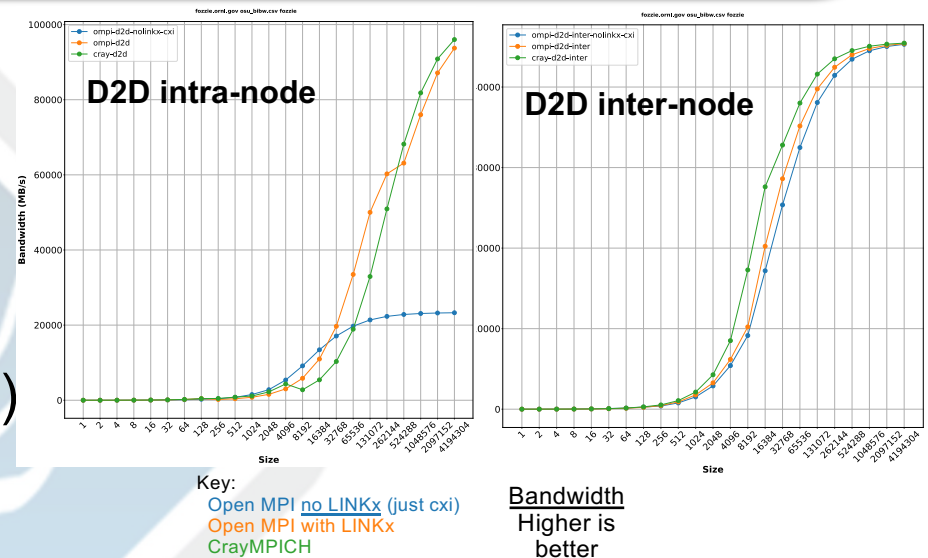
# New Libfabric “*LINKx*” provider

- **Challenge:** MTL framework limited to 1 active component
  - Cannot use both libfabric shared memory & CXI providers
- Create OFI libfabric provider to link other “core” providers
  - Enables Open MPI to use 1 provider for local and remote peers
  - *LINKx* chooses endpoint provider based on peer locality
    - Use CXI (inter-node) and SHM (intra-node)
  - *LINKx* shares completion & receive queues to improve performance, reduce memory overhead and honor ordering requirements



# Current Status

- OSU bi-directional bandwidth
  - Intra-node shows use of LINKx to avoid egress/ingress
- Status
  - Support for job launch (VNI, PALS)
  - Deployed on Frontier
  - Currently tuning collectives



## Getting Started on Frontier

```

module unload cray-mpich cray-pmi
module use /sw/frontier/ums/ums024/cce/15.0.0/modules
module load openmpi
mpirun --bind-to core --map-by ppr:1:l3cache --np $SLURM_NTASKS gpuwrapper.sh ./app
    
```


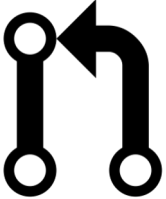




Questions?

# Where Do We Need Help?

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

We  



Come join us!