

ENABLER OF CO-DESIGN



UCX Community Meeting

SC'19

November 2019

This is an open, public standards setting discussion and development meeting of UCF. The discussions that take place during this meeting are intended to be open to the general public and all work product derived from this meeting shall be made widely and freely available to the public. All information including exchange of technical information shall take place during open sessions of this meeting and UCF will not sponsor or support any closed or private working group, standards setting or development sessions that may take place during this meeting. Your participation in any non-public interactions or settings during this meeting are outside the scope of UCF's intended open-public meeting format.

■ Mission:

- Collaboration between industry, laboratories, and academia to create production grade communication frameworks and open standards for data centric and high-performance applications

■ Projects

- **UCX – Unified Communication X** – www.openucx.org
- SparkUCX – www.sparkucx.org
- Open RDMA

Join

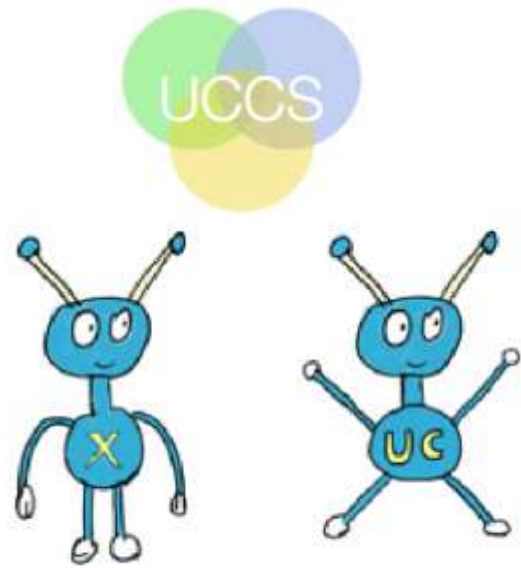
<https://www.ucfconsortium.org>
info@ucfconsortium.org

■ Board members

- **Jeff Kuehn**, UCF Chairman (Los Alamos National Laboratory)
- **Gilad Shainer**, UCF President (Mellanox Technologies)
- **Pavel Shamis**, UCF treasurer (Arm)
- **Brad Benton**, Board Member (AMD)
- **Duncan Poole**, Board Member (Nvidia)
- **Pavan Balaji**, Board Member (Argonne National Laboratory)
- **Sameh Sharkawi**, Board Member (IBM)
- **Dhabaleswar K. (DK) Panda**, Board Member (Ohio State University)
- **Steve Poole**, Board Member (Open Source Software Solutions)



UCX History



<https://www.hpcwire.com/2018/09/17/ucf-ucx-and-a-car-ride-on-the-road-to-exascale/>



Unified Communications X

An open-source, exascale-ready communications framework



- Solves decades-old problem in high-performance computing (HPC)
- Frees developers from hardware-specific implementations and laborious porting efforts
- Simplifies deployment of advanced research tools, regardless of system complexity
- Advances fields of artificial intelligence, machine learning, deep learning, and internet of things



Los Alamos
NATIONAL LABORATORY
EST. 1943

Advanced Micro Devices,
Argonne National Laboratory,
Arm Ltd., Mellanox Technologies,
NVIDIA, Stony Brook University,
Oak Ridge National Laboratory,
Rice University

- Support for x86_64, Power 8/9, Arm v8
- U-arch tuned code for Xeon, AMD Rome/Naples, Arm v8 (Cortex-A/N1/ThunderX2/Huawei)
- First class support for AMD and Nvidia GPUs
- Runs on Servers, Raspberry PI like platforms, SmartNIC, Nvidia Jetson platforms, etc.



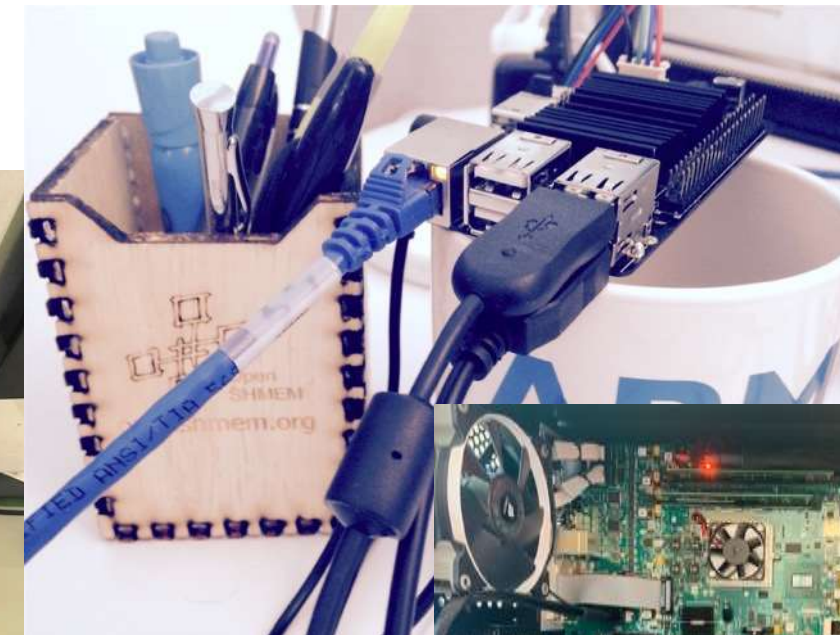
NVIDIA Jetson



Bluefield SmartNIC



Arm ThunderX2

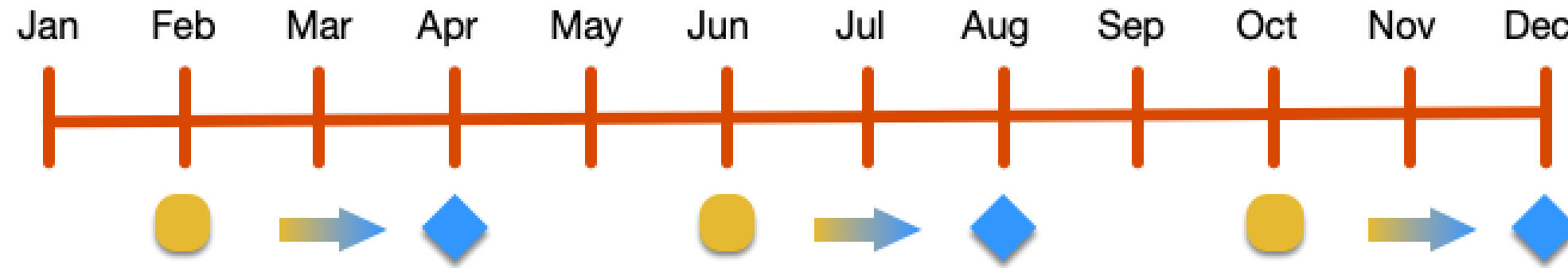


Odroid C2



N1 SDP

UCX annual release schedule



- v1.6.0 - July '19
- v1.6.1 - October '19
- v1.7.0 – End of November '19



V1.6.0

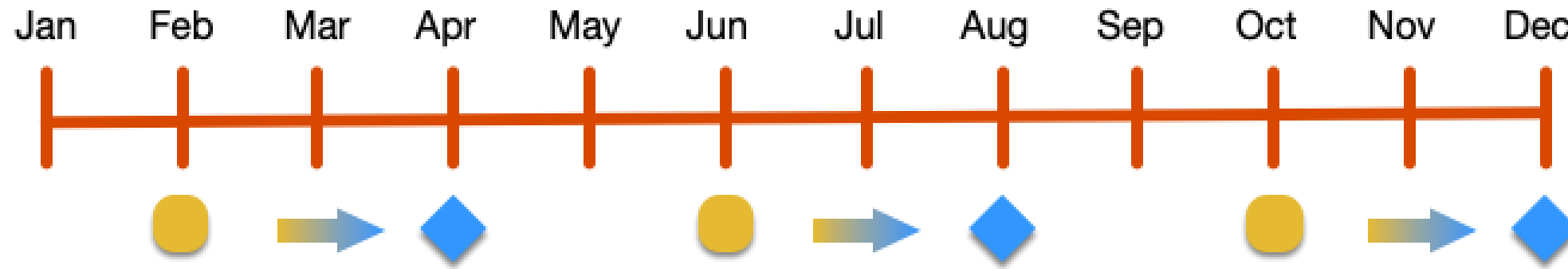
- AMD GPU ROCm transport re-design: support for managed memory, direct copy, ROCm GDR
- Modular architecture for UCT transports – runtime plugins
- Random scheduling policy for DC transport
- Improved support for Vebs API
- Optimized out-of-box settings for multi-rail
- Support for PCI atomics with IB transports
- Reduced UCP address size for homogeneous environments

V1.6.1

- Add Bull Atos HCA device IDs
- Azure Pipelines CI Infrastructure
- Clang static checker

- Added support for multiple listening transports
- Added UCT socket-based connection manager transport
- Updated API for UCT component management
- Added API to retrieve the listening port
- Added UCP active message API
- Removed deprecated API for querying UCT memory domains
- Refactored server/client examples
- Added support for dlopen interception in UCM
- Added support for PCIe atomics
- Updated Java API: added support for most of UCP layer operations
- Updated support for Mellanox DevX API
- Added multiple UCT/TCP transport performance optimizations
- Optimized memcpy() for Intel platforms

UCX annual release schedule



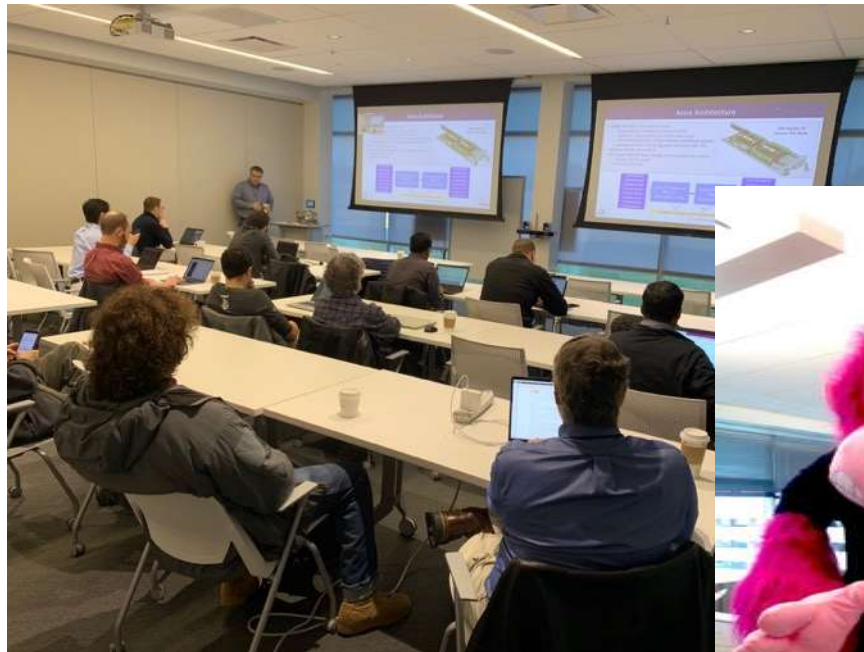
- v1.8.0 - April '20
- v1.9.0 - August '20
- v1.10.0 - December '20



- **UCX Python / UCP-Py**
 - <https://github.com/rapidsai/ucx-py>
 - Already integrated Dask and Rapids AI
- **UCX Java**
 - Java API official UCX release
- **Spark UCX**
 - www.sparkucx.org
 - <https://github.com/openucx/sparkucx>
- **Collective API**
 - For more details see collective API pull requests at github
- **Enhanced Active Message API**
- **FreeBSD and MacOS**
- Improved documentation

Save the date !

- UCX Hackathon: December 9-12 (Registration required)
 - <https://github.com/openucx/ucx/wiki/UCF-Hackathon-2019>
- Austin, TX



Laptop Stickers



- Arm
- Los Alamos National Laboratory
- Mellanox
- Argonne National Laboratory
- Stony Brook
- Charm++
- AMD
- ORNL
- Nvidia
- OSU

The background of the slide is a stylized, high-angle view of a city at night. The city lights are blurred into bokeh, with a mix of warm orange and yellow tones. Overlaid on this is a faint, light blue grid of plus signs (+) that covers the entire slide area.

arm

UCX Support for Arm

Pavel Shamis (Pasha), Principal Research Engineer

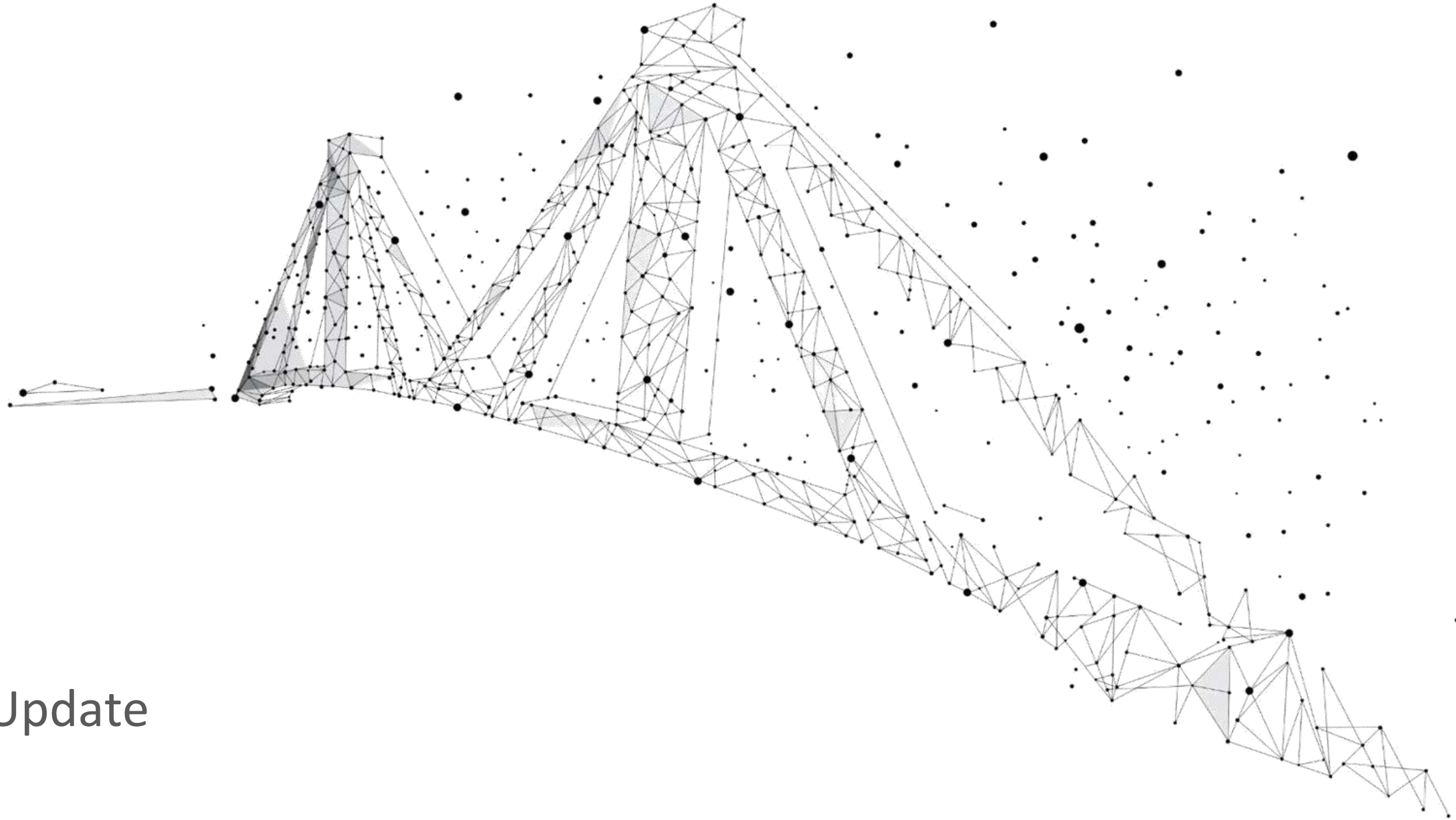
UCX on ThunderX2 (arm v8) at Scale



Recent Numbers: Arm + InfiniBand ConnectX6 2x200 Gb/s

	Put Ping-Pong Latency, 8B (usec)	Put Injection Rate, 8B (MPP/s)
UCX-UCT (low level), Accelerated	0.72	16.3
UCX-UCT (low level), Verbs	0.75	5.9
Verbs, IB_WRTIE_LAT/BW	0.95	32 (Post List, 2 QPs)
UCP, Verbs	1.07	5.6
UCP, Accelerated	0.81	15.9

- ConnectX-6 200Gbs X 2 / port 1<->2 (no switch)
- PCIe Gen4
- Internal UCX version (will be unstreamed)



UCX

Mellanox Update

November 2019



High-level overview

Applications

HPC (MPI, SHMEM, ...)

Storage, RPC, AI

Web 2.0 (Spark, Hadoop)

UCX

UCP – High Level API (Protocols)
Transport selection, multi-rail, fragmentation

HPC API:
tag matching, active messages

I/O API:
Stream, RPC, remote memory access, atomics

Connection establishment:
client/server, external

UCT – Low Level API (Transports)

RDMA

RC

DCT

UD

iWarp

GPU / Accelerators

CUDA

AMD/ROCM

Others

Shared
memory

TCP

OmniPath

Cray

OFA Verbs Driver

Cuda

ROCM

Hardware

UCX v1.7 Advantages

- UCX is the default communication substrate for HPC-X AND many open source HPC Runtimes
 - Open MPI and OSHMEM
 - HCOLL
 - BUPC (UCX conduit, WIP)
 - Charm++/UCX
 - NCCL/UCX – WIP
 - MPICH
 - SparkUCX - WIP
- Java and Python bindings
- Full support for GPU Direct RDMA, GDR copy, and CUDA IPC
- CUDA aware - three-stage pipelining protocol
- Support for HCA atomics including new bitwise atomics
- Shared memory (CMA, KNEM, xpmem, SysV, mmap)
- Support for non-blocking memory registration and On-Demand Paging ODP
- Multithreaded support
- Support for hardware tag matching
- Support for multi-rail and socket direct
- Support for PCIe atomics
- Support for MEMIC – HCA memory
- In-box in many Linux distros, and more to come

**R&D
100**

Unified Communication X

UNIFYING Network and GPU Communication Seamlessly,
Transparently, Elegantly

Open MPI / UCX

CUDA aware UCX

- CUDA Aware Tag API for data movement on GPU clusters
- Intra-node GPU-GPU, GPU-HOST, HOST-GPU
- Inter-node GPU-GPU, GPU-HOST, HOST-GPU
- Optimal protocols
 - GPUDirectRDMA
 - CUDA-IPC
 - GDRCOPY
 - Pipelining
- Efficient memory type detection

Performance: System and Software description

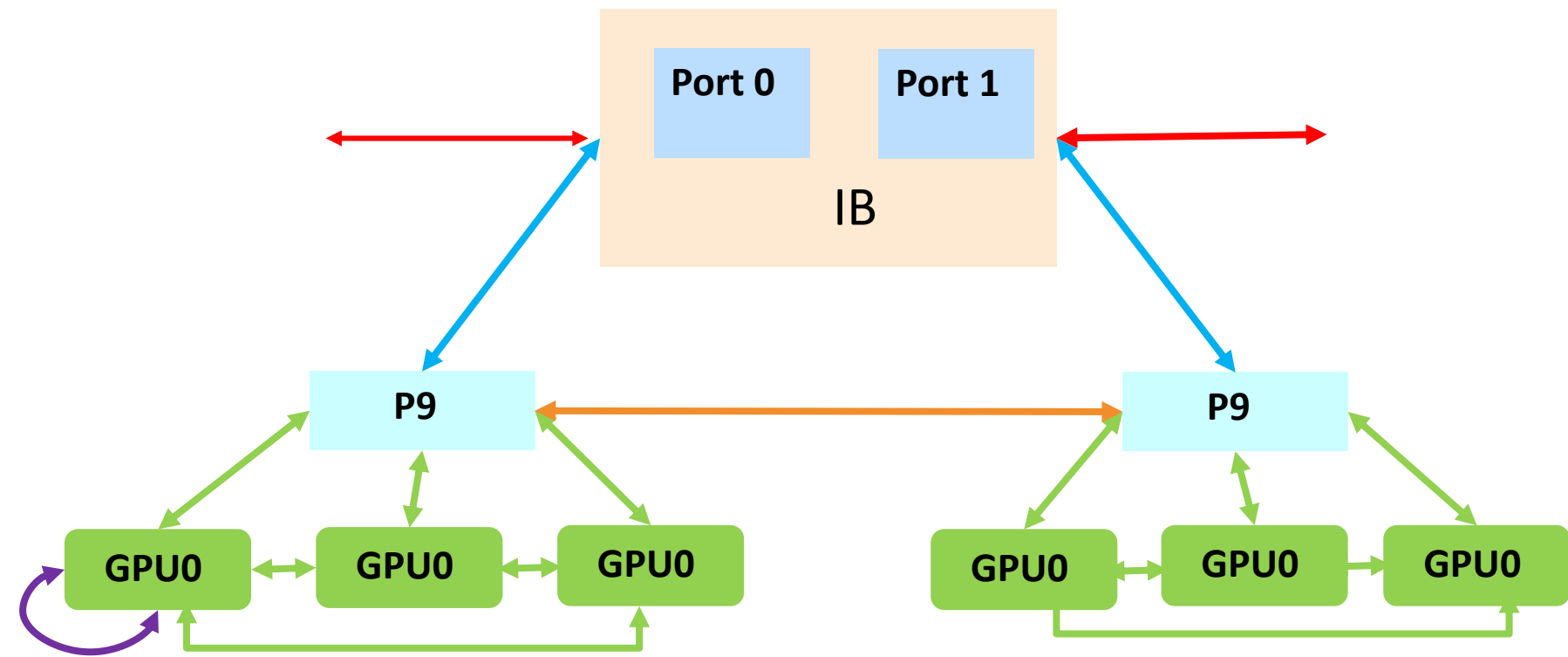
Hardware






- Summit Supercomputer
 - GPU: 6 x Tesla V100 NVLink
 - HCA: ConnectX-5
 - CPU: PowerPC

Software

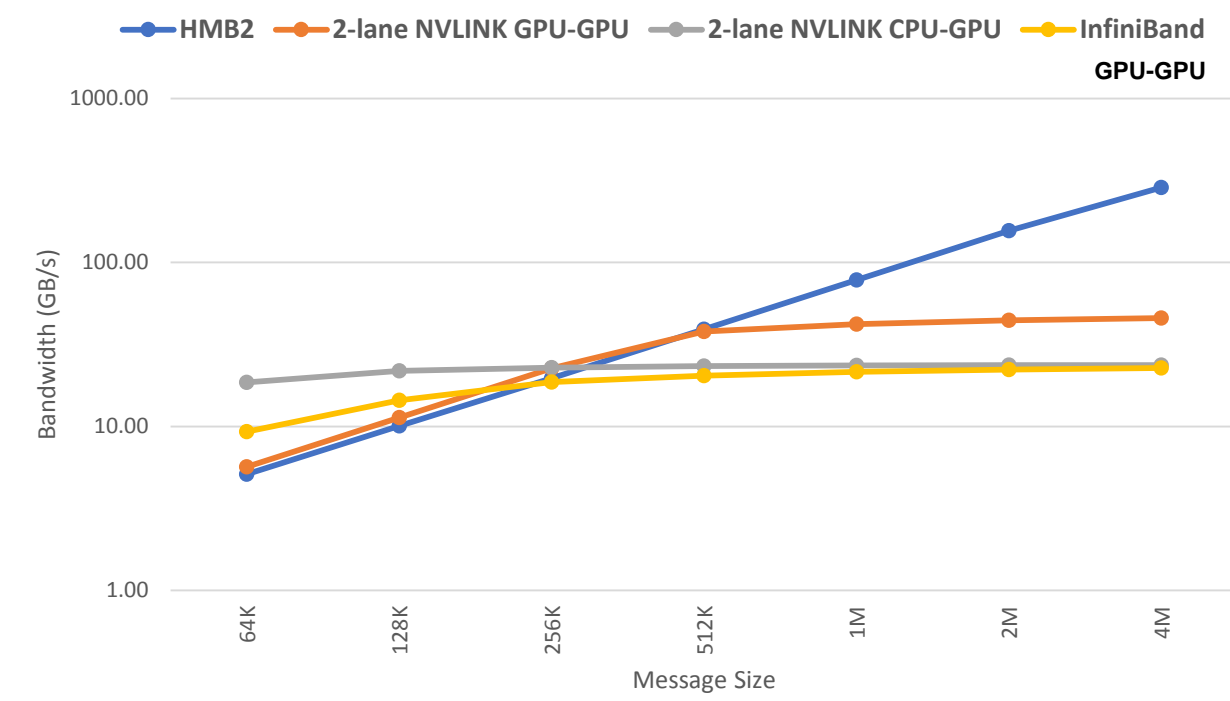
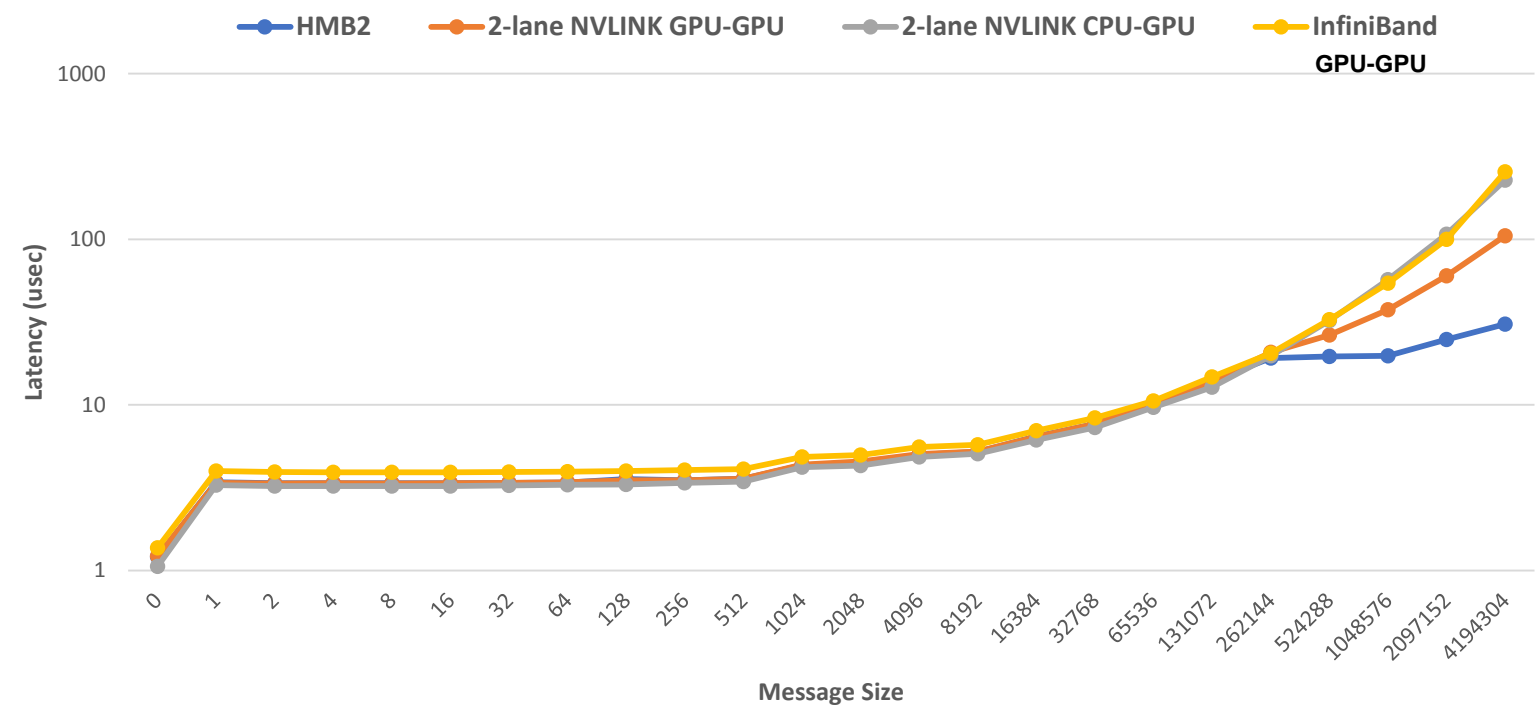
- CUDA 10.1
- OpenMPI-4.0.2
- UCX 1.7
 - Tuning
 - UCX_RNDV_SCHEME=get_zcopy
 - UCX_RNDV_THRESH=1

System Architecture



-  2 lane NVLink: Between GPU-GPU and CPU_GPU
-  8 lane PCIe Gen4
-  EDR Infiniband Interconnect
-  Inter processor X-Bus
-  HBM2

OMB: Latency & Bandwidth



Performance Summary



	GPU HBM2	2-Lane NVLink GPU-GPU	2-Lane NVLink CPU-GPU	IB EDR x2 GPU-GPU
Theoretical Peak BW	900 GB/s	50 GB/s	50 GB/s	25 GB/s
Available Peak BW	723.97 GB/s	46.88 GB/s	46 GB/s	23.84 GB/s
UCX Peak BW	349.6 GB/s	45.7 GB/s	23.7 GB/s	22.7 GB/s
% Peak	48.3%	97.5%	51.5%	95.2%

MPICH / UCX

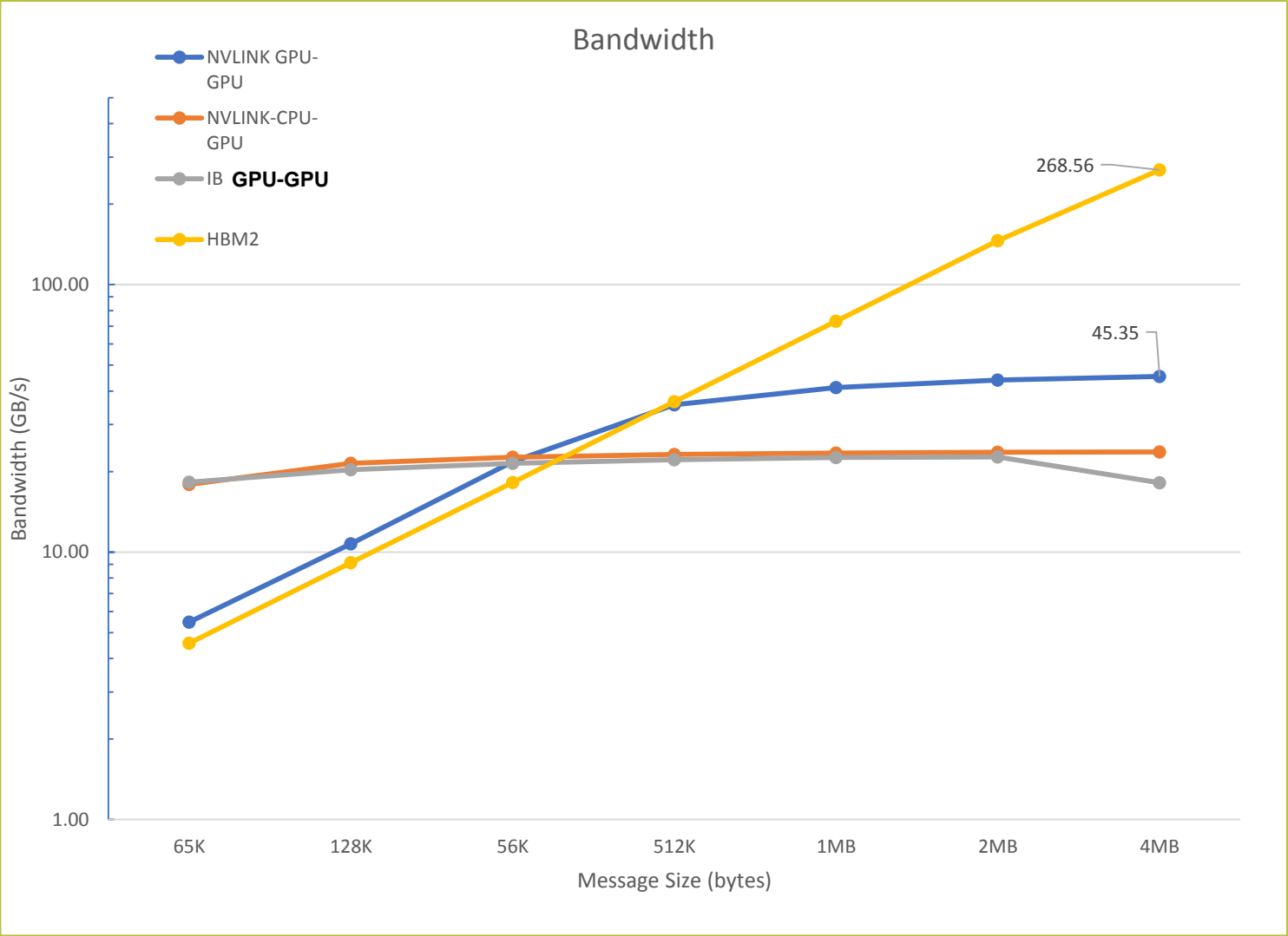
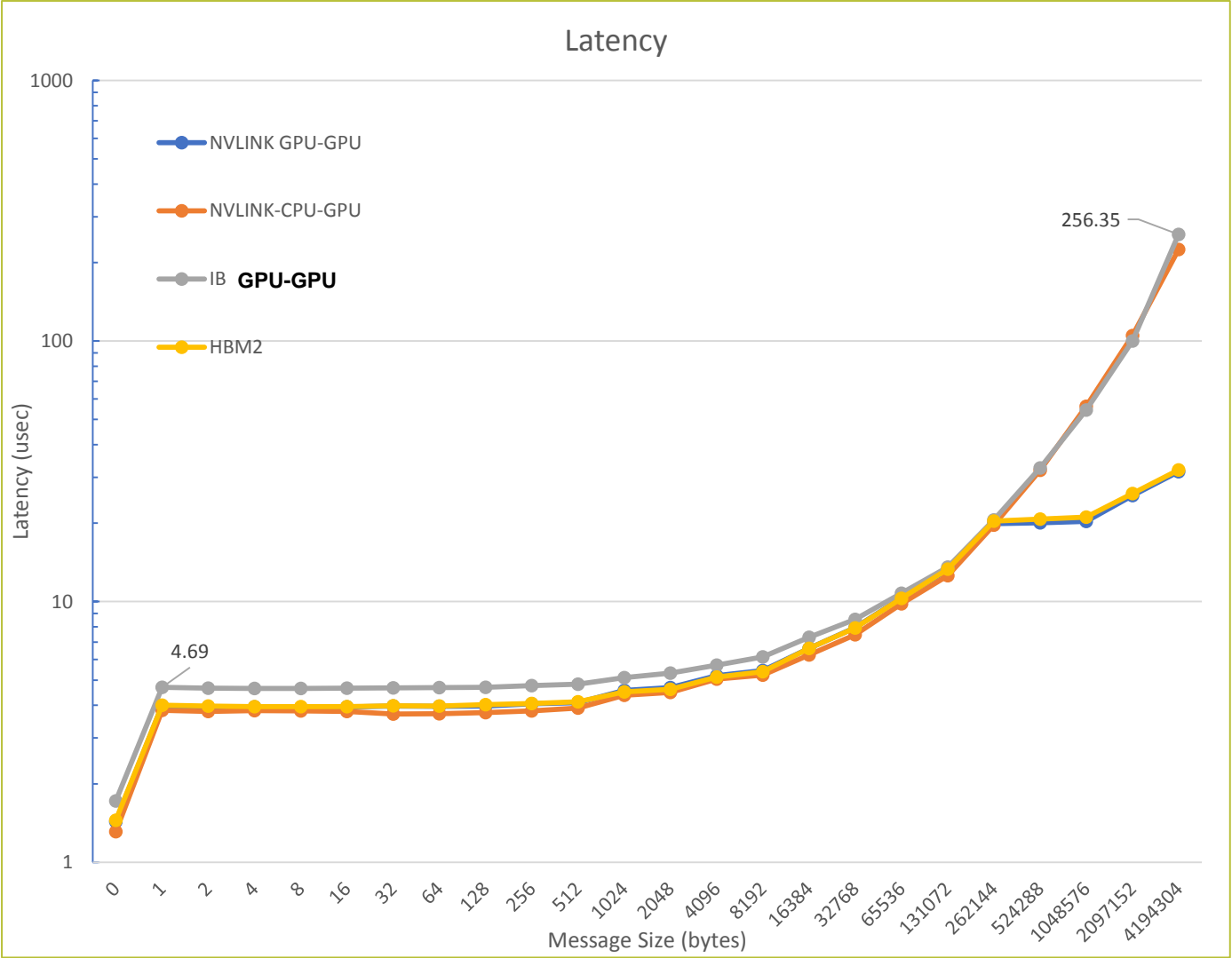
Performance: System and Software description

Hardware

- Summit Supercomputer
 - GPU: 6 x Tesla V100 NVLink
 - HCA: ConnectX-5
 - CPU: PowerPC

Software

- CUDA 10.1
- MPICH
 - master branch (12b9d564fa13d12d11463d4405db04d3354314f3)
- UCX 1.7
 - Tuning
 - UCX_RNDV_SCHEME=get_zcopy
 - UCX_RNDV_THRESH=1



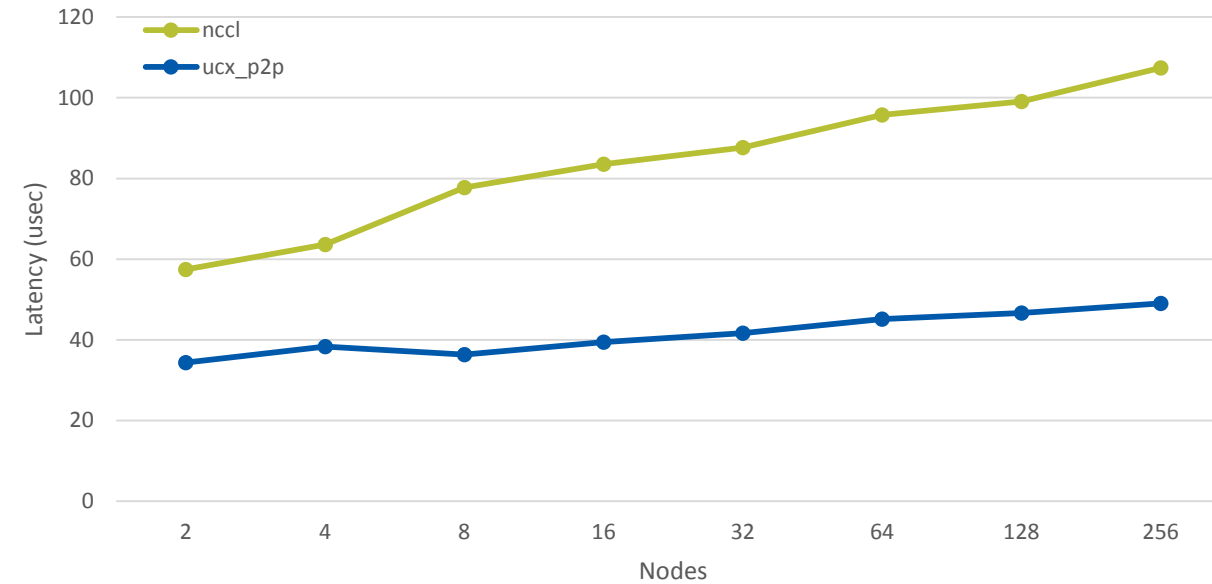
Performance Summary



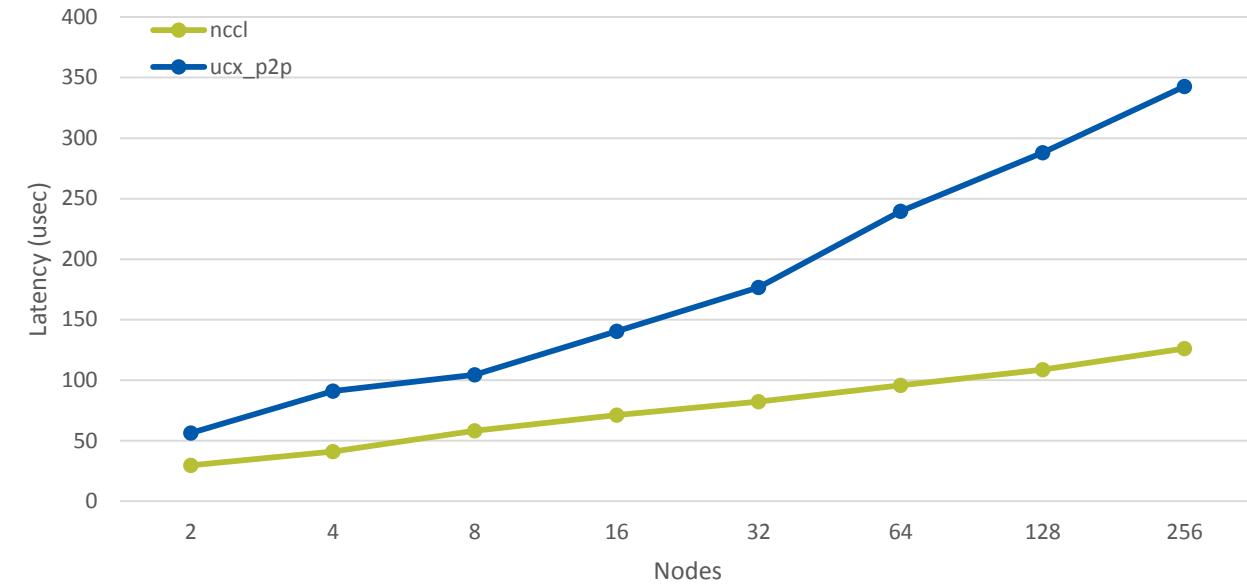
	GPU HBM2	2-Lane NVLink GPU-GPU	2-Lane NVLink CPU-GPU	IB EDR x2 GPU-GPU
Theoretical Peak BW	900	50	50	25
Available Peak BW	723.97	46.88	46	23.84
UCX Peak BW	346.6	46.6	23.7	22.6
% Peak	47.87%	99.40%	51.52%	94.80%

Collective: osu_allreduce

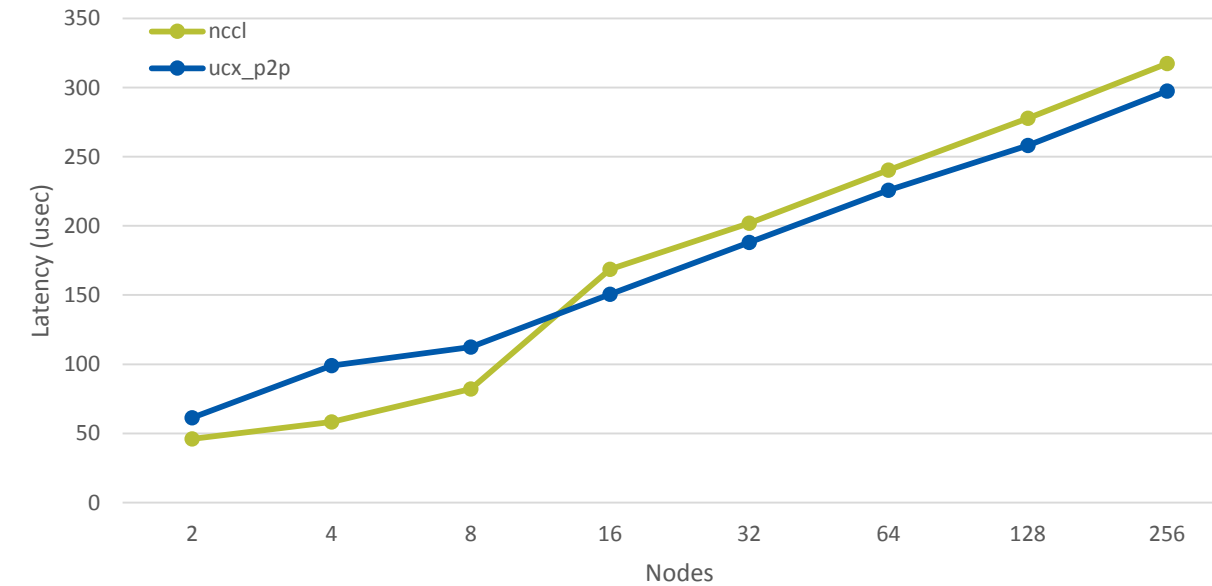
4B, osu_allreduce -cuda; MPICH w/HCOLL



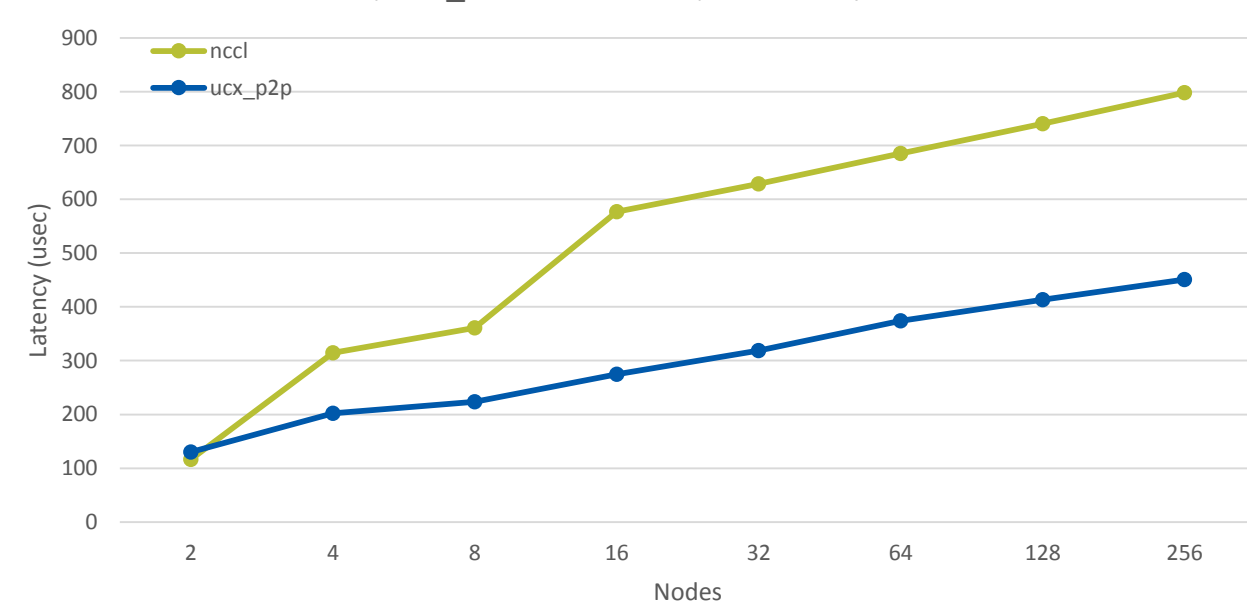
4K, osu_allreduce -cuda; MPICH w/HCOLL



32K, osu_allreduce -cuda; MPICH w/HCOLL



1MB, osu_allreduce -cuda; MPICH w/HCOLL



NCCL / UCX

NCCL Inter-Node P2P Communication Plugin

Key Features

- Replaces **inter-node** point-to-point communication
- Dedicated API exposed by NCCL
- Three-phased communication
 - Connection establishment
 - Data transfer
 - Connection Closure

API

- Connection establishment
 - **Listen, Connect, Accept**
- Data transfer (Non-Blocking)
 - **Send, Receive, Test**
- Connection Closure
 - **CloseListen, CloseSend, CloseReceive**

System and Software description

Hardware

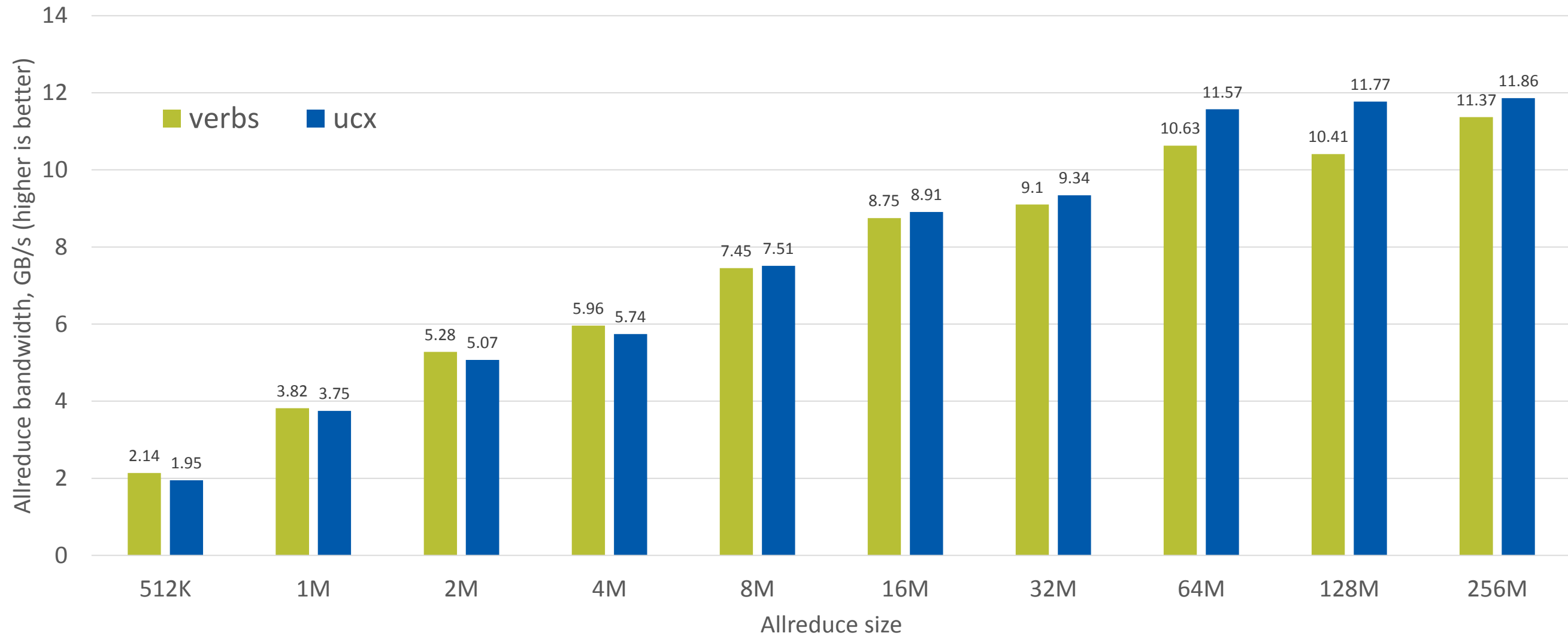
- 32 Nodes of Summit Supercomputer
 - GPU: 6 x Tesla V100 NVLink
 - HCA: 1 x ConnectX-5
 - CPU: PowerPC

Software

- Nvidia NCCL 2.4.7
- CUDA 10.1
- UCX 1.7
 - Tuning
 - UCX_RNDV_SCHEME=get_zcopy
 - UCX_RNDV_THRESH=1
- NCCL UCX plugin – HPC-X v2.6 preview

NCCL Internal Verbs vs NCCL UCX Plugin

- UCX plugin outperforms NCCL Verbs implementation up to 13% on large messages



Unified Communication X



Scaling Microsoft Azure HPC with HPC-X/UCX

Azure HPC HBv2 virtual machines for HPC

- State of the art VMs feature a wealth of new technology, including:
 - AMD EPYC 7742 CPUs (Rome)
 - 2.45 GHz Base clock / 3.3 GHz Boost clock
 - 480 MB L3 cache, 480 GB RAM
 - 340 GB/s of Memory Bandwidth
 - 200 Gbps HDR InfiniBand (SRIOV) with Adaptive Routing
 - 900 GB SSD (NVMeDirect)

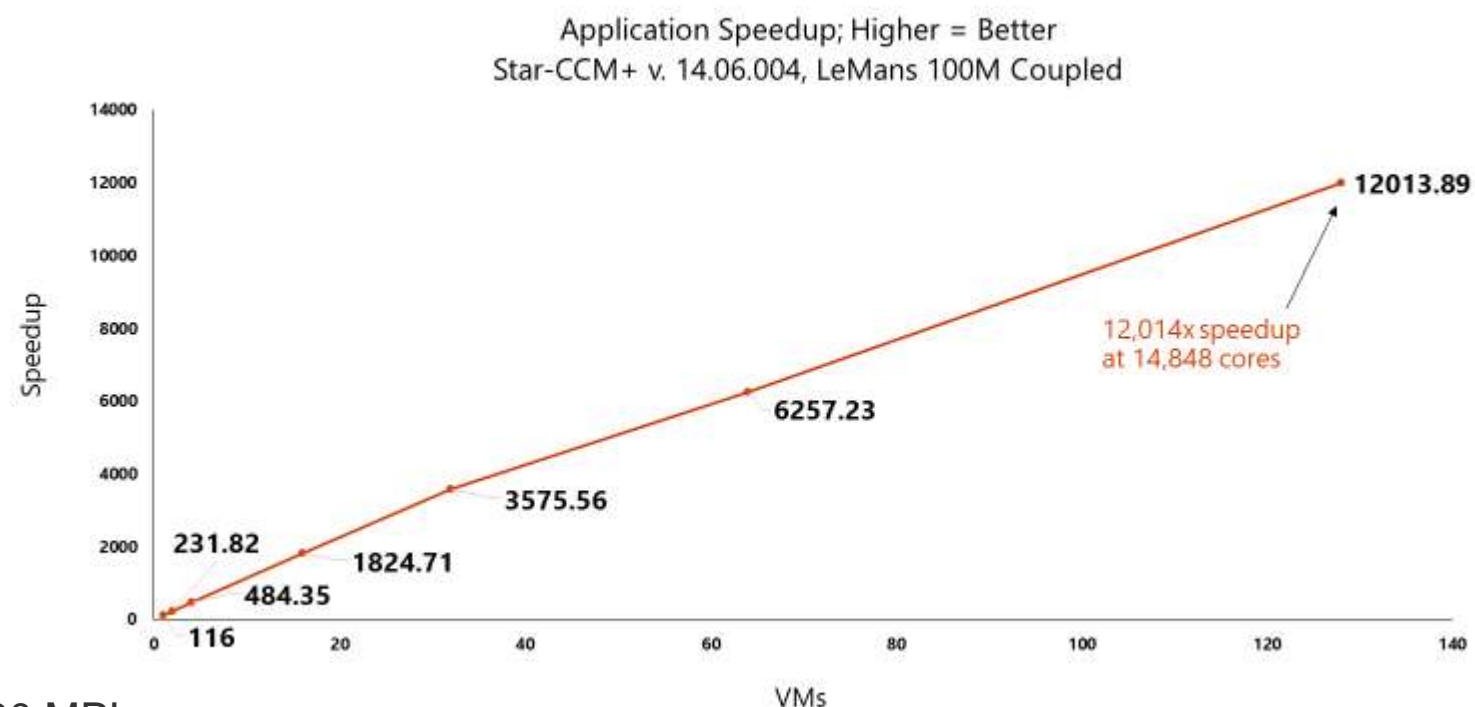
Star-CCM + on HBv2

App: Siemens Star-CCM+

Version: 14.06.004

Model: LeMans 100M Coupled Solver

Configuration Details: 116 MPI ranks were run (4 ranks from each of 29 NUMA) in each HBv2 VM in order to leave nominal resources to run Linux background processes. In addition, Adaptive Routing was enabled and DCT (Dynamic Connected Transport) was used as the transport layer, while HPC-X version 2.50 (UCX v1.6) was used for MPI. Azure CentOS HPC 7.6 image was used from <https://github.com/Azure/azhpc-images>



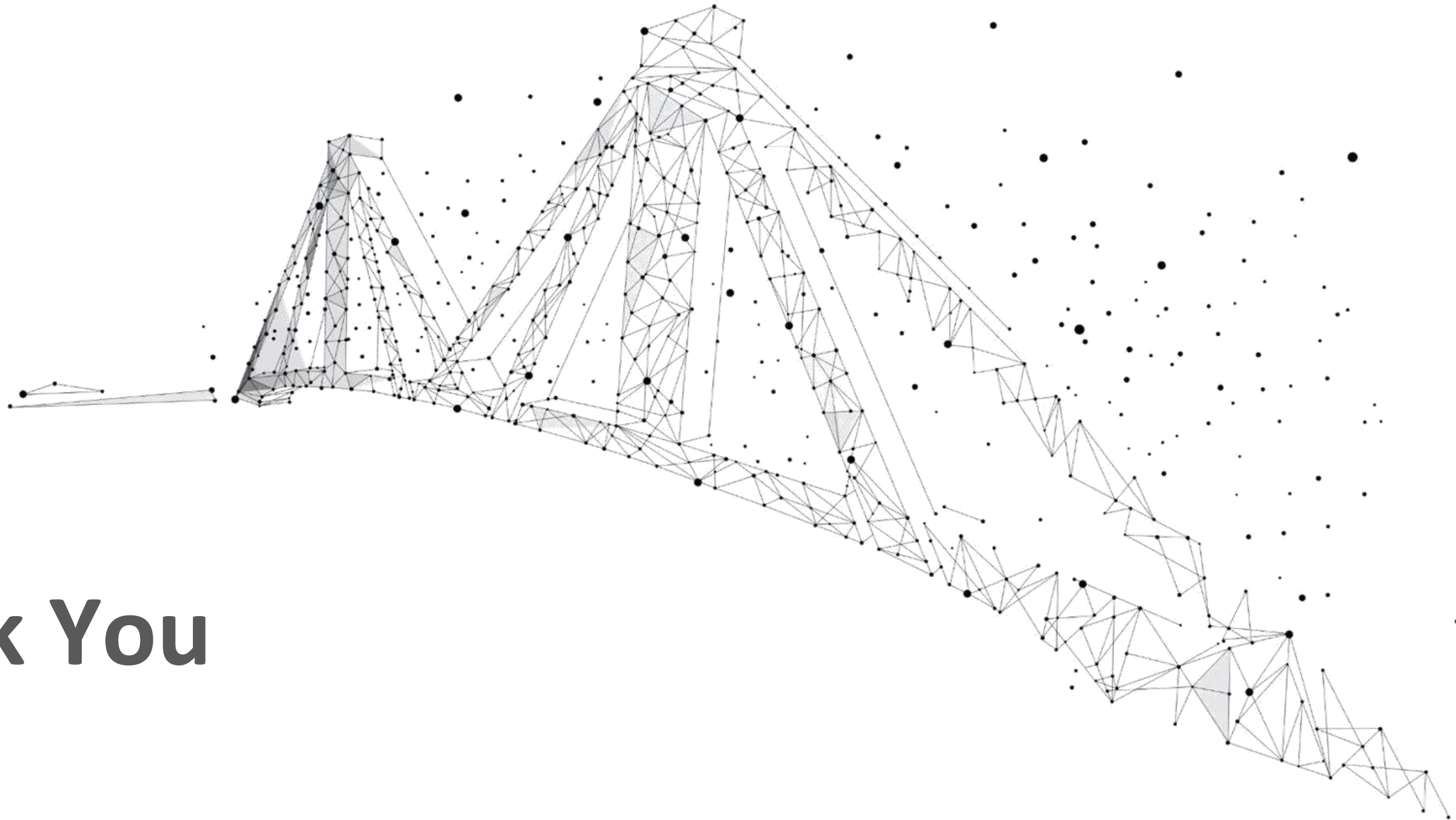
Summary: Star-CCM+ was scaled at 81% efficiency to nearly 15,000 MPI ranks delivering an application speedup of more than 12,000x. This compares favorably to Azure's previous best of more than 11,500 MPI ranks, which itself was a [world-record for MPI scalability on the public cloud](#).

ANSYS Fluent on HBv2

- **App:** ANSYS Fluent
- **Version:** 14.06.004
- **Model:** External Flow over a Formula-1 Race Car (f1_racecar_140m)
- **Configuration Details:** 60 MPI ranks were run (2 out of 4 cores per NUMA) in each HBv2 VM in order to leave nominal resources to run Linux background processes and give ~6 GB/s of memory bandwidth per core. In addition, Adaptive Routing was enabled and DCT (Dynamic Connected Transport) was used as the transport layer, while **HPC-X version 2.50 (UCX v1.6) was used for MPI**. Azure CentOS HPC 7.6 image was used from <https://github.com/Azure/azhpc-images>
- **Summary:** HBv2 VMs scale super linearly (112%) up to the top end measured number of VMs (128). The Fluent Solver Rating measured at this top-end level of scale is 83% more performance than the current leader submission on ANSYS public database for this model (<https://bit.ly/2OdAExM>).



Microsoft Azure



Thank You



UCX Support in MPICH

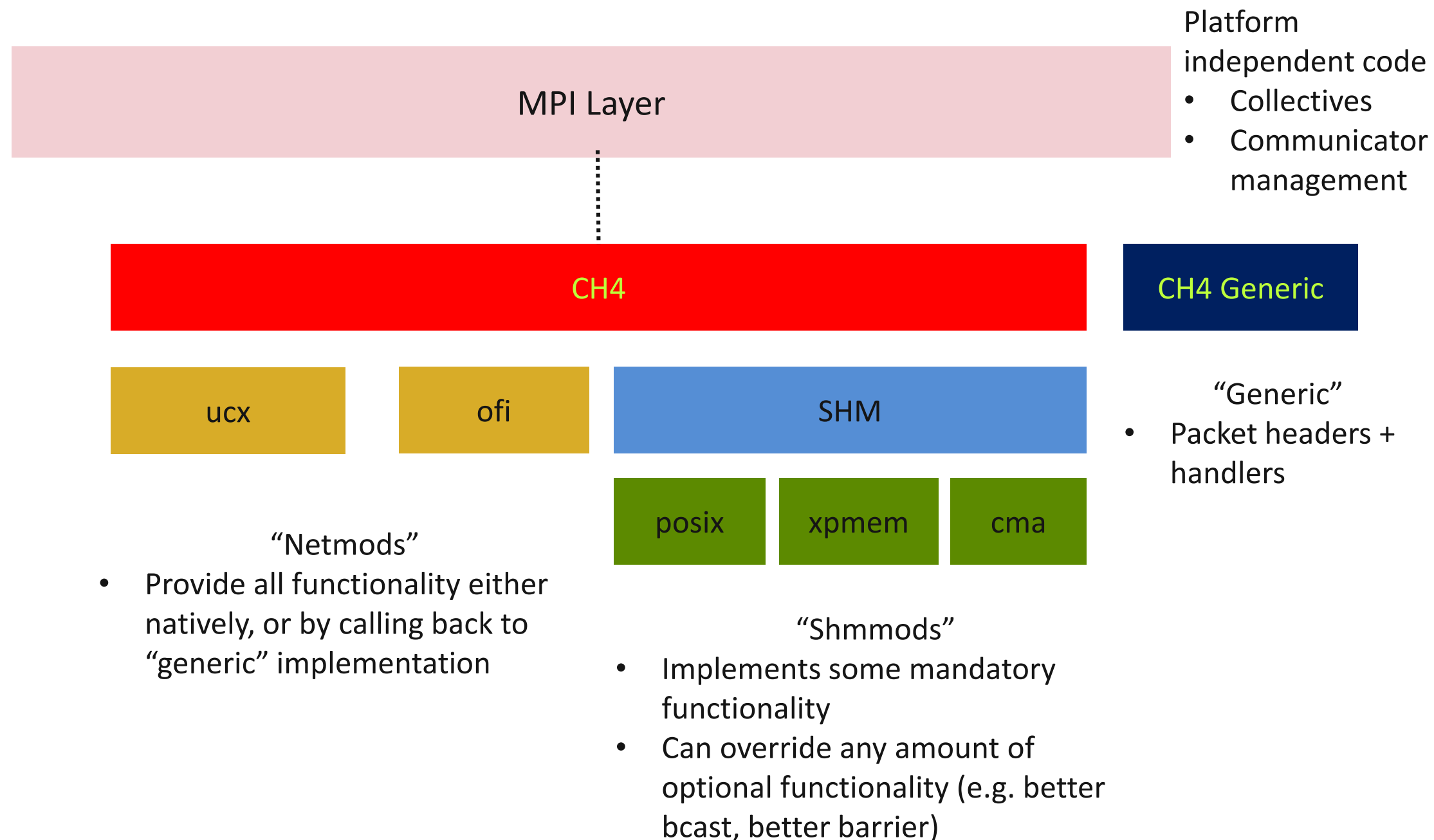
Yanfei Guo

Assistant Computer Scientist

Argonne National Laboratory

Email: yguo@anl.gov

MPICH layered structure: CH4



Benefit of using UCX in MPICH

- Separating general optimizations and device specific optimizations
 - Lightweight and high-performance communication
 - Native communication support
 - Simple and easy to maintain
 - MPI can benefit from new hardware quicker
- Better hardware support
 - Accelerated verbs with Mellanox hardware
 - Support for GPUs

MPICH/UCX with Accelerated Verbs

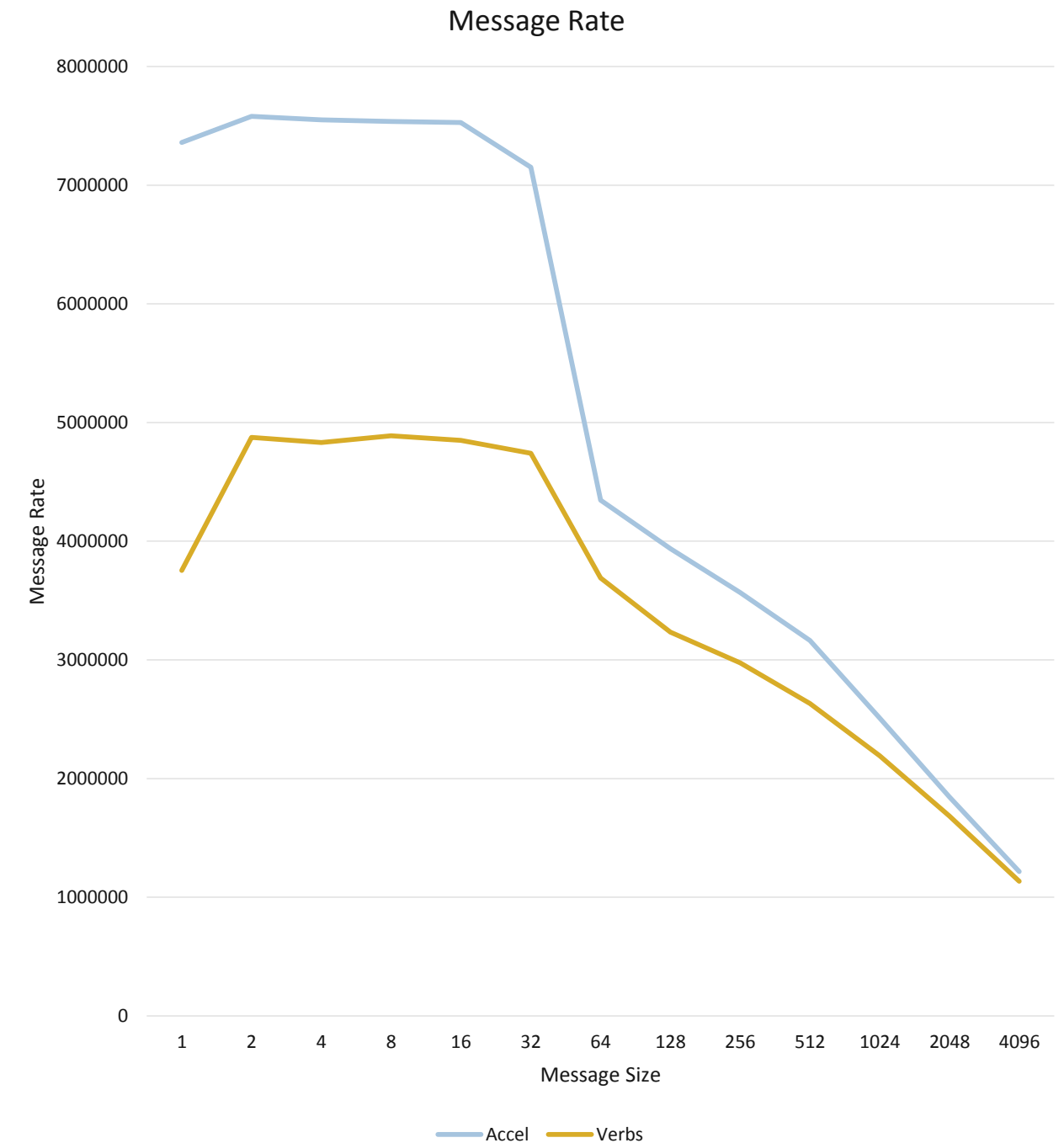
- UCX_TLS=rc_mlx5,cm
- Lower overhead
 - Low latency
 - Higher message rate

OSU Latency: **0.99us**

OSU BW: **12064.12 MB/s**

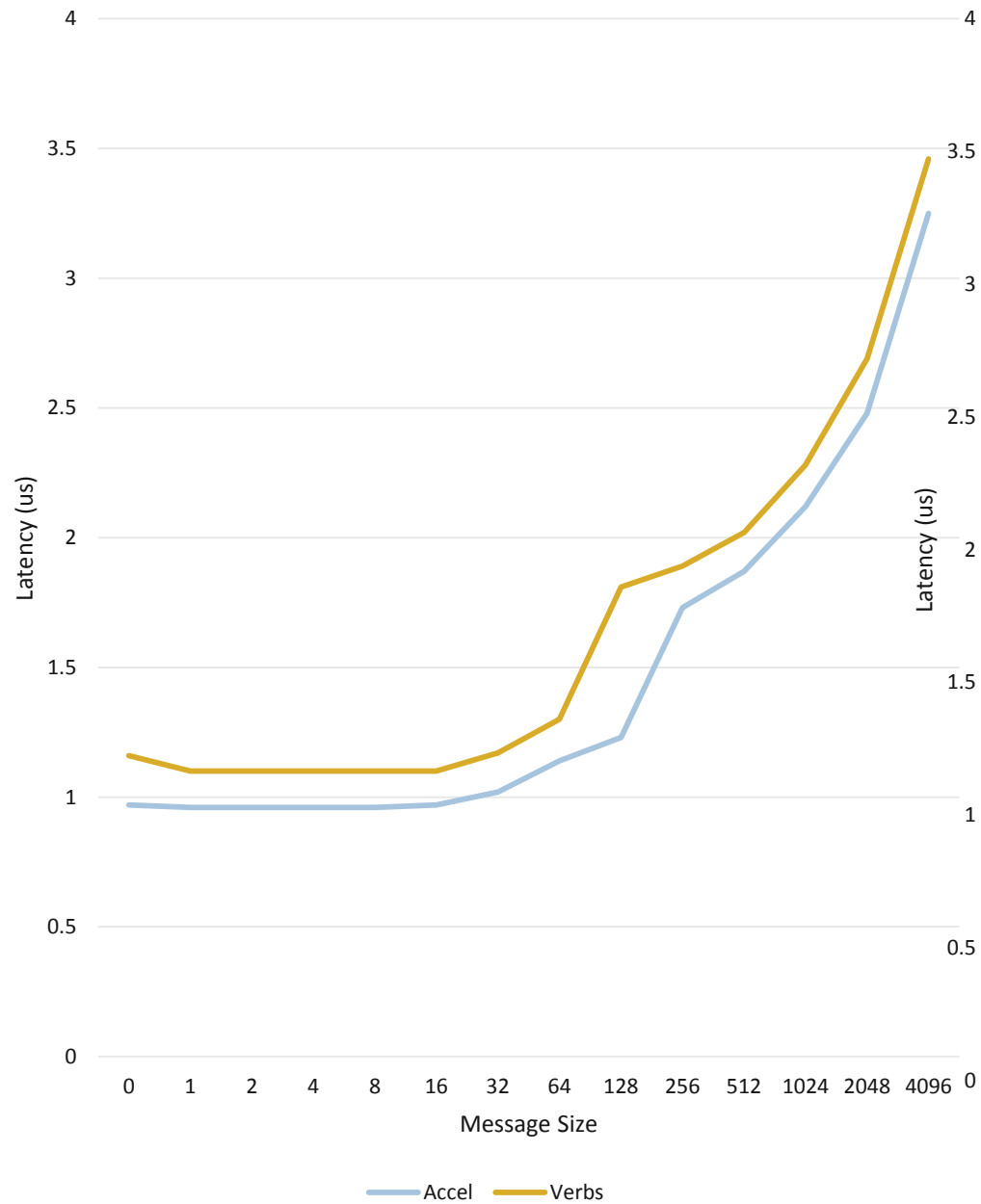
Argonne JLSE Thing Cluster

- Intel E5-2699v3 @ 2.3 GHz
- Connect-X 4 EDR
- HPC-X 2.2.0, OFED 4.4-2.0.7

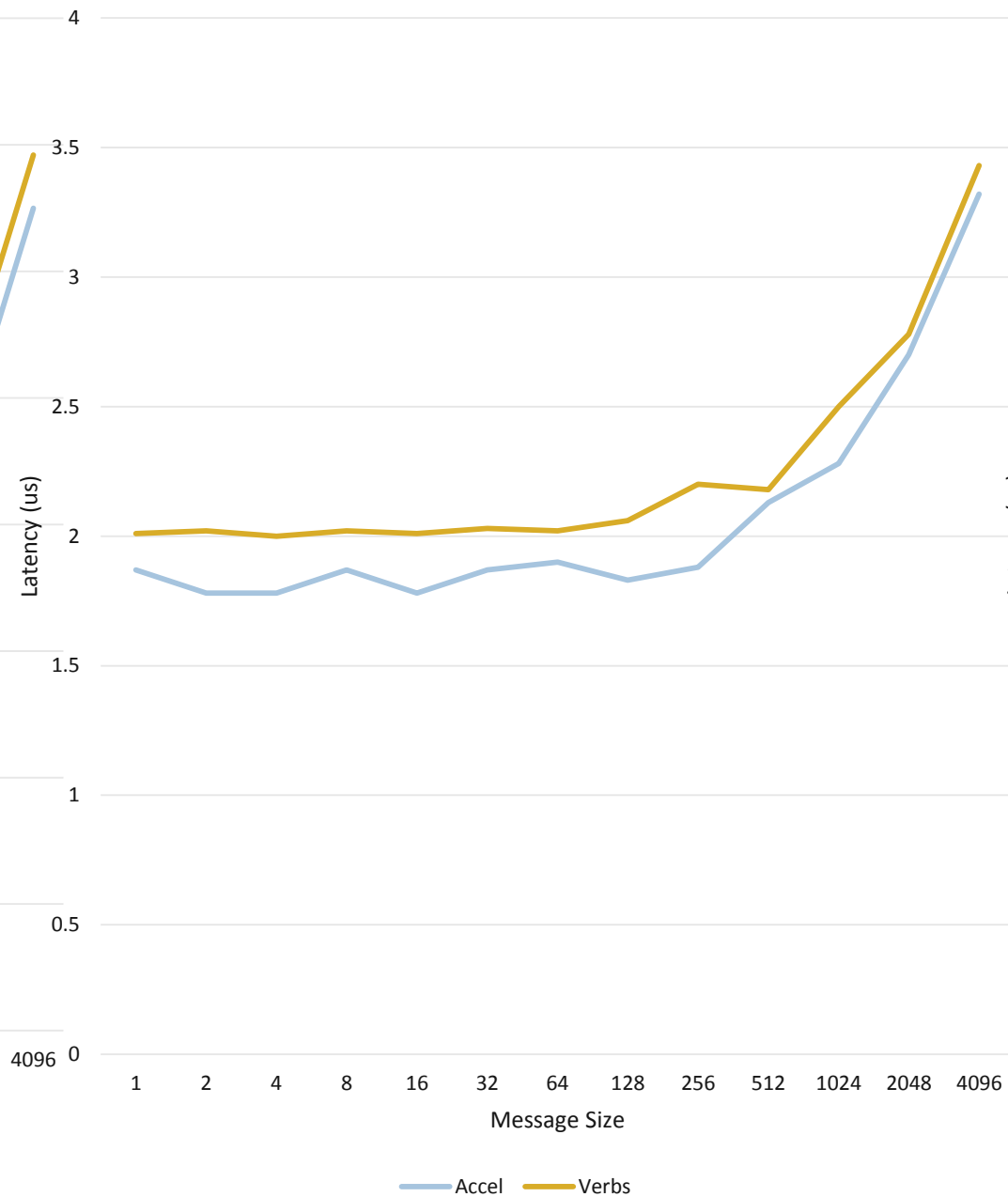


MPICH/UCX with Accelerated Verbs

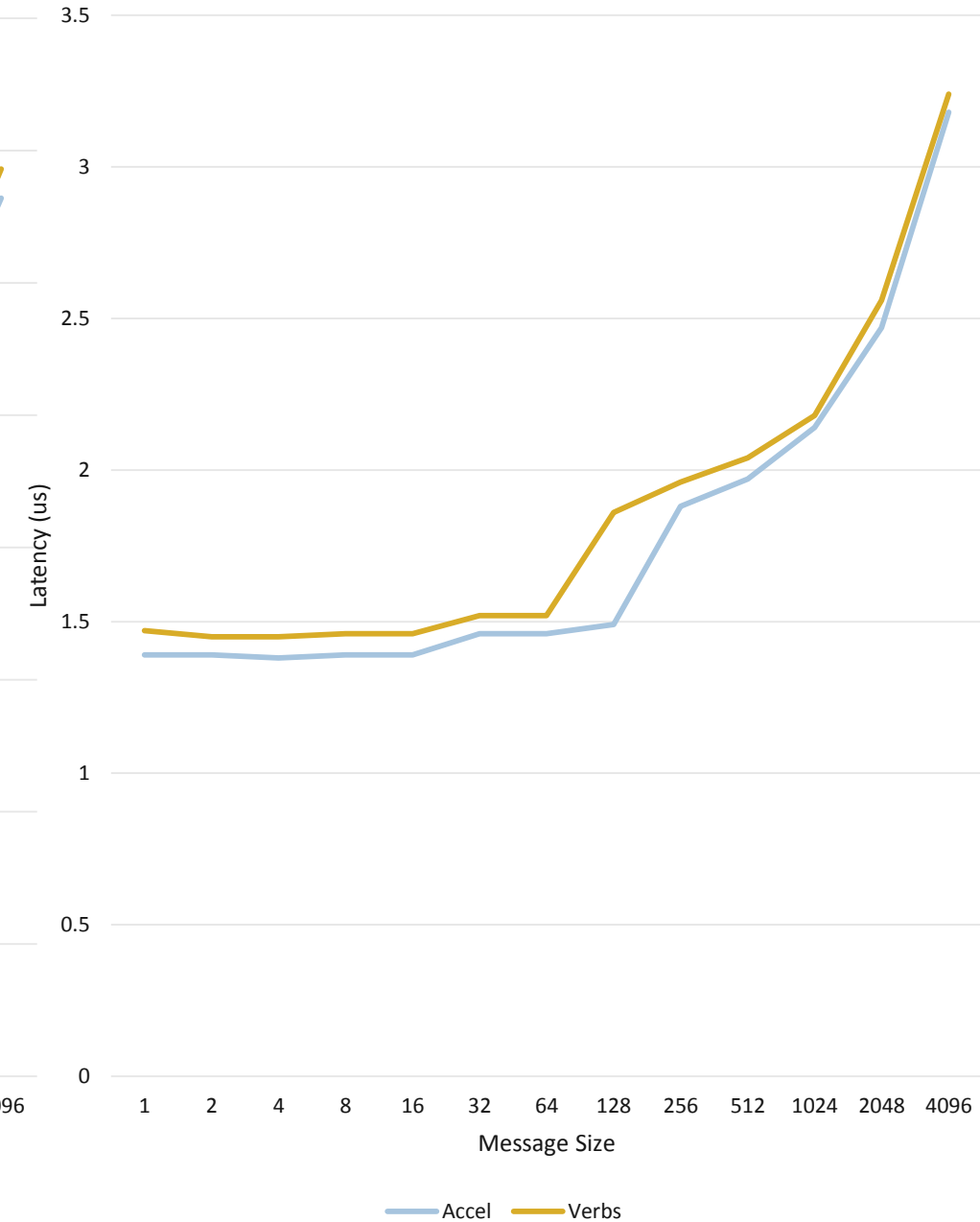
pt2pt latency



MPI_Get Latency



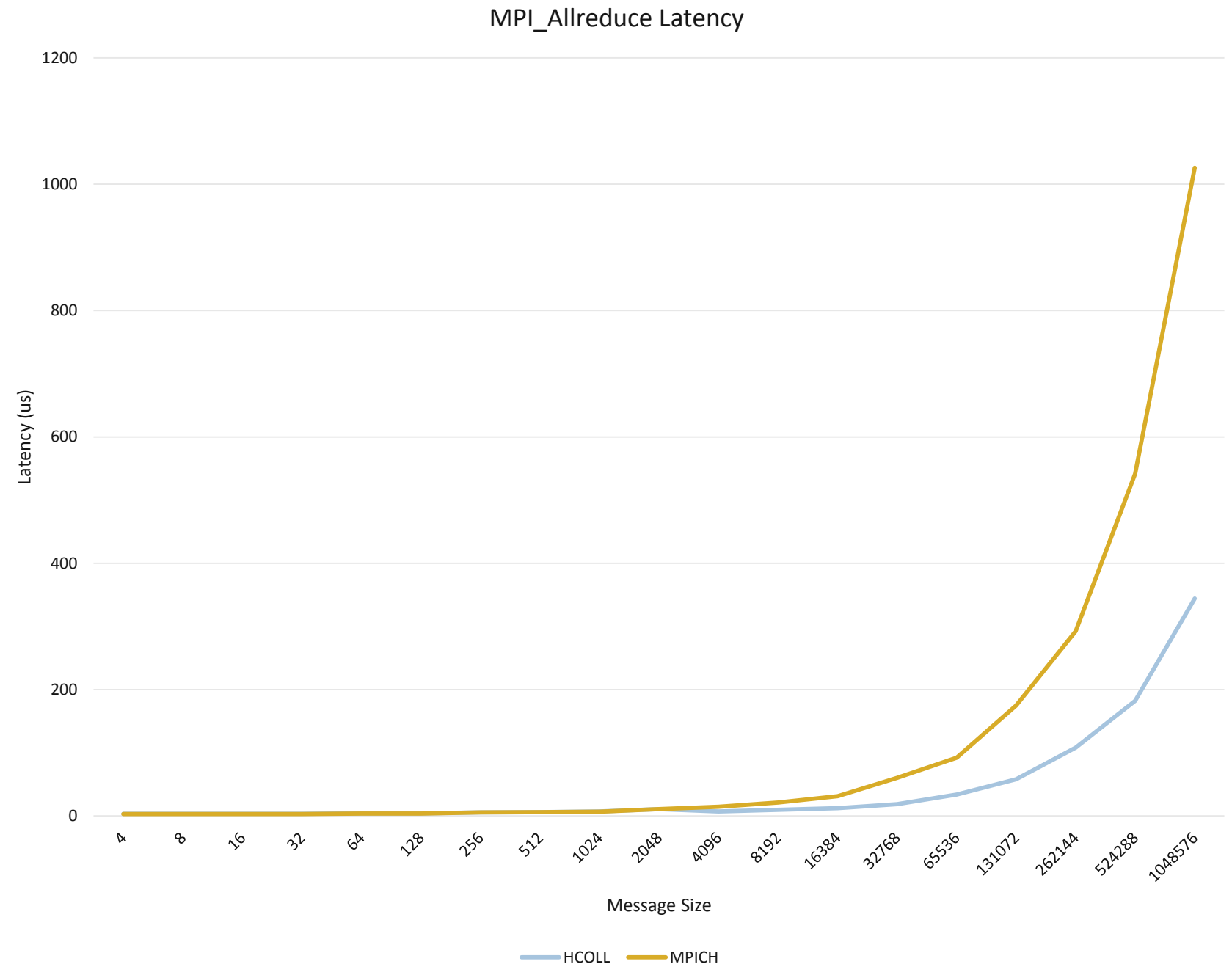
MPI_Put Latency



MPICH/UCX with HCOLL

Argonne JLSE Thing Cluster

- Intel E5-2699v3 @ 2.3 GHz
 - Connect-X 4 EDR
 - HPC-X 2.2.0, OFED 4.4-2.0.7
- 6 nodes, ppn=1



UCX Support in MPICH

- UCX Netmod Development
 - MPICH Team
 - Mellanox
 - NVIDIA
- MPICH 3.3.2 just released, 3.4a2 coming soon
 - Includes an embedded UCX 1.6.1
- Native path
 - pt2pt (with pack/unpack callbacks for non-contig buffers)
 - contiguous put/get rma for win_create/win_allocate windows
- Emulation path is CH4 active messages (hdr + data)
 - Layered over UCX tagged API
- Not yet supported
 - MPI dynamic processes

Hackathon on MPICH/UCX

- Earlier Hackathons with Mellanox
 - Full HCOLL and UCX integration in MPICH 3.3
 - Including HCOLL non-contig datatypes
 - MPICH CUDA support using UCX and HCOLL, tested and documented
 - <https://github.com/pmodels/mpich/wiki/MPICH-CH4:UCX-with-CUDA-support>
 - Support for FP16 datatype (non-standard, MPIX)
 - IBM XL and ARM HPC Compiler support
 - Extended UCX RMA functionality, under review
 - <https://github.com/pmodels/mpich/pull/3398>

Upcoming plans

- Native UCX atomics
 - Enable when user supplies certain info hints
 - <https://github.com/pmodels/mpich/pull/3398>
- Extended CUDA support
 - Handle non-contig datatypes
 - <https://github.com/pmodels/mpich/pull/3411>
 - <https://github.com/pmodels/mpich/issues/3519>
- Better MPI_THREAD_MULTIPLE support
 - Utilizing multiple workers (Rohit looking into this now)
- Extend support for FP16
 - Support for C_Float16 available in some compilers (MPIX_C_FLOAT16)
 - Missing support when GPU/Network support FP16 but CPU does not

MPICH

- <http://github.com/pmodels/mpich>
 - Submit an issue or pull request!
- Schedule a hackathon

Enabling Low-Overhead Multi-threaded
Communication in OpenSHMEM using UCX

Wenbin Lu
Tony Curtis
Barbara Chapman

UCX BoF SC19, Denver, CO

November 19th, 2019



UCX Networking Layer for Charm++

UCX Community BoF

SC 19

Nitin Bhat
Software Engineer
Charmworks Inc.



What is Charm++?

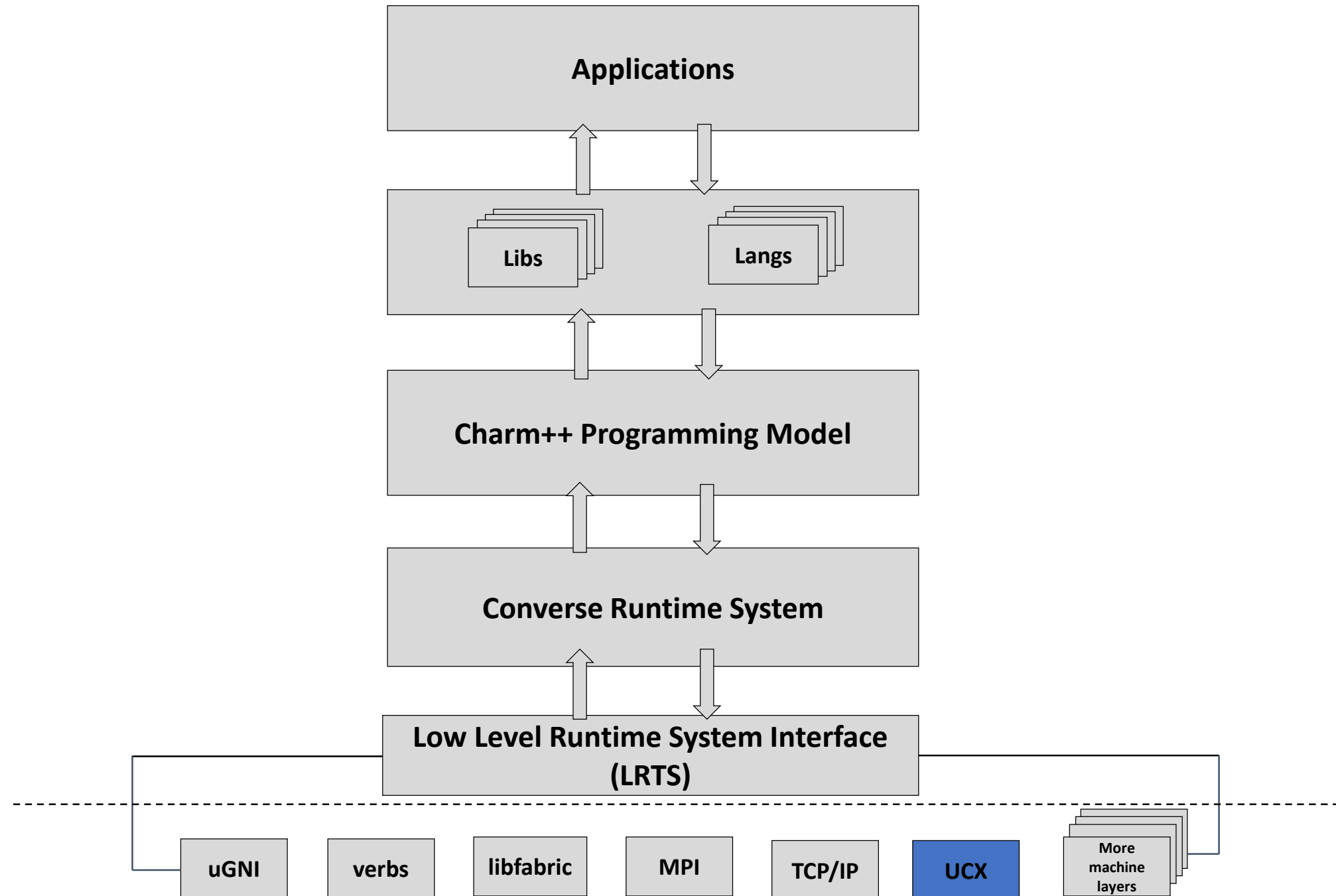
- Charm++ is a generalized approach to writing parallel programs
 - An alternative to the likes of MPI, UPC, GA, etc.
 - But not sequential languages such as C, C++, and Fortran
- Represents:
 - The style of writing parallel programs
 - The runtime system
 - And the entire ecosystem that surrounds it
- Three design principles:
 - Over-decomposition, Migratability, Asynchrony
- Enables:
 - Load Balancing, Shrink Expand, Fault Tolerance



Why we needed a new layer?

- Verbs layer was difficult to maintain/not working on new InfiniBand machines
- MPI layer was not scaling well
- We are interested in the runtime (and not so much on networking layers) so we wanted a portable and performant layer
- UCX offered
 - Portability
 - High performance
 - Ease of maintenance

Charm++ Architecture



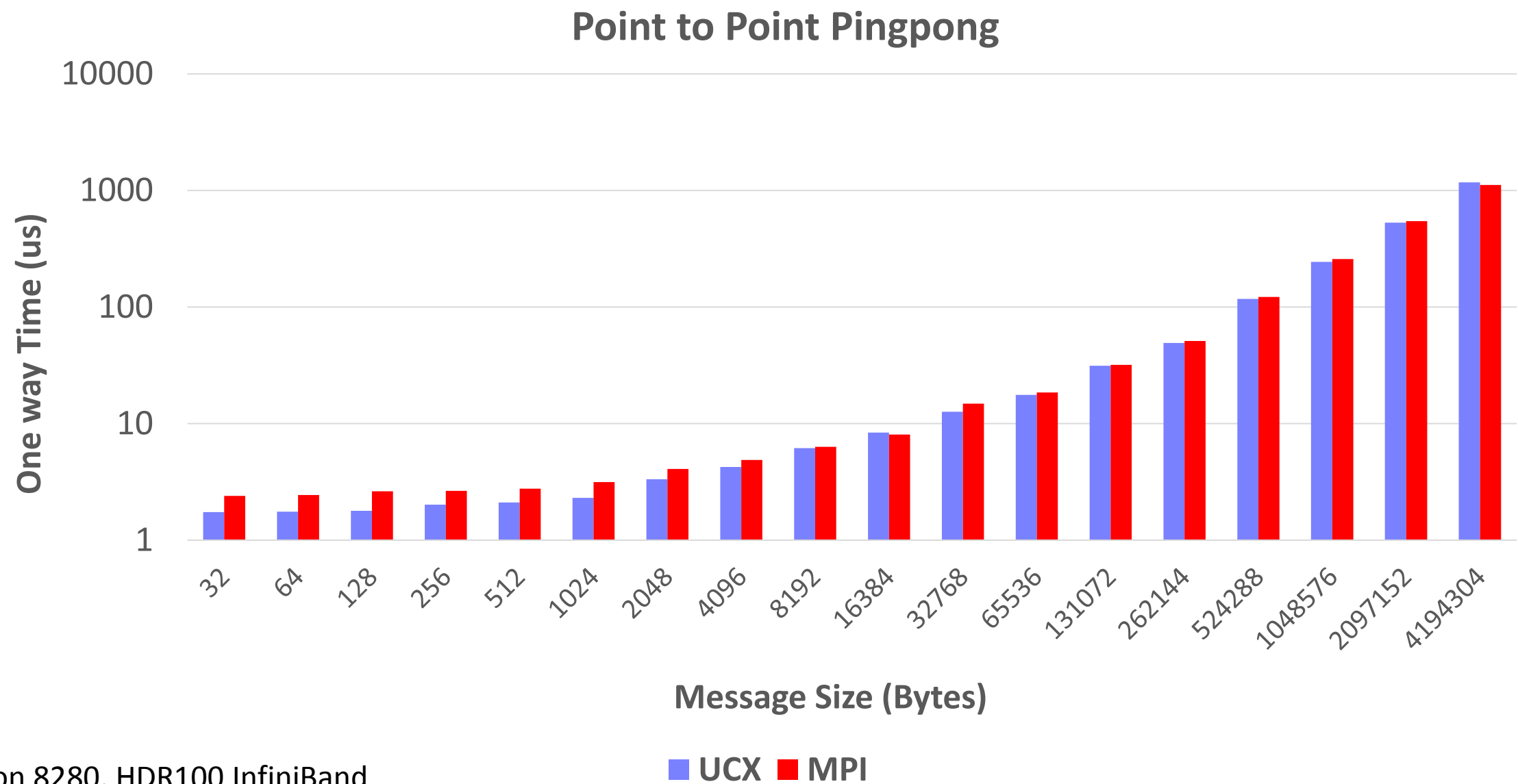
UCX Machine Layer Implementation

- Init
 - Process management: **simple pmi/slurm pmi/PMIx**
 - Each process :
 - **ucp_init**
 - **ucp_worker_create**
 - **ucp_ep_create**
 - Prepost recv buffers: **ucp_tag_recv_nb**
- Regular API
 - Send: **ucp_tag_send_nb**
 - Recv: **ucp_tag_recv_nb/ucp_tag_msg_recv_nb**
- Zero copy API
 - Send metadata message using Regular API
 - RDMA operations using **ucp_put_nb/ucp_get_nb**

Micro Benchmarks

Charm++ p2p Pingpong Benchmark - Frontera (TACC)

■ Up to 47% better than MPI

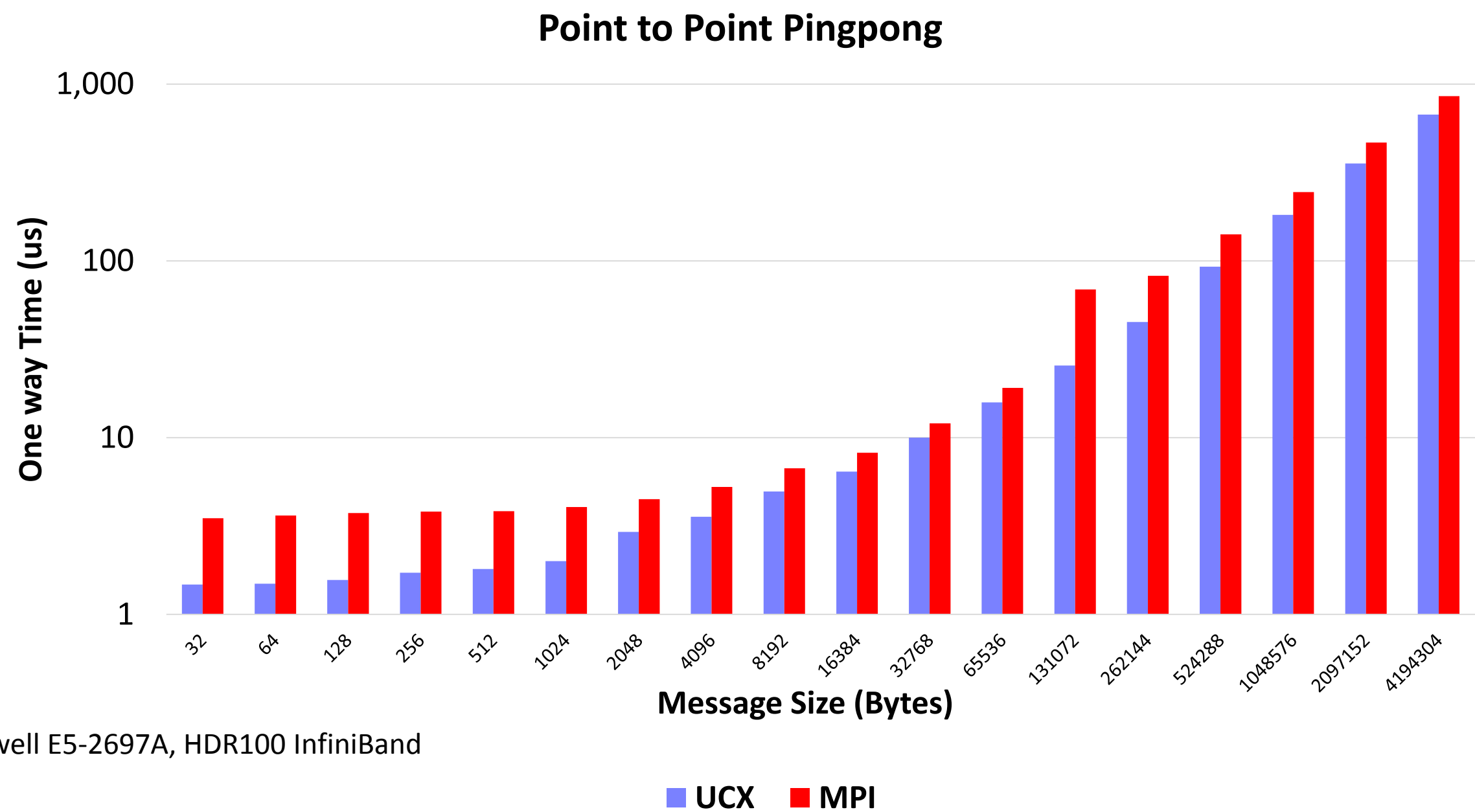


Intel Xeon 8280, HDR100 InfiniBand



Charm++ p2p Pingpong Benchmark - Thor (HPC Advisory Council)

■ Up to 63% better than MPI

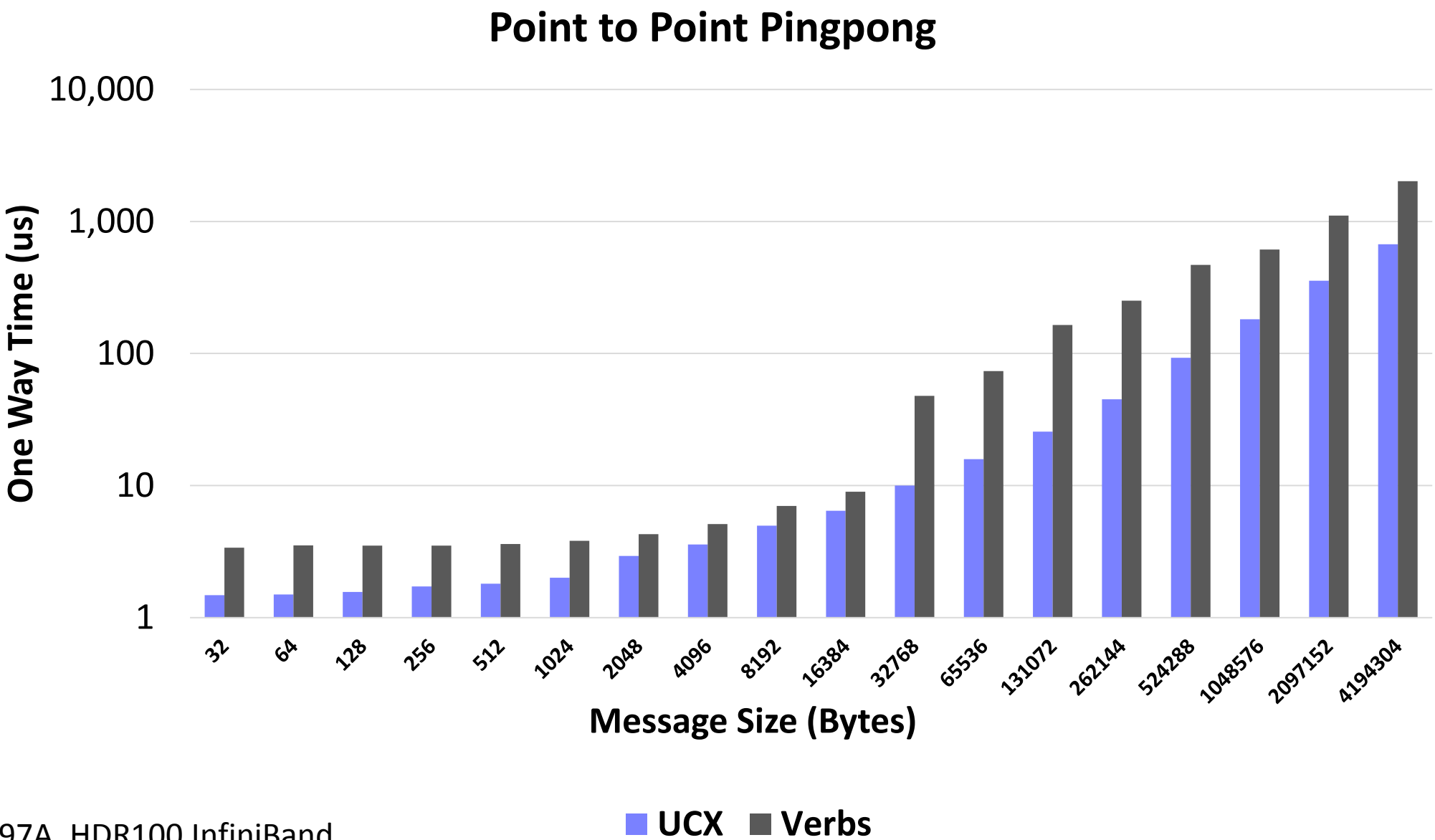


Intel Broadwell E5-2697A, HDR100 InfiniBand



Charm++ p2p Pingpong Benchmark - Thor (HPC Advisory Council)

■ Up to 87% better than Verbs



Intel Broadwell E5-2697A, HDR100 InfiniBand



Application Performance

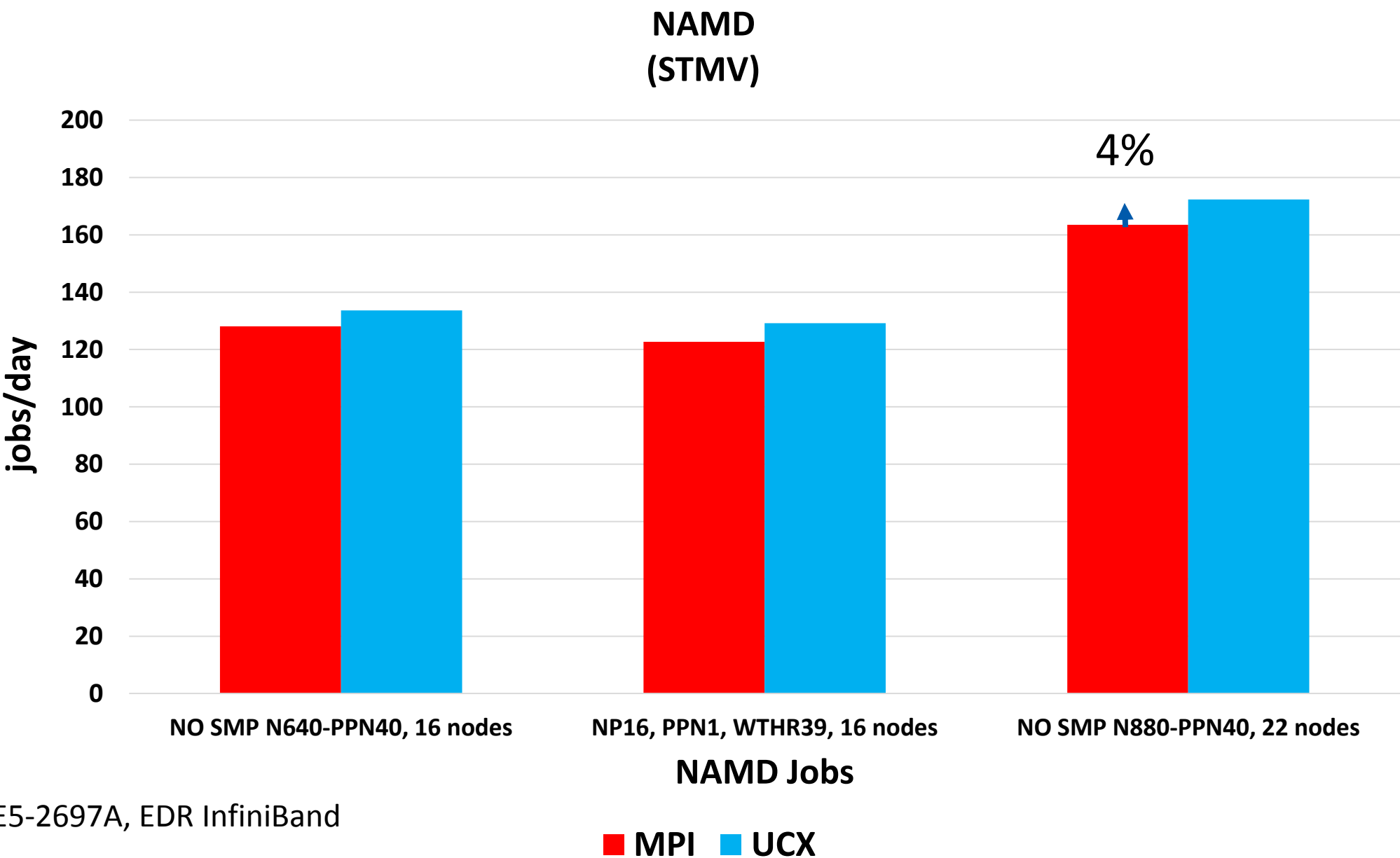
NAMD

- Nanoscale Molecular Dynamics (NAMD), is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems
- NAMD scales to hundreds of cores for typical simulations and beyond 500,000 cores for the largest simulations
- NAMD is written using Charm++ parallel programming model
- It is noted for its parallel efficiency and is often used to simulate large systems (millions of atoms)



NAMD (STMV) - Thor

- UCX Machine Layer is 4% faster than MPI Machine Layer for STMV (1M)

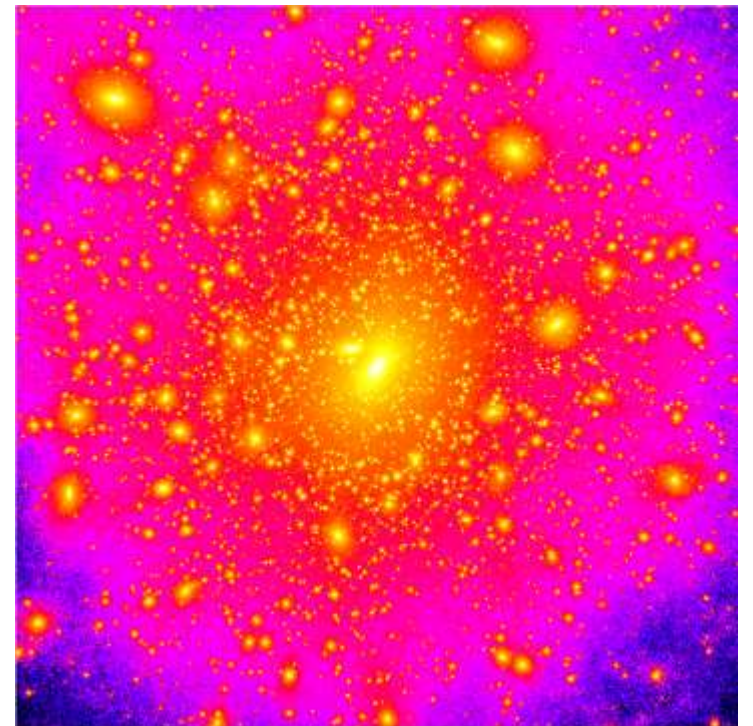


Intel Broadwell E5-2697A, EDR InfiniBand



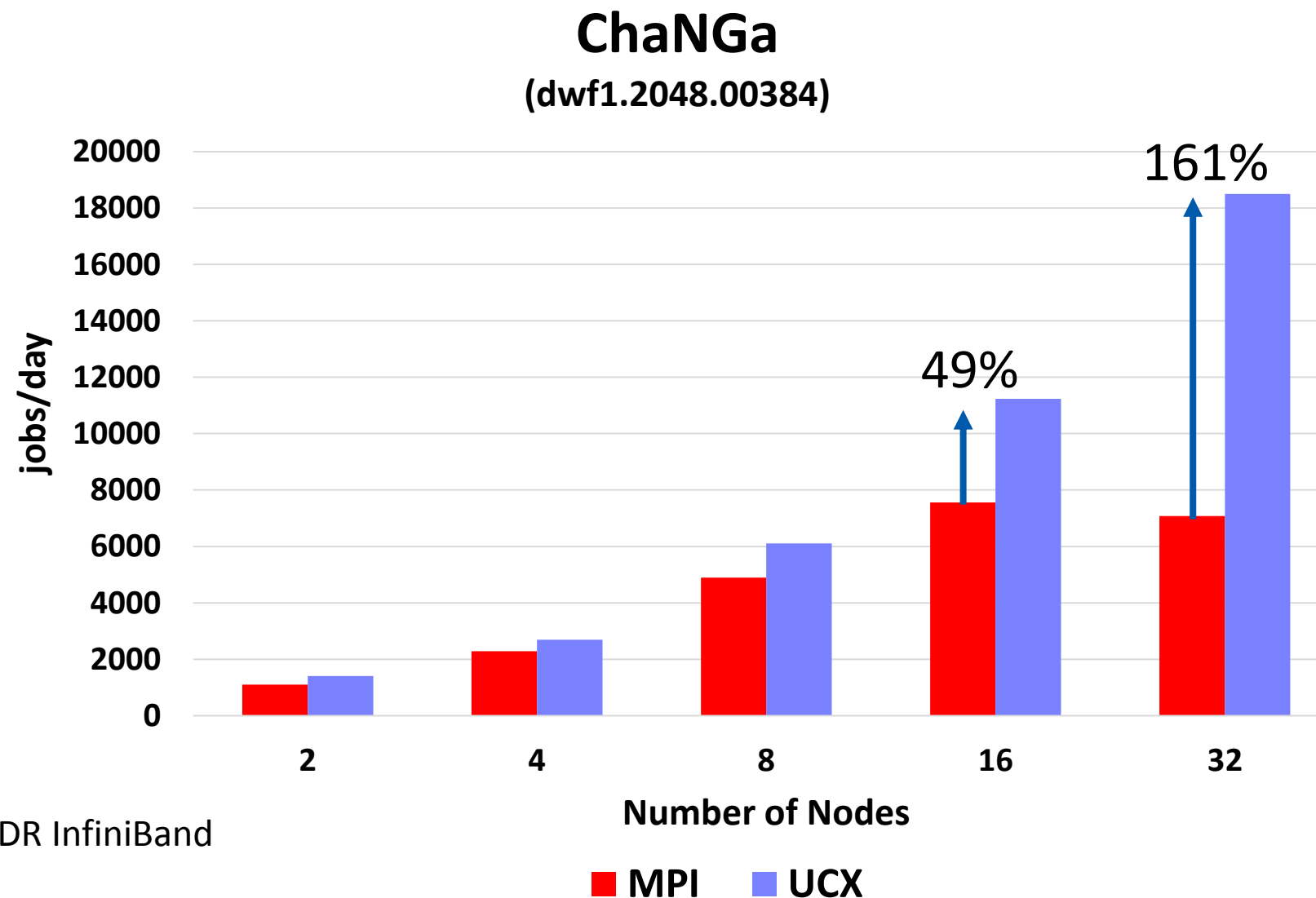
ChaNGa

- Cosmological simulation framework "ChaNGa" is a collaborative project with Prof. Thomas Quinn (University of Washington) supported by the NSF
- ChaNGa (Charm N-body GrAvity solver) is a code to perform collisionless N-body simulations
- ChaNGa can perform cosmological simulations with periodic boundary conditions in comoving coordinates or simulations of isolated stellar systems
- ChaNGa's uses dynamic load balancing scheme of the Charm++ runtime system to obtain good performance on parallel systems



ChaNGa (dwf 5M) – Thor

- UCX Machine provides 49% higher performance at 16 nodes
- UCX Machine provides 161% higher performance at 32 nodes
- Performance reduction demonstrated with MPI Machine Layer beyond of 16 nodes

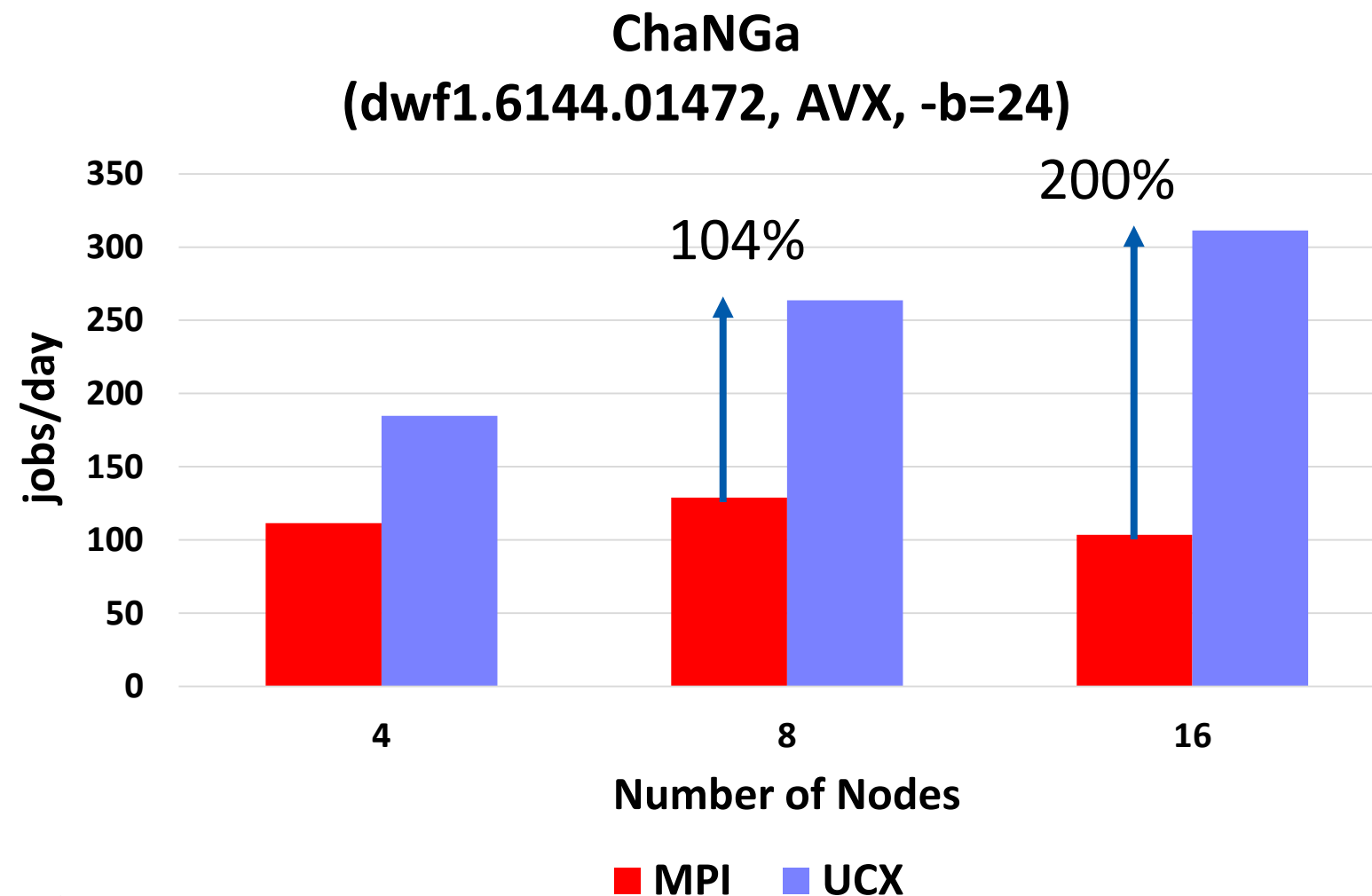


Intel Broadwell E5-2697A, EDR InfiniBand



ChaNGa (dwf 50M) - Thor

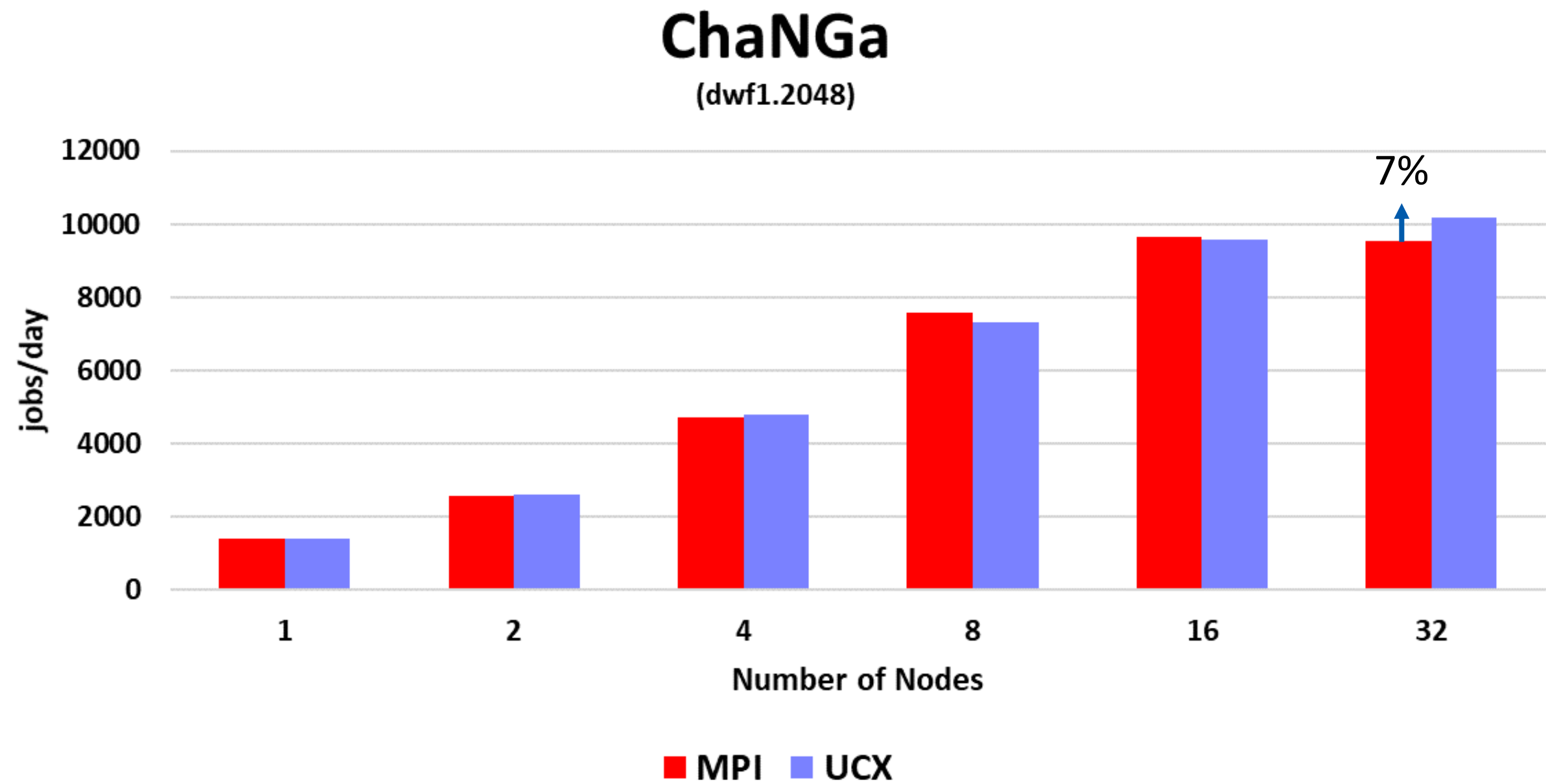
- UCX Machine provides 104% higher performance at 8 nodes
- UCX Machine provides 200% higher performance at 16 nodes
- Performance reduction demonstrated with MPI Machine Layer beyond of 16 nodes



Intel Broadwell E5-2697A, EDR InfiniBand

ChaNGa (dwf 5M) - Frontera (TACC)

■ Initial results

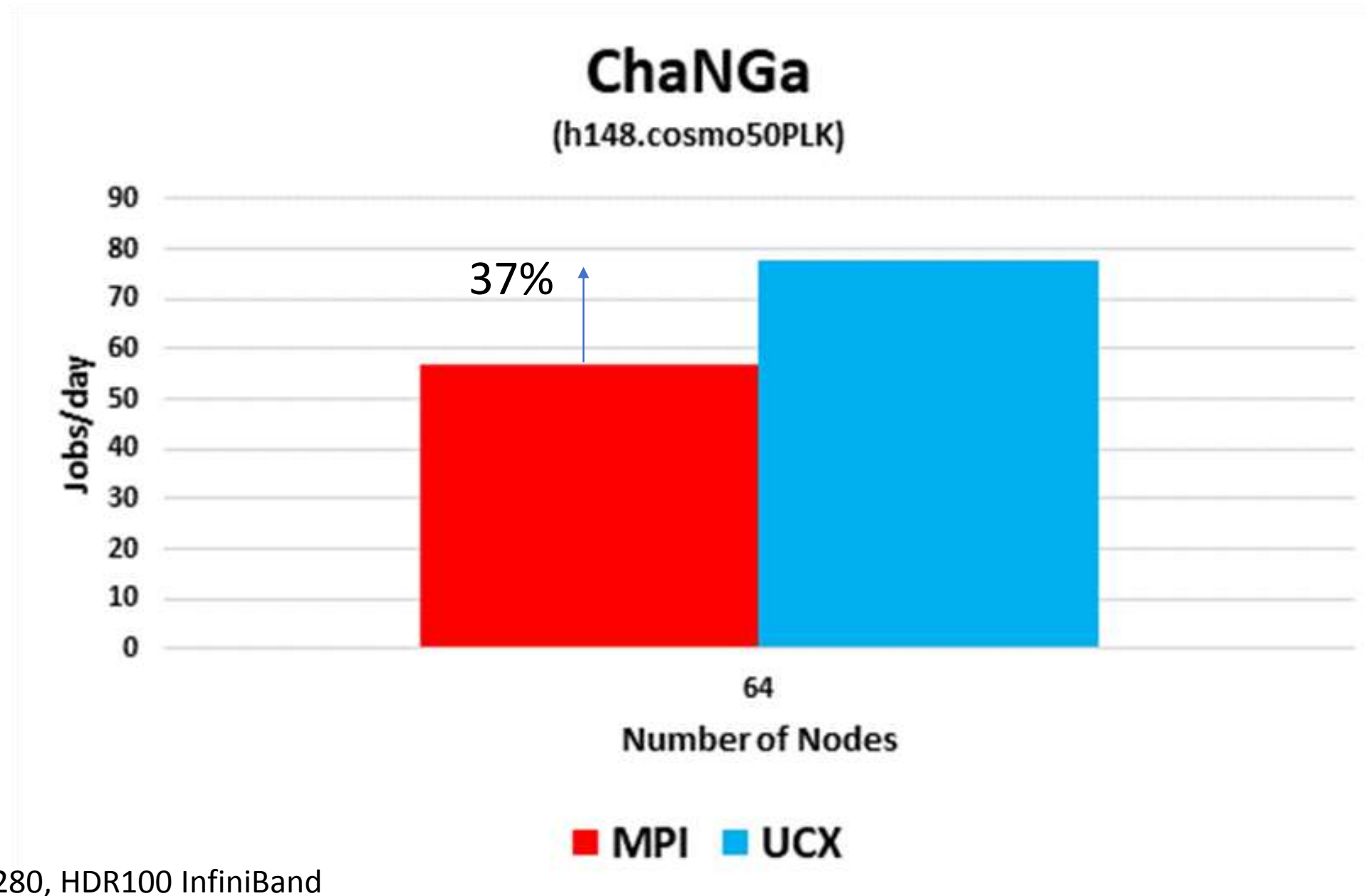


Intel Xeon 8280, HDR100 InfiniBand



ChaNGa (h148 - 550M) - Frontera (TACC)

- Milky way with a supermassive Black hole in the middle



Intel Xeon 8280, HDR100 InfiniBand

Conclusions and Future work

- UCX layer has been a performant layer as shown by initial testing and results.
- Future work
 - Testing on machines with other vendor networks (uGNI, PAMI etc.)
 - Performance analysis and tuning



Acknowledgements

- Mellanox
 - Ophir Maor
 - Yong Qin
 - Mikhail Brinskii
 - Yossi Itigin
- Charmworks, Inc
 - Evan Ramos
 - Eric Bohm
 - Sam White

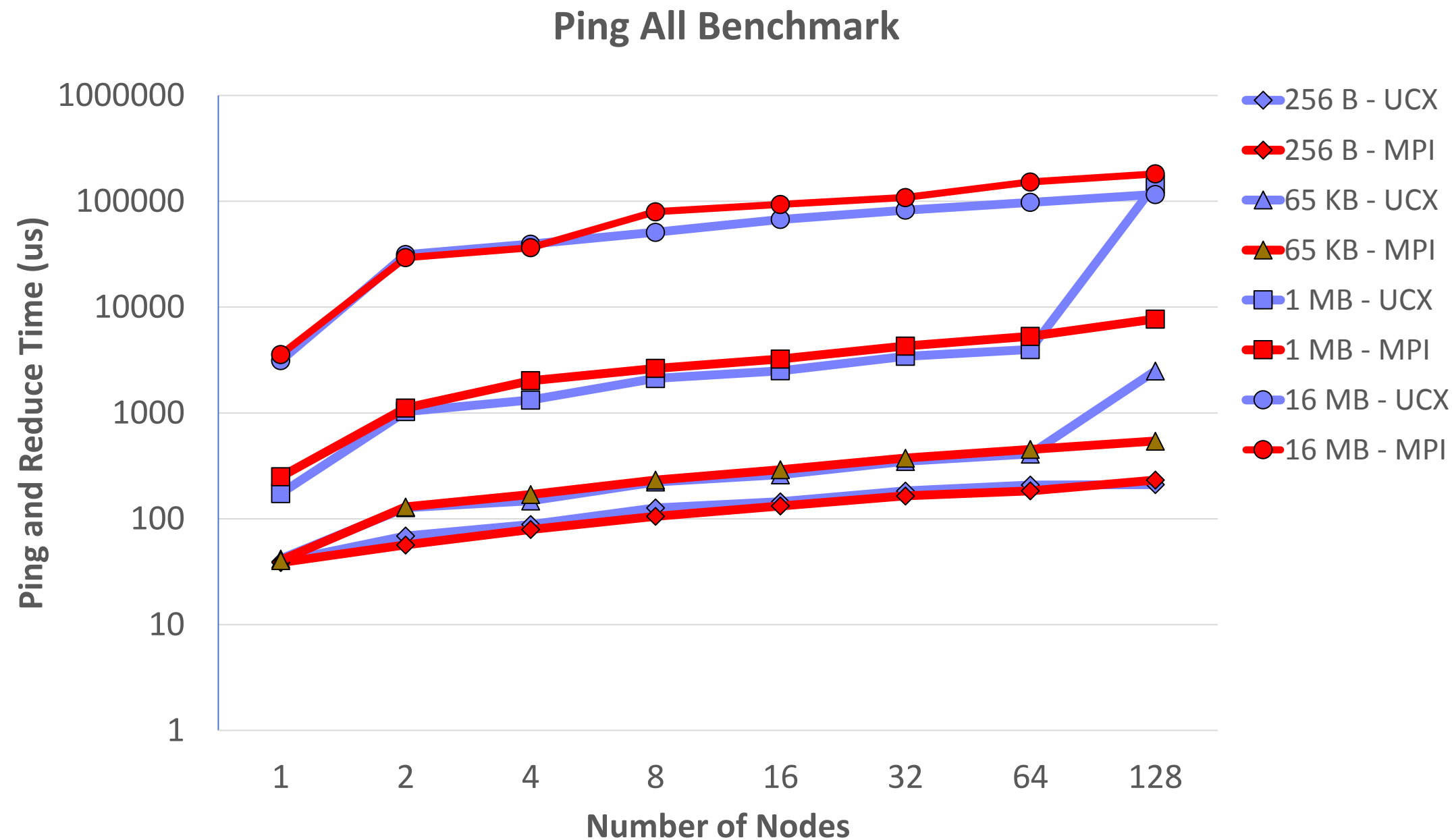


Thank You

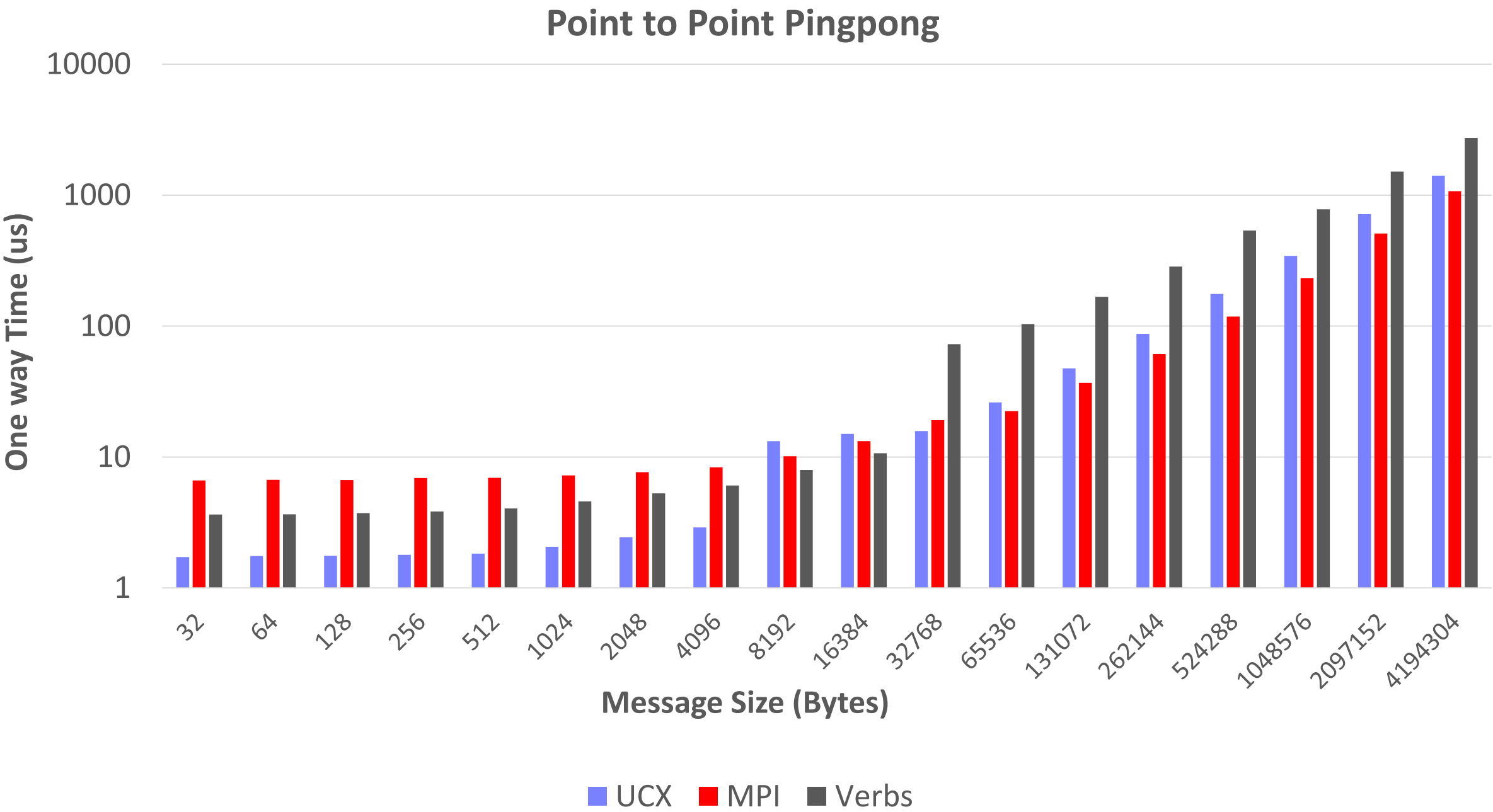


Extra Slides

Charm++ bcast Ping All Benchmark – Frontera (TACC)

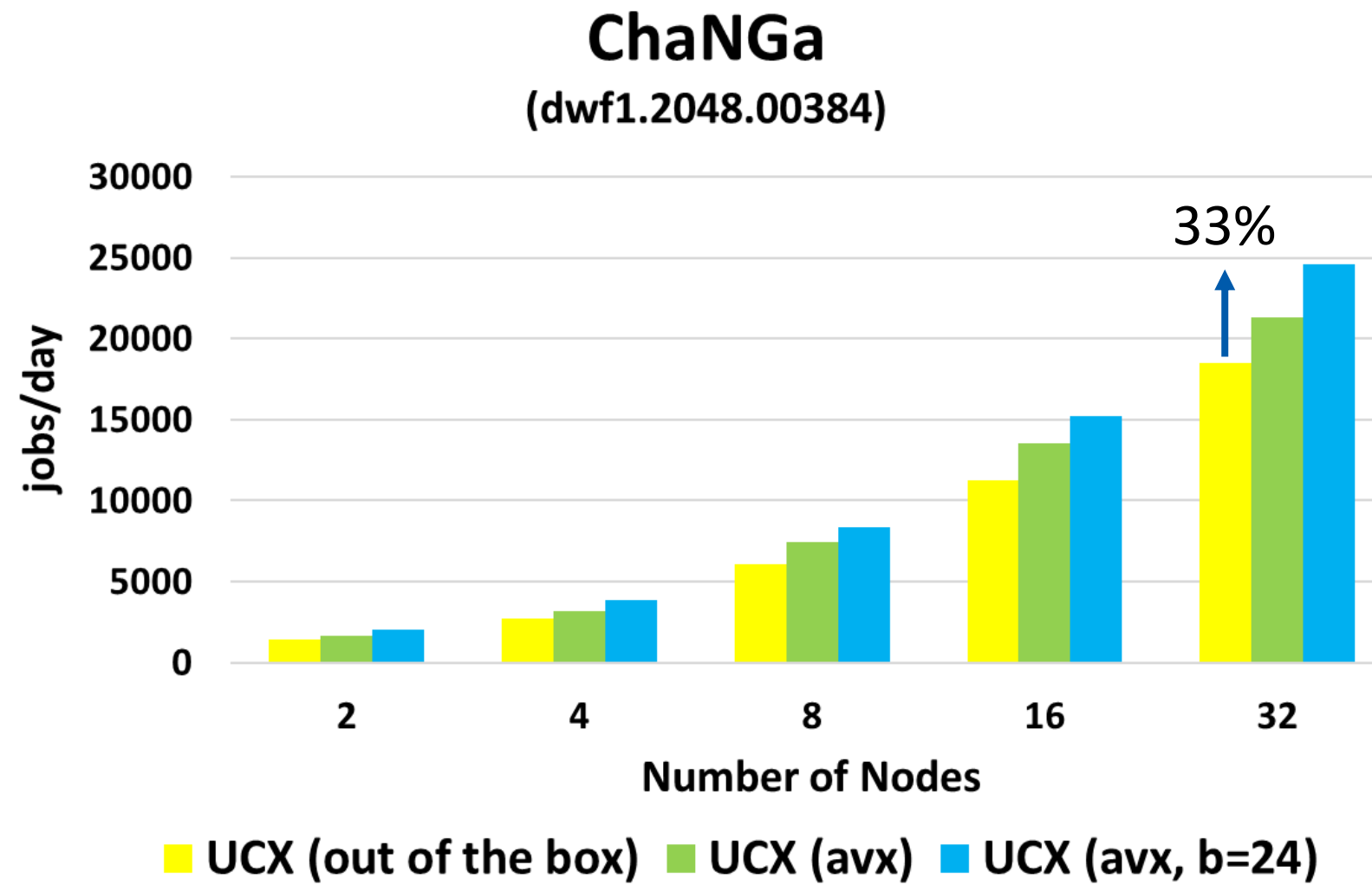


Charm++ p2p Pingpong Benchmark - iForge (NCSA)



ChaNGa (dwf 5M) UCX Performance Optimizations

- UCX Machine provides 33% higher performance when optimized comparing to out of the box performance



Intel Broadwell E5-2697A, EDR InfiniBand



ROCM UCX INTEGRATION

UCX community BoF

Supercomputing 2019

AMD
RADEON INSTINCT



Leverages OpenUCX For Scale-up and Scale-out Distributed Programming Models

- Next generation open source HPC communication framework
- Built off the foundation of MXM, UCCS, PAMI
- Broad Industry support including:
 - IBM, ARM, LANL, Mellanox, NVIDIA, ORNL, SBU, UT, UH and AMD
- Rich platform for supporting MPI, OpenSHMEM, PGAS



ROCM SOFTWARE PLATFORM

An Open Source foundation for Hyper Scale and HPC-class GPU computing

Graphics core next headless Linux® 64-bit driver

- Large memory single allocation
- Peer-to-Peer Multi-GPU
- Peer-to-Peer with RDMA
- Systems management API and tools



HSA drives rich capabilities into the ROCm hardware and software

- User mode queues
- Architected queuing language
- Flat memory addressing
- Atomic memory transactions
- Process concurrency & preemption



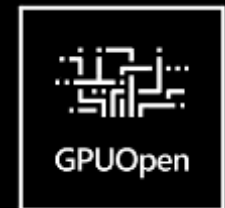
Rich compiler foundation for HPC developer

- LLVM native GCN ISA code generation
- Offline compilation support
- Standardized loader and code object format
- GCN ISA assembler and disassembler
- Full documentation to GCN ISA



“Open Source” tools and libraries

- Rich Set of “Open Source” math libraries
- Tuned “Deep Learning” frameworks
- Optimized parallel programming frameworks
- CodeXL profiler and GDB debugging



ANNOUNCING ROCm 3.0:

PRE-EXASCALE STACK FOR HPC & ML



OpenMP for GPUs

100% Open

Makes CUDA Portable

PyTorch, TensorFlow Up-streamed

Datacenter-Ready at Scale

AMD EXASCALE STACK

Applications	HPC Apps		ML Frameworks	
Cluster Deployment	Singularity	SLURM	Docker	Kubernetes
Tools	Debugger	Profiler, Tracer	System Valid.	System Mgmt.
Portability Frameworks	Kokkos	RAJA	GridTools	ONNX
Math Libraries	RNG, FFT	Sparse	BLAS, Eigen	MIOpen
Scale-out Comm. Libraries	OpenMPI	UCX	MPICH	RCCL
Programming Models	OpenMP	HIP	OpenCL™	Python
Processors	CPU + GPU			
	<div>Future</div> <div>Beta/Early</div> <div>Production</div>			

ROCm for Distributed Systems

CPU can directly access to GPU memory

- Expose entire GPU frame buffer as addressable memory through PCIe BAR (LargeBar feature)
- Map GPU pages to CPU pages that allow CPU to directly load/store from/to GPU memory

▲ HCA to directly access GPU memory : ROCnRDMA feature

- Leverages Mellanox's PeerDirect feature
- Allows IB HCA to directly read/write data from/to GPU memory
- Available and enabled by default in ROCm
- Integrated into ROCm drivers

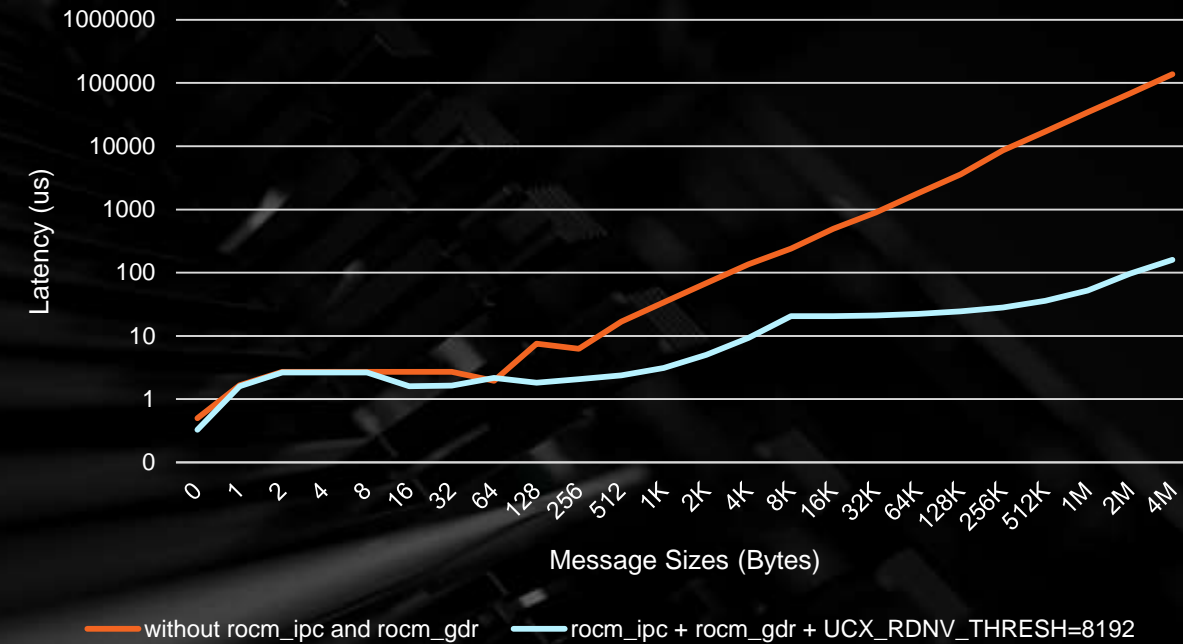
▲ IPC for intra-node communication

- ROCm-IPC in UCT for interprocess communications among GPUs
- Improvement from the original CMA TL

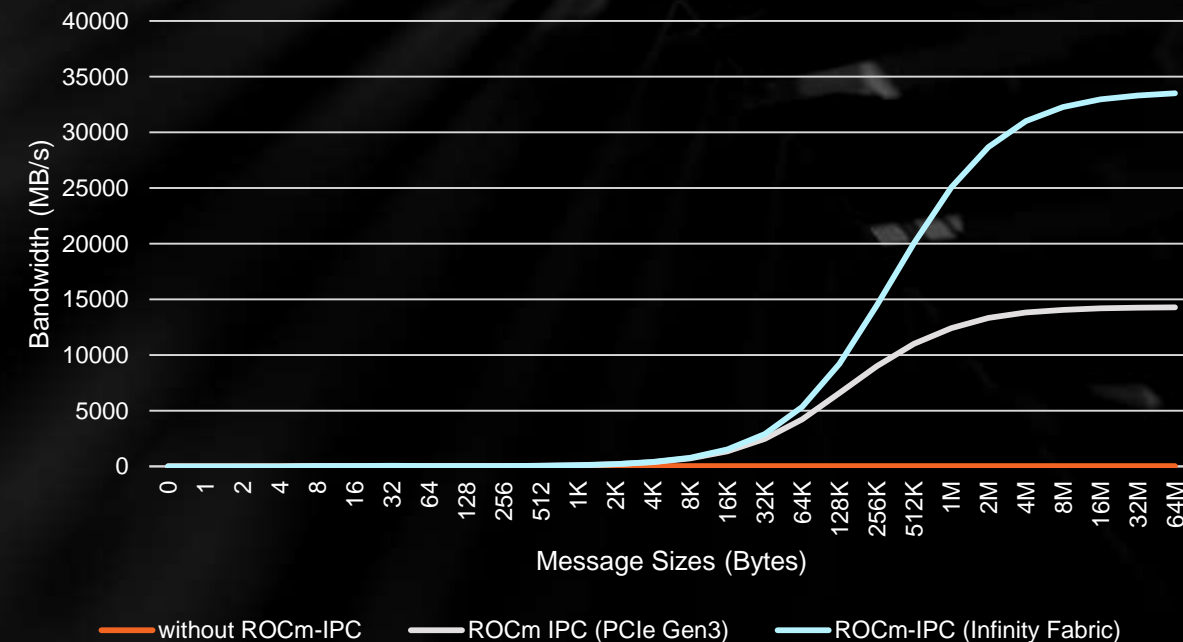
UCX OVER ROCM: INTRA-NODE SUPPORT

- Improvements for ROCm support
 - rocm_ipc: for intra-node cross process zcopy, support ROCm or host memory
 - rocm_cpy: for intra-process short and zcopy, support ROCm or host memory
 - rocm_gdr: use gdr_copy for fast read speed from GPU device memory
 - Enabled perftest and gtest for ROCm support
- ROCM-IPC provides efficient support for large messages
 - 1.61 us for 16 Bytes, 36 us for 512KBytes transfer for intra-node (D-D)
 - 33 GB/s for 32MBytes for intranode D-D transfer over Infinity Fabric
- Test Configuration:
 - AMD Radeon Instinct MI50 GPUs on PCIe Gen3 platform
 - Hip-ified OSU Micro Benchmarks

OSU Micro Benchmarks - Latency
(Intranode, Device-to-Device)



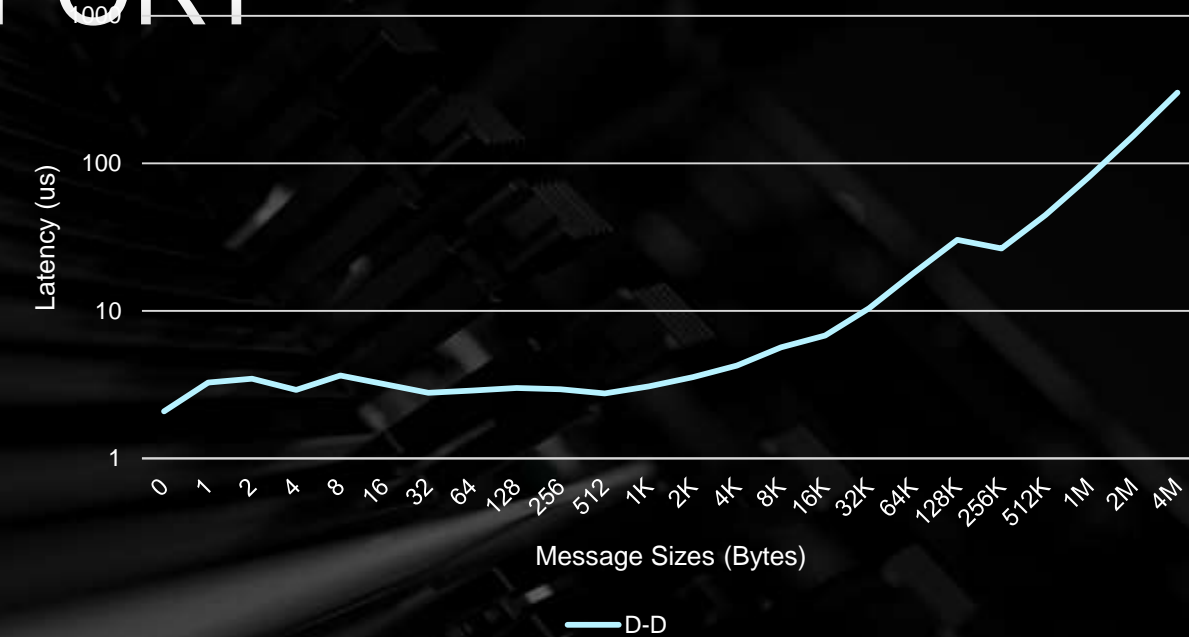
OSU Micro Benchmarks - Bandwidth
(Intranode, Device-to-Device)



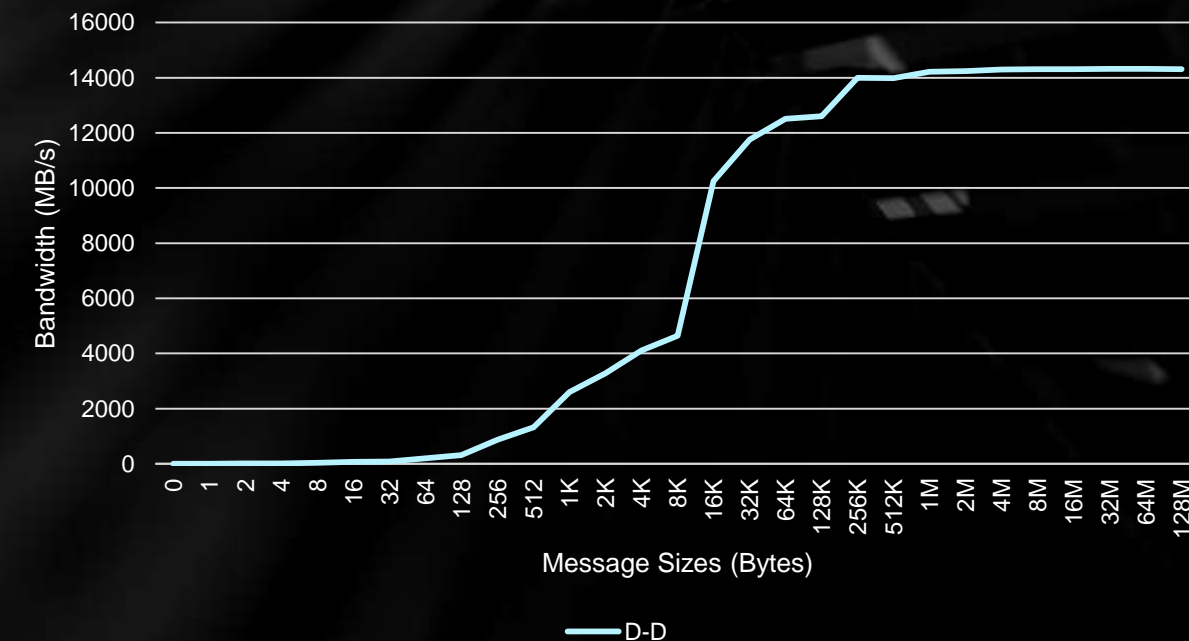
UCX OVER ROCM: INTER-NODE SUPPORT

- Takes advantage of LargeBar capability to support eager protocols
 - Eager protocols can run directly from GPU buffers
- Integrated ROCnRDMA to design rendezvous (RNDV) protocols
- Optimization and tuning work to be continued
 - Enhanced and optimized GPU-Aware protocols for pipeline
- Performance shown UCX over ROCm
 - 2.9 us for 4 Bytes transfer for inter-nodes
 - Full EDR IB bandwidth achieved on large messages
- Expect EPYC Rome Gen4 platform to deliver full HDR IB bandwidth with MI50
 - HDR IB, GPU device and EPYC Rome platform are PCIe Gen4 capable
- Test Configuration:
 - AMD Radeon Instinct MI50 GPUs on x86 PCIe Gen3 platform
 - Mellanox ConnectX-5 EDR InfiniBand
 - HIP-ified OSU Micro Benchmarks

OSU Micro Benchmarks - Latency
(Inter-node, Device-to-Device)



OSU Micro Benchmarks - Bandwidth
(Inter-node, Device-to-Device)



DISCLAIMERS AND ATTRIBUTIONS

The information contained herein is for informational purposes only, and is subject to change without notice. Timelines, roadmaps, and/or product release dates shown in these slides are plans only and subject to change. “Polaris”, “Vega”, “Radeon Vega”, “Navi”, “Zen” and “Naples” are codenames for AMD architectures, and are not product names.

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.

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

©2019 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo, Ryzen, Threadripper, EPYC, 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.



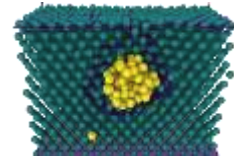
- UCX has been deployed at ORNL:
 - Summit: Power9+NVIDIA
 - DGX-2 systems: x84+NVIDIA
 - Visualization Clusters (Rhea)
 - Wombat: ARM+NVIDIA
- Performance Portable "communication API" between "diverse" set of architectures
- Helping co-designing next generation PM: RDMA/OpenSHMEM, Python/Dask for HPC, Accelerator-based
- Help us to evaluate new systems "very fast" and efficiently.

NVIDIA+ARM evaluation on “Wombat” (NCCS) used UCX

Applications:



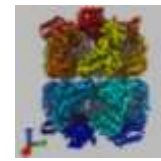
CoMet
Comparative
Genomics



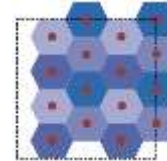
LAMMPS
Molecular
Dynamics



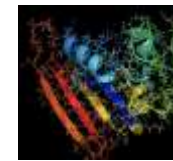
NAMD
Molecular
Dynamics



VMD
MD
Visualize.



DCA++
Material
Science



Gromacs
Molecular
Dynamics



Gamera
Earthquake
Simulator



LSMS
Material
Science

Benchmarks & Mini-apps:



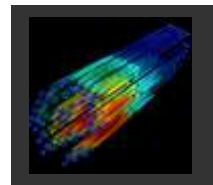
BabelStream
Memory
Transfer



**Tea
Leaf**
Heat
Cond.



**Clover
Leaf**
Lagrangian
Eulerian
hydrodynamic



MiniSweep
Radiation
Transport

SNAP
Radiation
Transport



**Patatrack
Pixel
Reconstruction**

Parallel Prog Models & Sci. Libraries:



Kokkos
C++
Prog.
Model



Open MPI
Distributed
Prog.
Model



**CUDA /
CUDA
Fortran**
GPU Prog.



UCX
Comm.
Framework



**Magma
SLATE**
Sci.
Libraries

Hardware:

HPE Apollo 70 Preproduction nodes

CPU: **ARM ThunderX2**

2 Sockets, 28 Cores/socket, 4 threads/per core, 2.0GHz, 256 GB RAM

GPU's: **NVIDIA Volta GV100** (2 per node) with 32 GB HBM2 each

4 nodes with NVIDIA GPUs

EDR InfiniBand

Software:

RHEL 8

CUDA 10.2.107 (aka “Drop 2”), PGI “Dev version”

Installed Nov 7. Most user's results are with **10.2.91** (“Drop 1”)

GCC 8.2.1 is the default compiler and **armclang 19.3** available

Open MPI 3.1.4 and **4.0.2rc3**

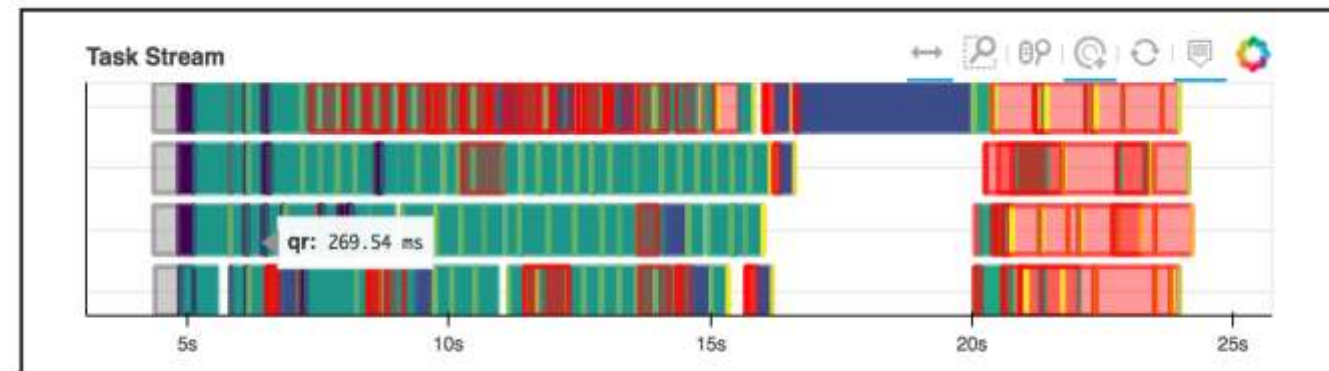
UCX 1.7.0

Evaluation: https://www.olcf.ornl.gov/wp-content/uploads/2019/11/ARM_NVIDIA_APP_EVALUATION-1.pdf

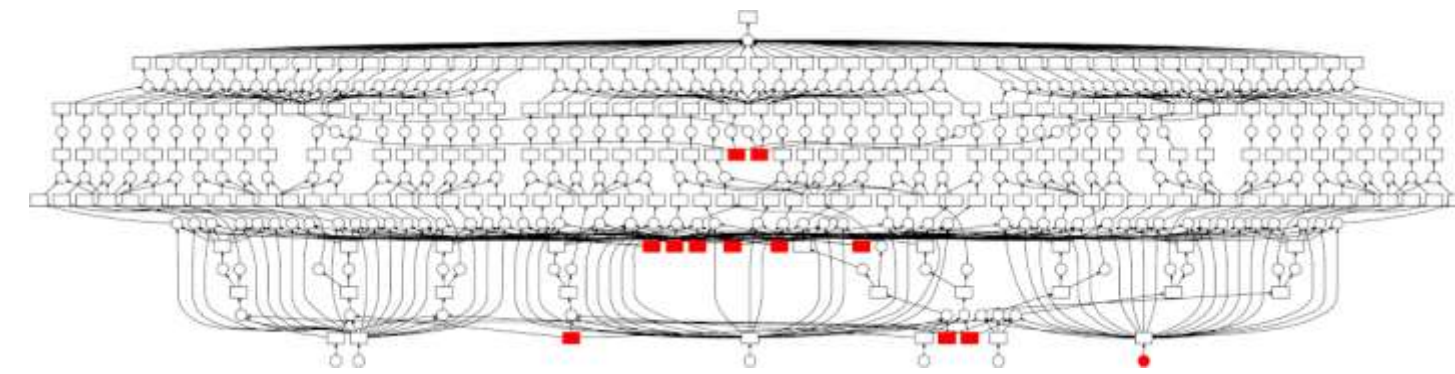
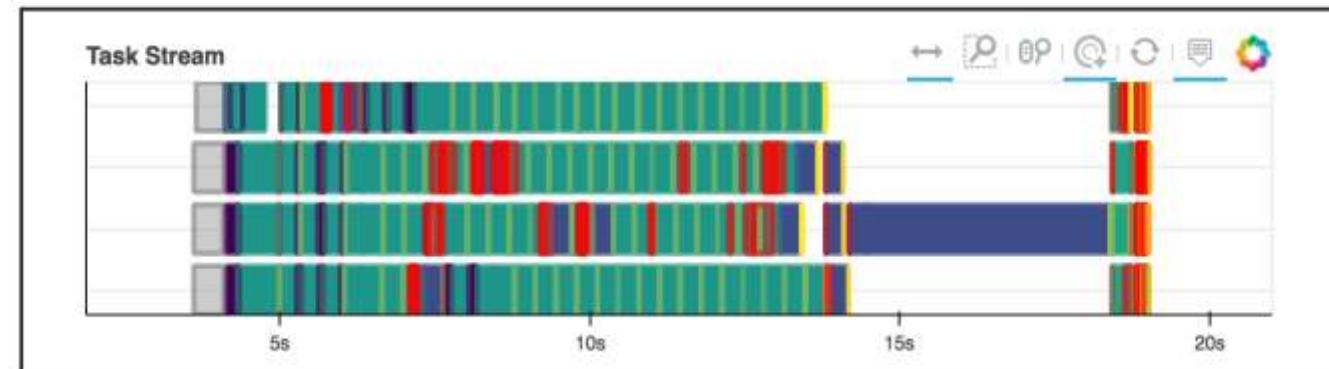
DASK + UCX on DGX-1

Helping converge HPC to Data Sciences, AI, and BigData

Before UCX:



After UCX:



- Science mission loves Python
 - Easy to learn, free, numpy is similar to MATLAB
- RAPIDS
 - Open source data science python libraries
 - Single thread multi accelerator
- DASK
 - Distributed tasking framework for python
- UCX
 - Client-server model, python bindings, efficient RDMA, etc

Source: *Matthew Rocklin, Rick Zamora*, **Experiments in High Performance Networking with UCX and DGX**
<https://blog.dask.org/2019/06/09/ucx-dgx>



NVIDIA UCX UPDATE

Akshay Venkatesh, Sreeram Potluri, CJ Newburn

ENABLING HPC AND DATA SCIENCES

Providing a Common CUDA-aware Runtime

- MPI
 - OpenMPI
 - MPICH
 - Parastation MPI
- Dask
 - RAPIDS / CuML

MPICH



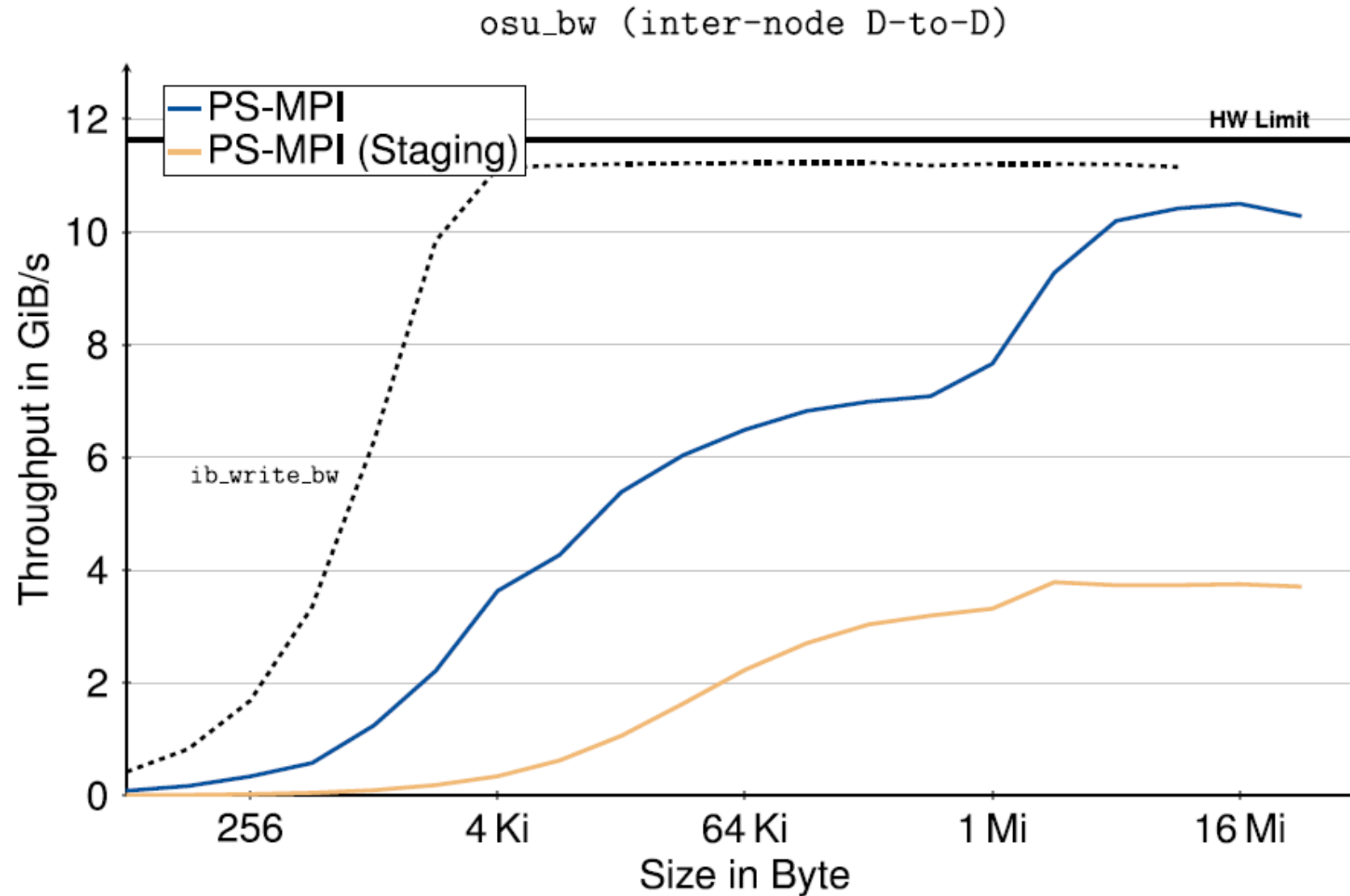
OPEN MPI

ParaStation
MPI

RAPIDS



Preliminary Results

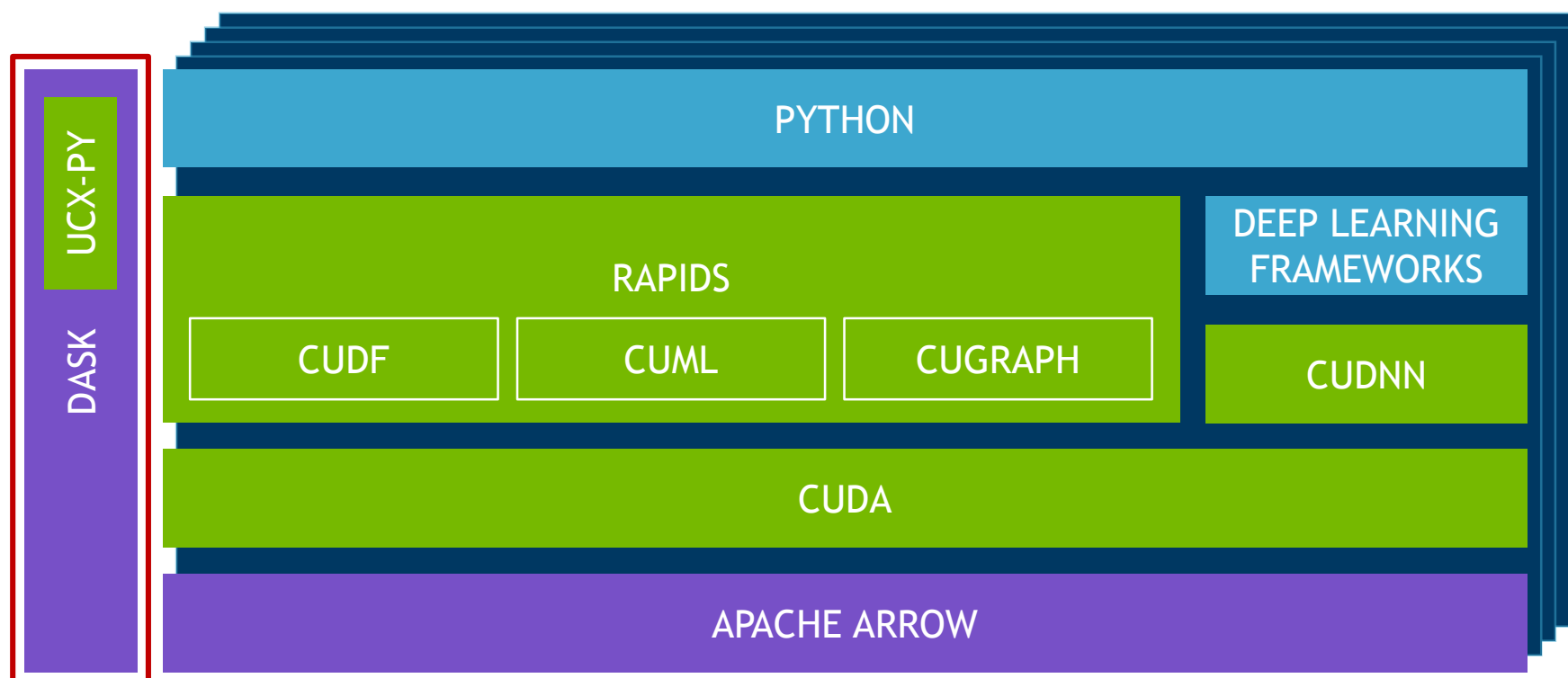


Two JUWELS nodes @ JSC

- Dual-socket Intel Xeon Gold 6148
- Dual EDR-InfiniBand (ConnectX-4)
- 4x Nvidia V100 GPU
- OSU Bandwidth (5.6.2)
- Compiled using GCC/8.3.0
- UCX v1.7.0rc1

DASK AND RAPIDS

- RAPIDS uses dask-distributed for data distribution over python sockets
- Communication of python objects backed by GPU buffers needed
- Critical to leverage IB, NVLINK, Sockets

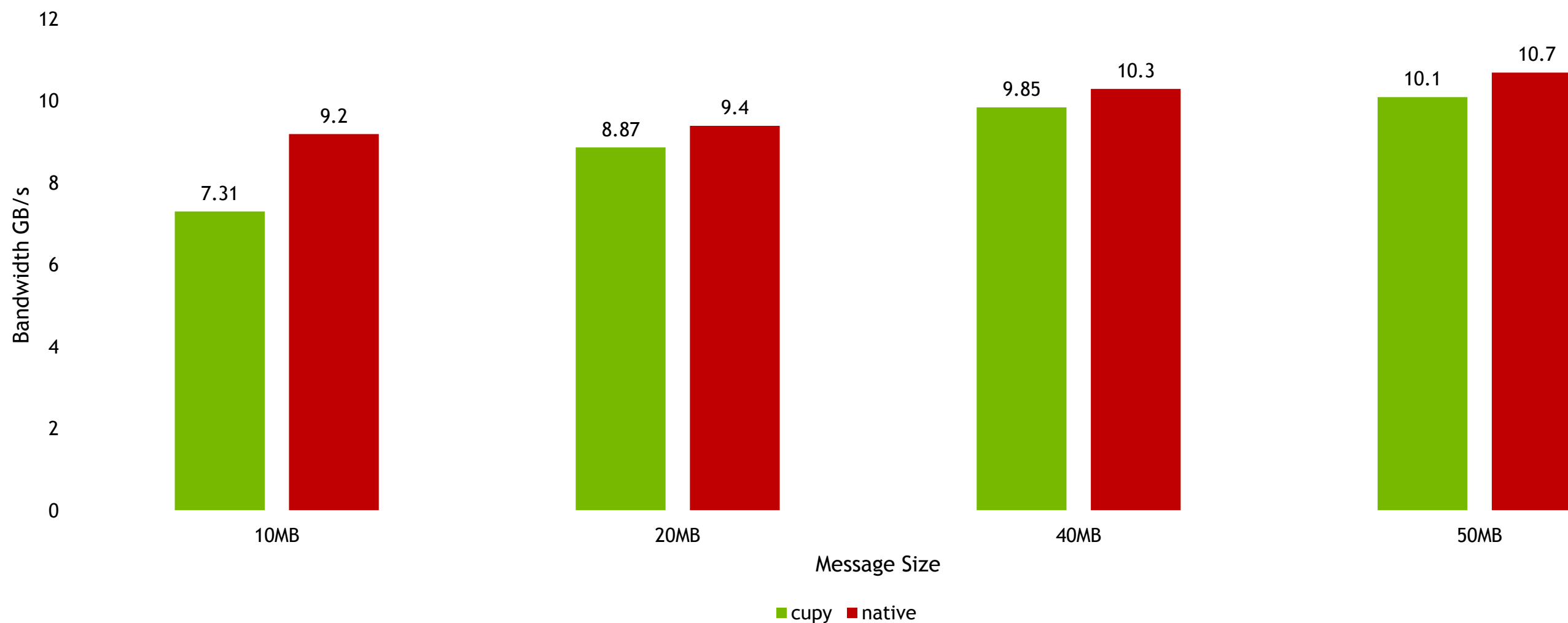


FEATURES IN UCX-PY AND UCX

- Python interface
 - Coroutine support
 - CUDA-array interface to move device memory-backed objects
- Interoperability with coroutines
 - blocking progress with CUDA transport
- Client-server API
 - support with sockets and IB

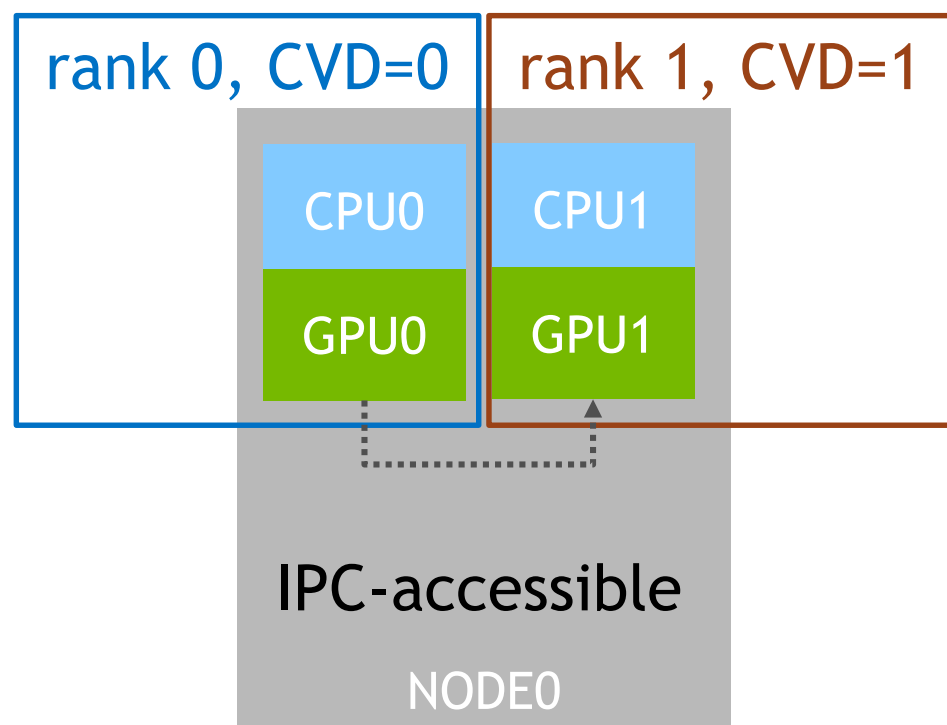
DEVICE MEMORY BANDWIDTH

Cupy bandwidth between 2 Summit nodes



SUPPORT FOR CUDA_VISIBLE_DEVICES

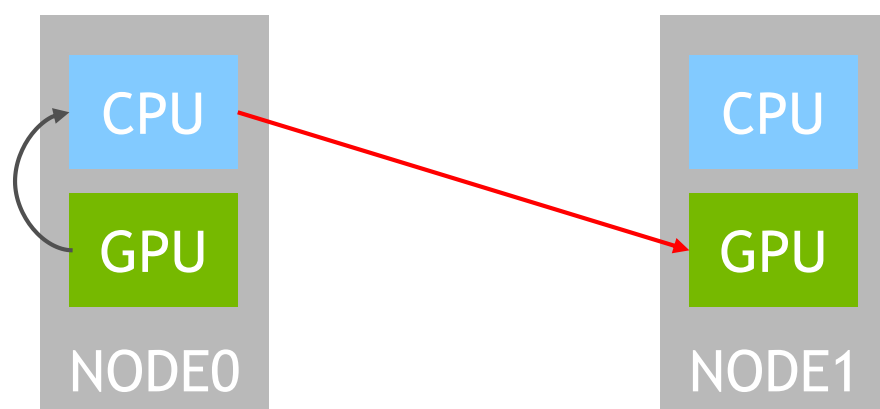
Leveraging P2P and NVLink



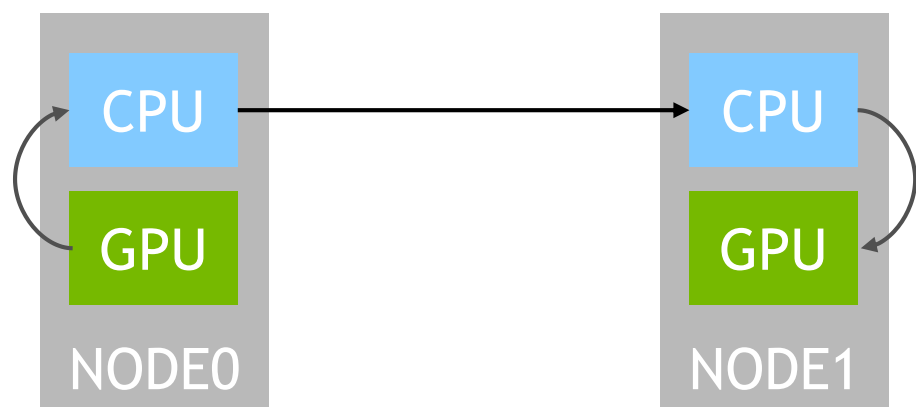
- Job managers like SLURM to carve out GPUs within a node for ranks using `CUDA_VISIBLE_DEVICES`
- Also used by task schedulers like DASK
- CUDA 10.1 enabled CUDA-IPC between devices in different visibility domains
- UCX now leverages this feature

PERFORMANCE ACROSS PLATFORMS

3-state Pipelining



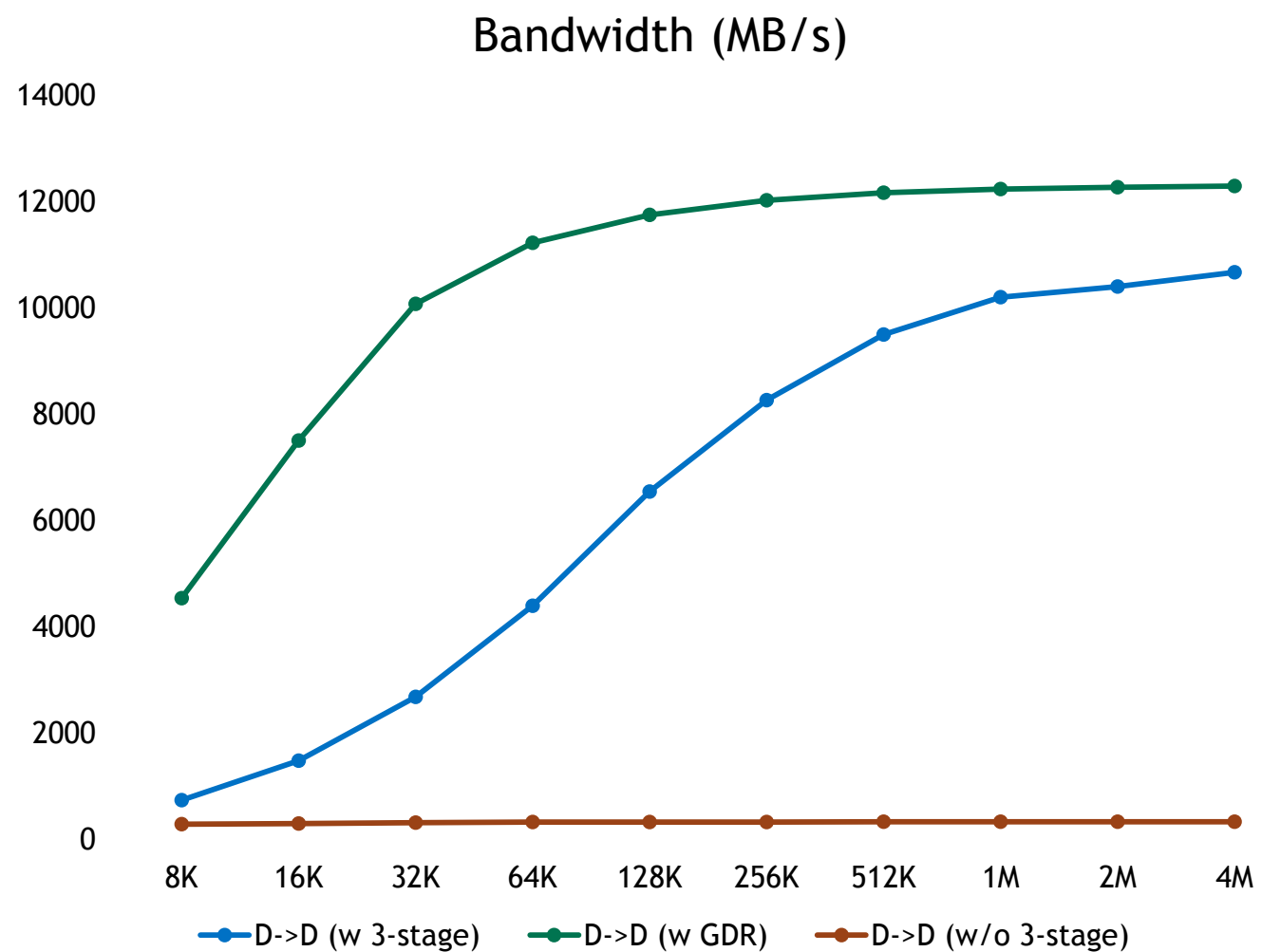
Current Rendezvous pipeline needs GDR



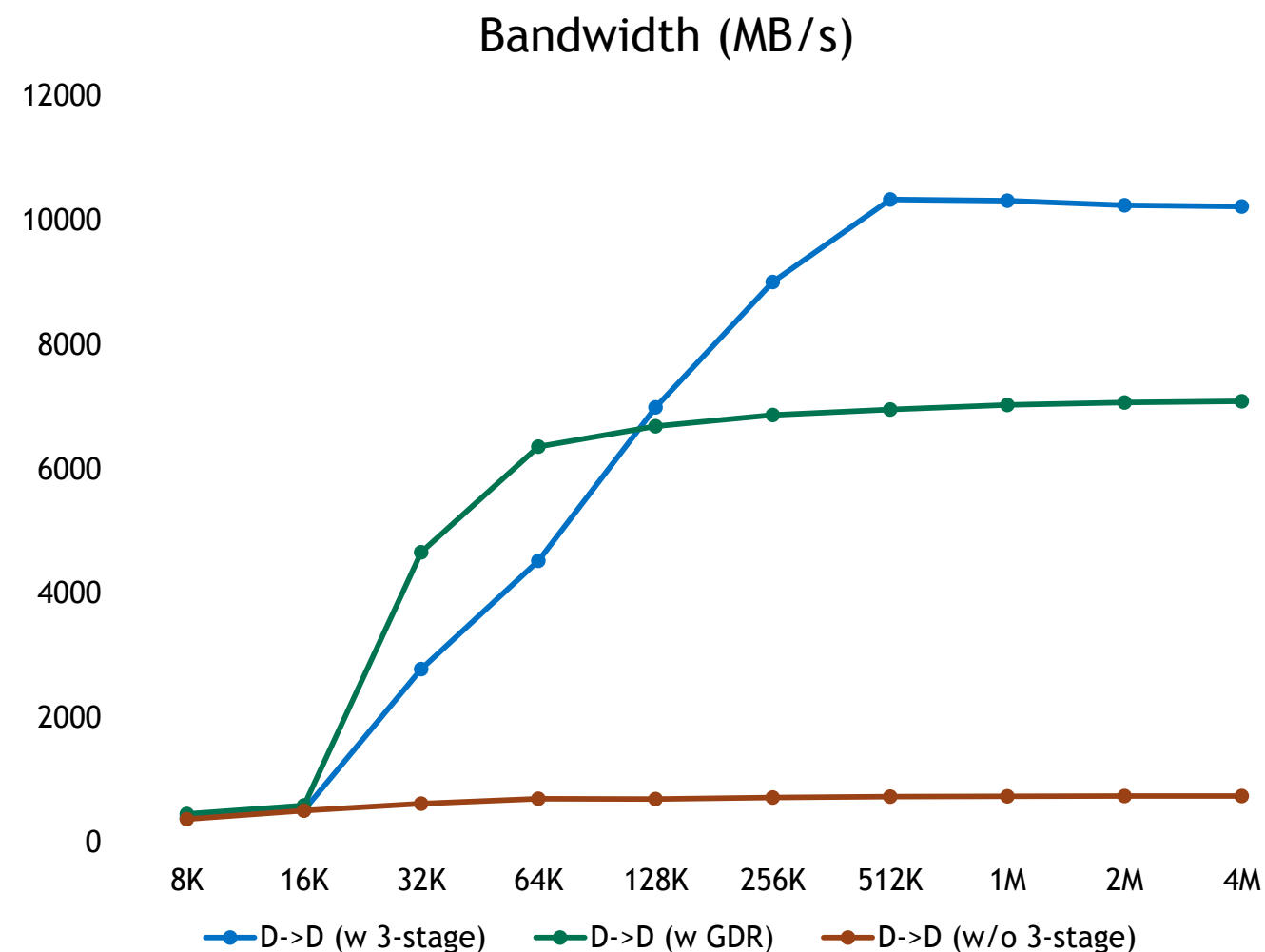
3-stage pipeline doesn't need GDR

- Increasing use on high-end as well as low end servers
- GPUDirect RDMA is not performant on all platforms
 - Limited PCIe P2P performance on most CPUs
- Efficient staging to host is required
 - Need for broader platform support
 - Addressed in UCX master (Mellanox contribution)

3-STAGE PIPELINE PERFORMANCE



GPU and NIC connected by PCIe Switch



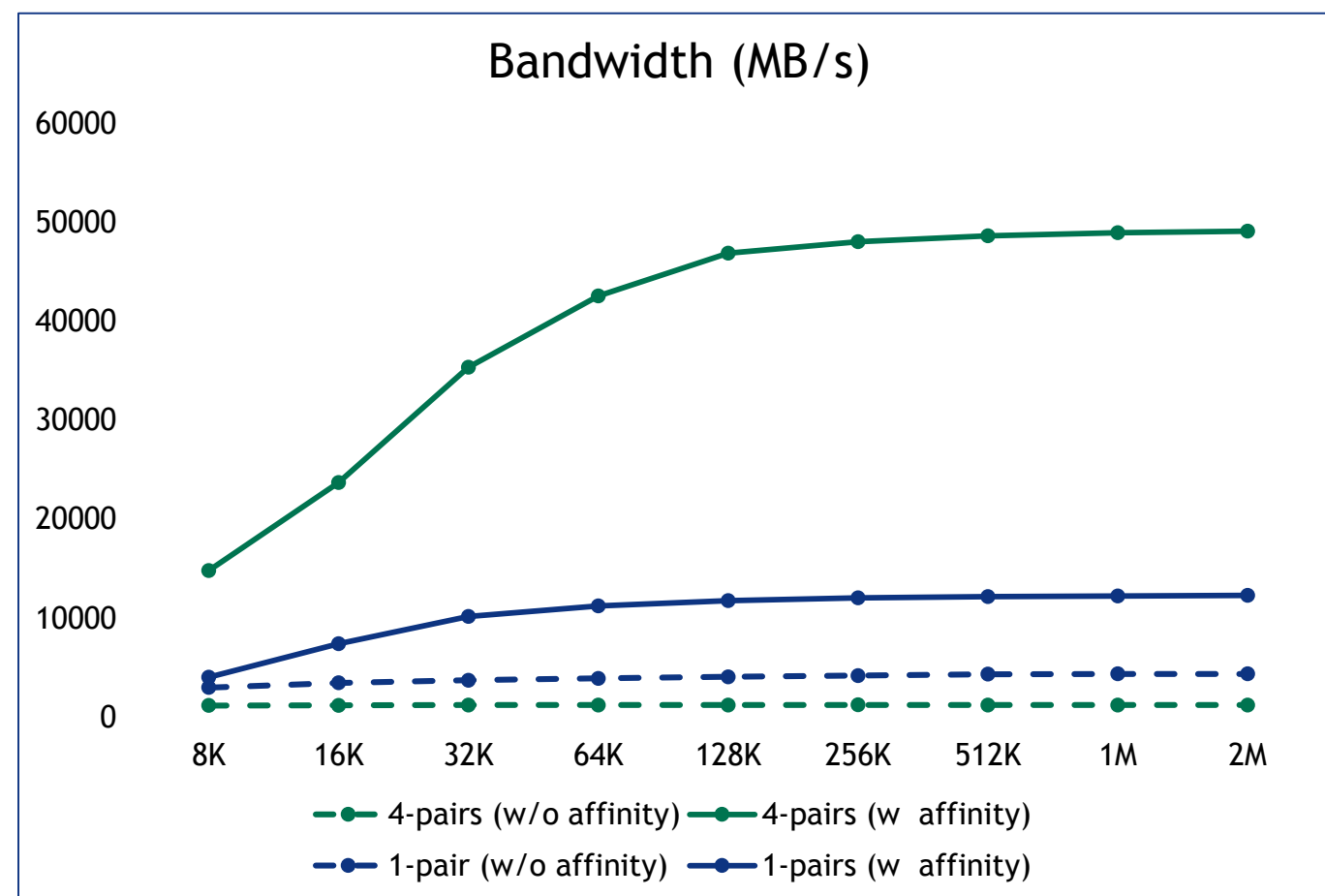
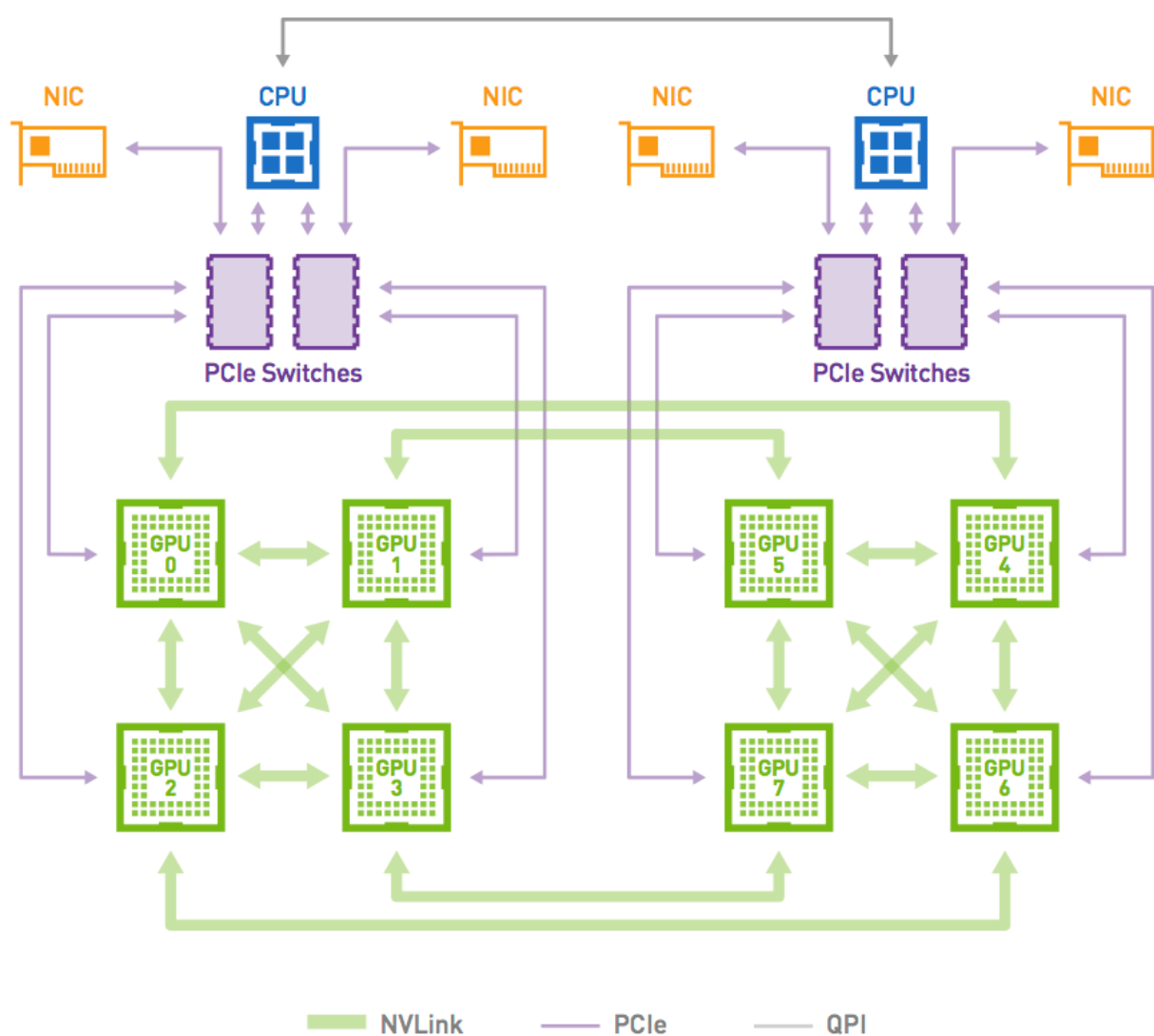
GPU and NIC connected to CPU

FUTURE FEATURES

- Topology aware NIC and threshold selection
- Extend 3-stage pipeline for intra-node cases and managed memory
- Optimizations for cloud deployments
- GPU-support with UCX Java-bindings

EFFECT OF GPU-HCA AFFINITY

DGX-1, V100, CX-5: Inter-node osu_mbw_mr



FUTURE DIRECTIONS

- Open source reference implementation of CUDA-aware runtime
- Enabling HPC and Data Science libraries/platforms
- Optimize across architectures in bare metal and cloud



THANK YOU



MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

Enhancing MPI Communication using Hardware Tag Matching: The MVAPICH Approach

Talk at UCX BoF (SC '19)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

E-mail: panda@cse.ohio-state.edu

<http://www.cse.ohio-state.edu/~panda>

Introduction, Motivation, and Challenge

- HPC applications require high-performance, low overhead data paths that provide
 - Low latency
 - High bandwidth
 - High message rate
 - Good overlap of computation with communication
- Hardware Offloaded Tag Matching
- Can we exploit tag matching support in UCX into existing HPC middleware to extract peak performance and overlap?

Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
 - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.1), Started in 2001, First version available in 2002 (SC '02)
 - MVAPICH2-X (MPI + PGAS), Available since 2011
 - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
 - Support for Virtualization (MVAPICH2-Virt), Available since 2015
 - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
 - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
 - **Used by more than 3,050 organizations in 89 countries**
 - **More than 615,000 (> 0.6 million) downloads from the OSU site directly**
 - Empowering many TOP500 clusters (Nov '19 ranking)
 - 3rd, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center i
 - 5th, 448, 448 cores (Frontera) at TACC
 - 8th, 391,680 cores (ABCI) in Japan
 - 14th, 570,020 cores (Neurion) in South Korea and many others
 - Available with software stacks of many vendors and Linux Distro (RedHat, SuSE, and OpenHPC)

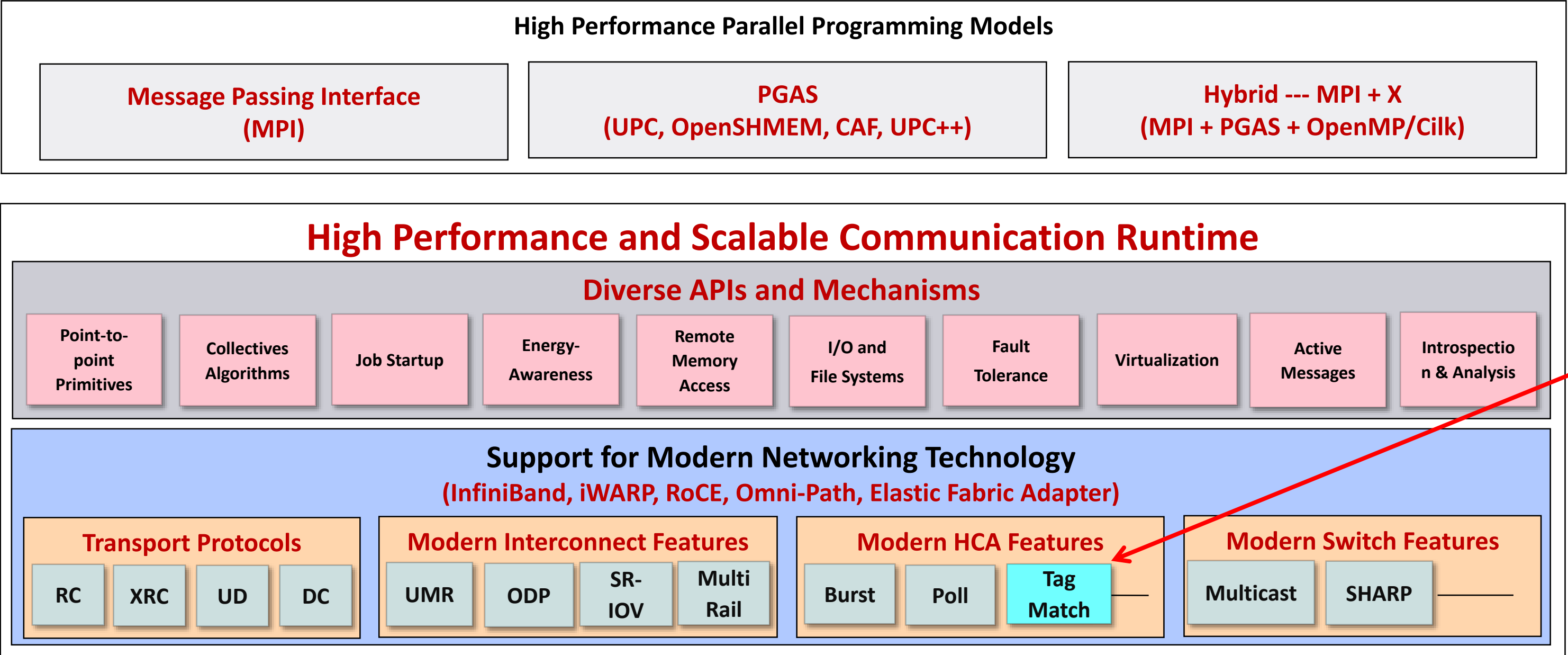


– <http://mvapich.cse.ohio-state.edu>

Partner in the #5th TACC Frontera System

- Empowering Top500 systems for over a decade

The MVAPICH Approach

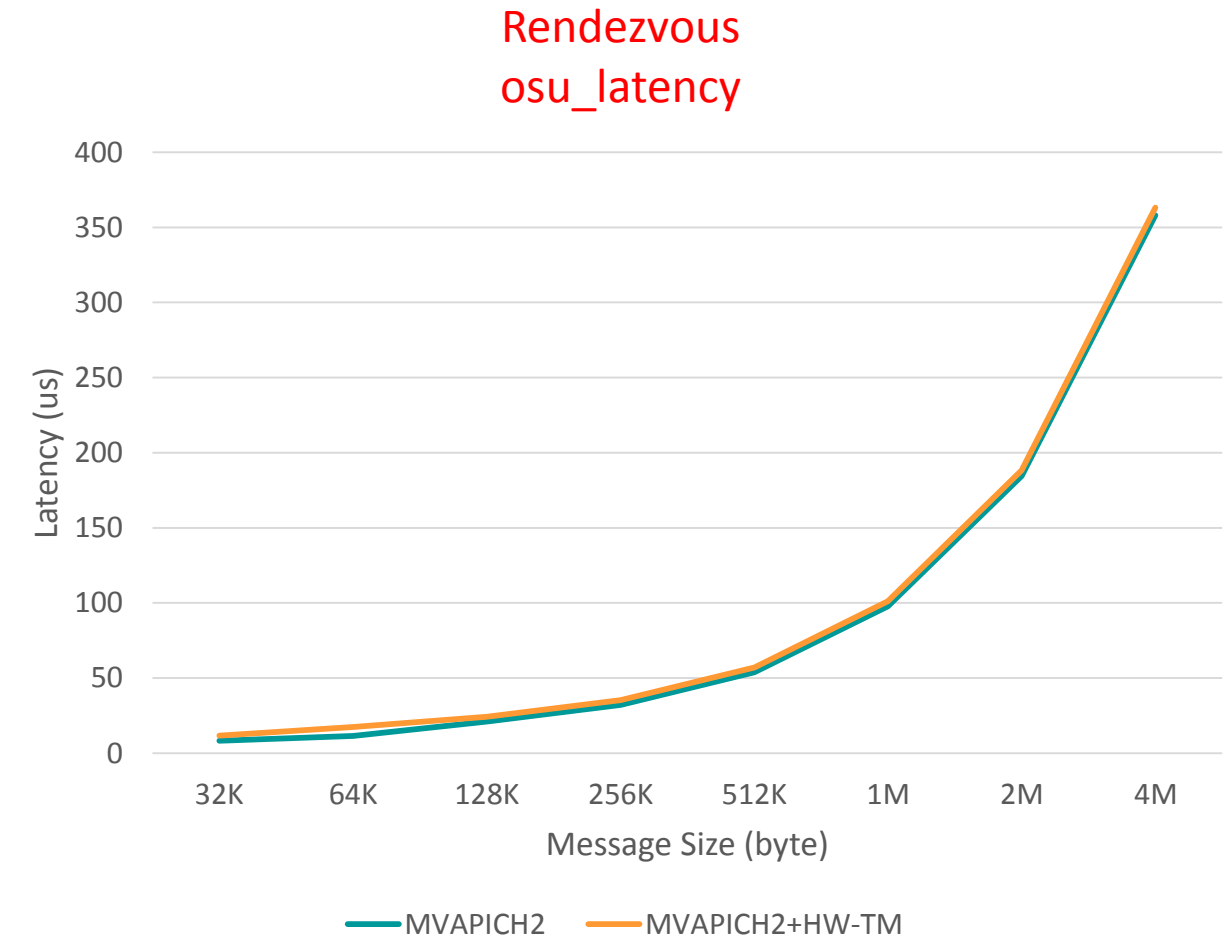
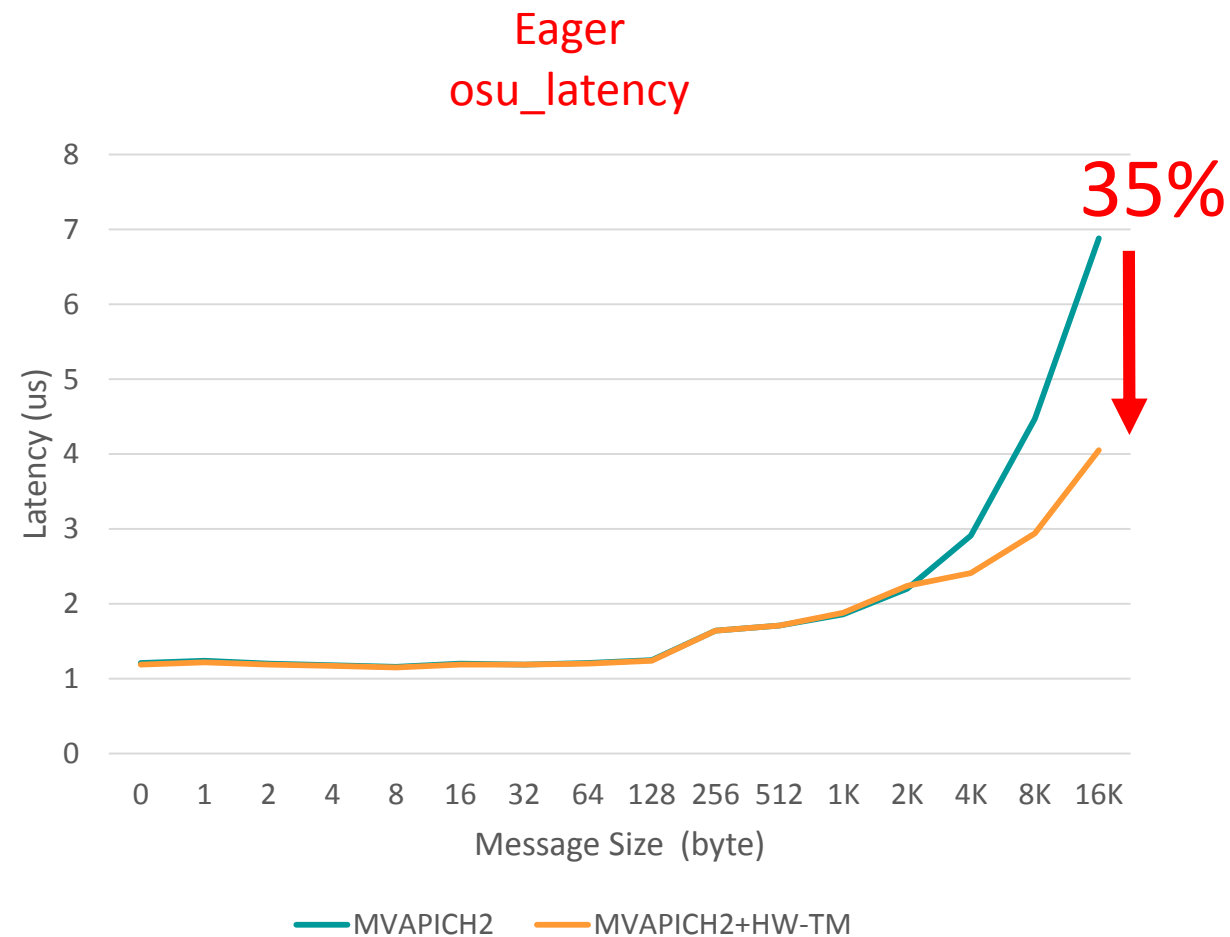


* Upcoming

Hardware Tag Matching Support

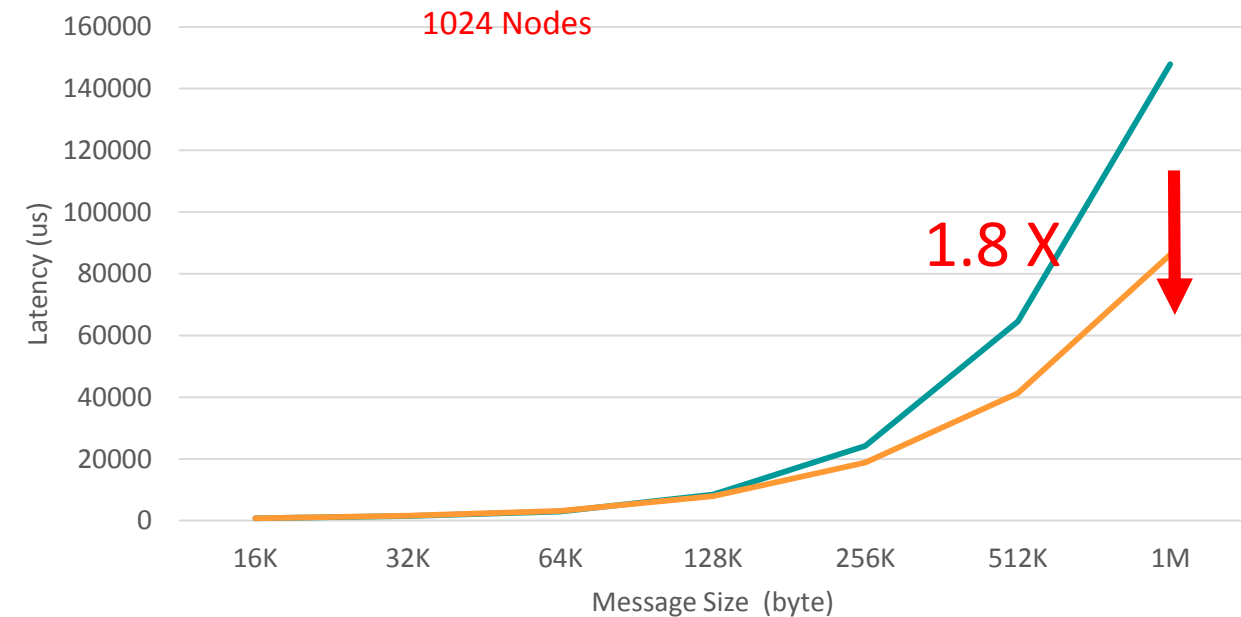
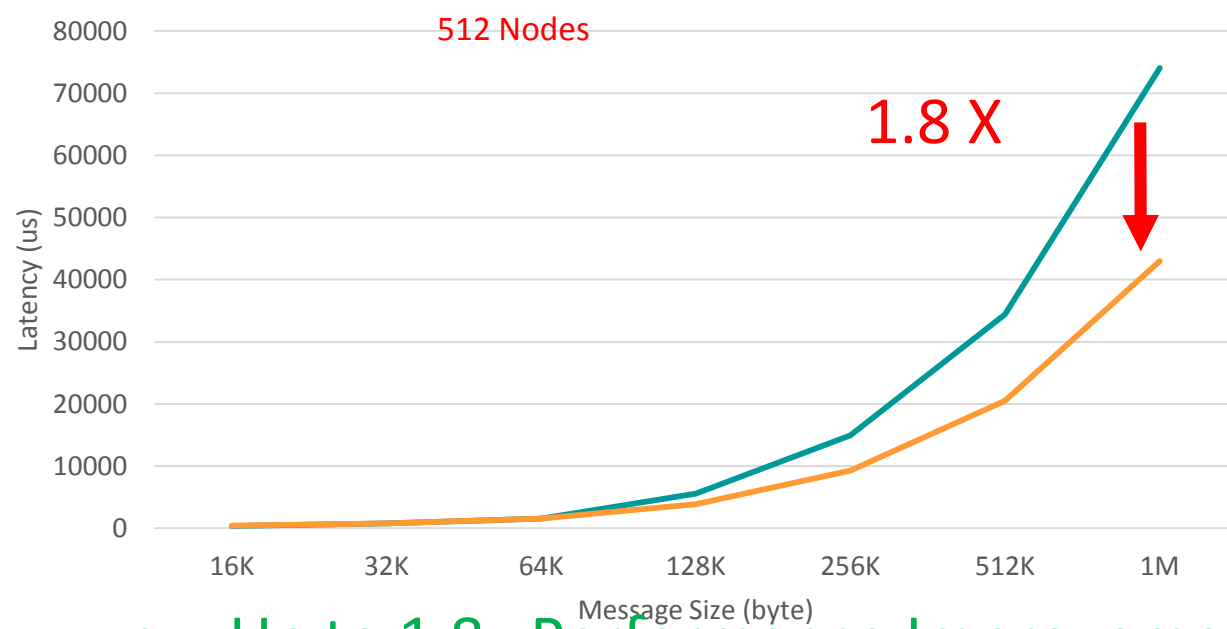
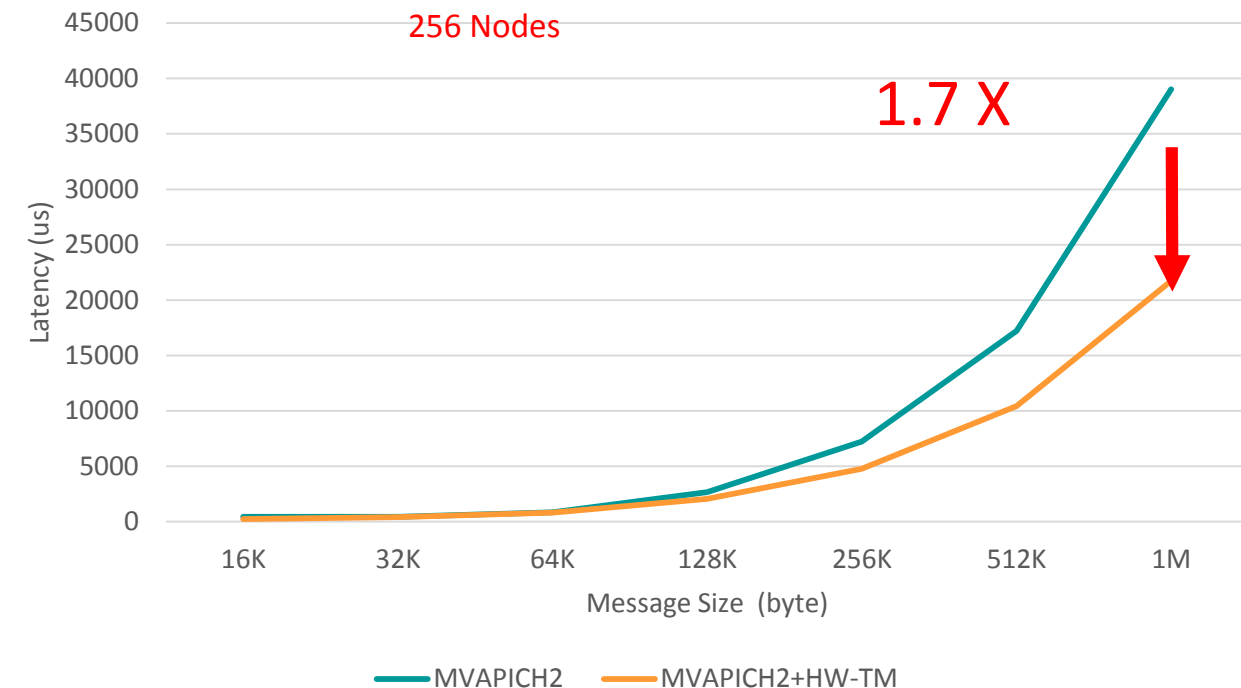
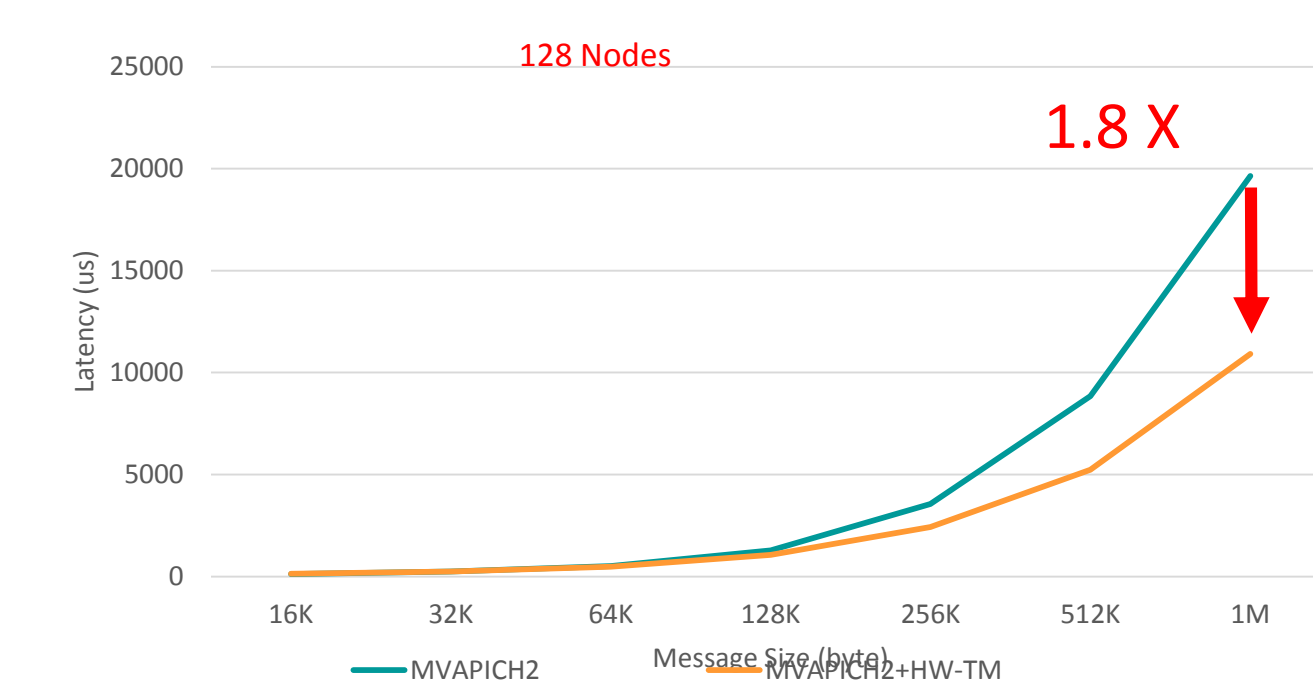
- Offloads the processing of point-to-point MPI messages from the host processor to HCA
- Enables zero copy of MPI message transfers
 - Messages are written directly to the user's buffer without extra buffering and copies
- Provides rendezvous progress offload to HCA
 - Increases the overlap of communication and computation

Impact of Zero Copy MPI Message Passing using HW Tag Matching (Point-to-point)



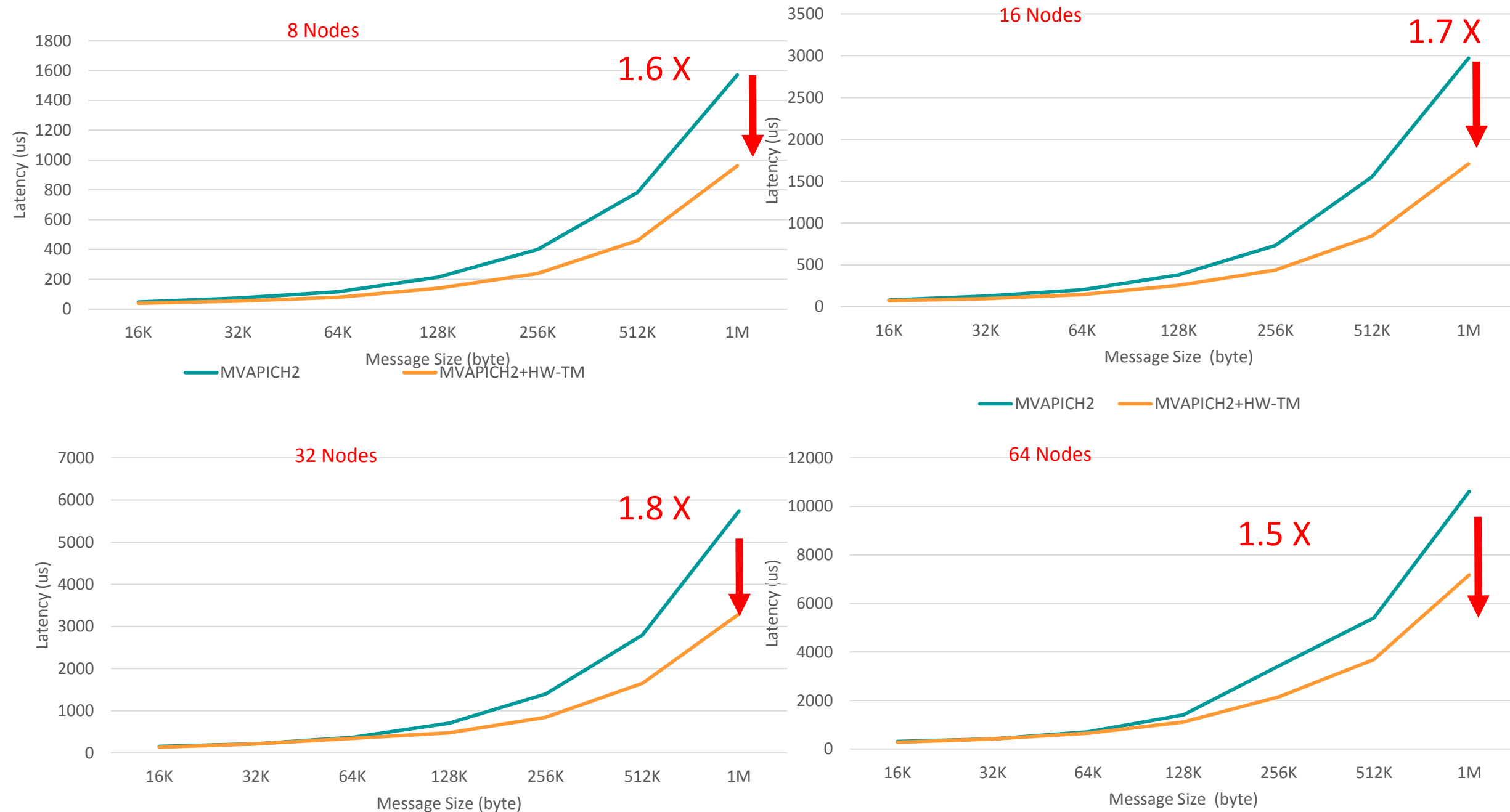
Removal of intermediate buffering/copies can lead up to 35% performance improvement in latency of medium messages on TACC Frontera

Performance of MPI_Isscatterv using HW Tag Matching on Frontera



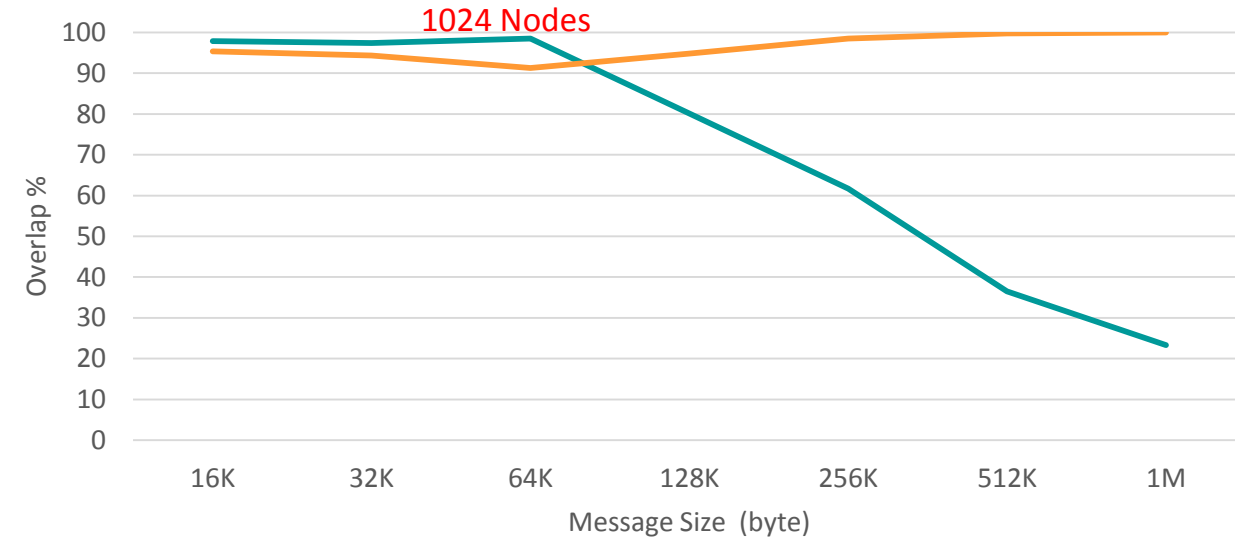
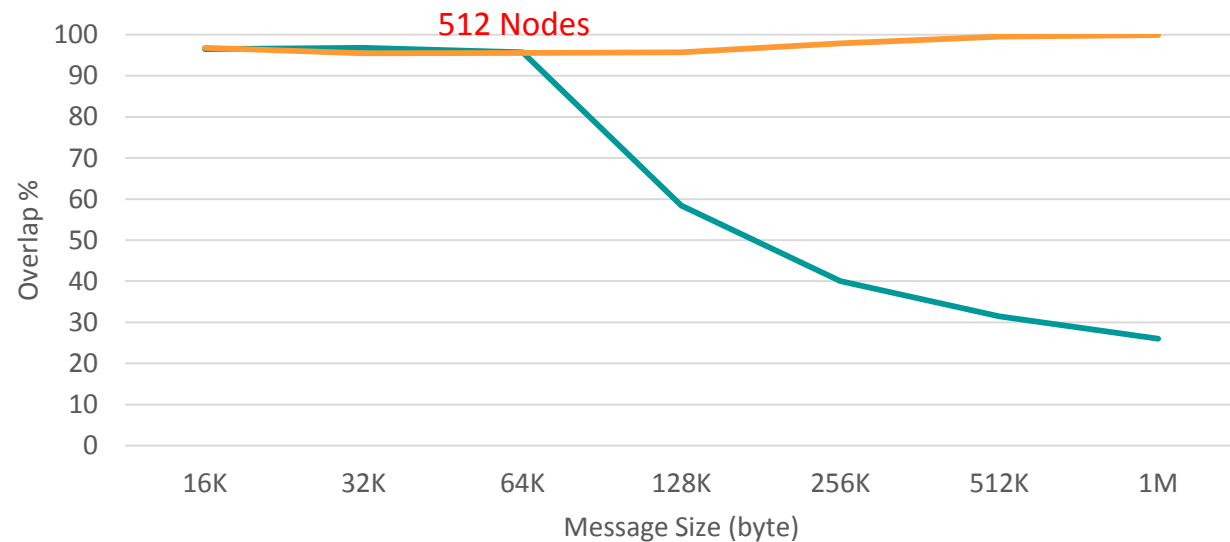
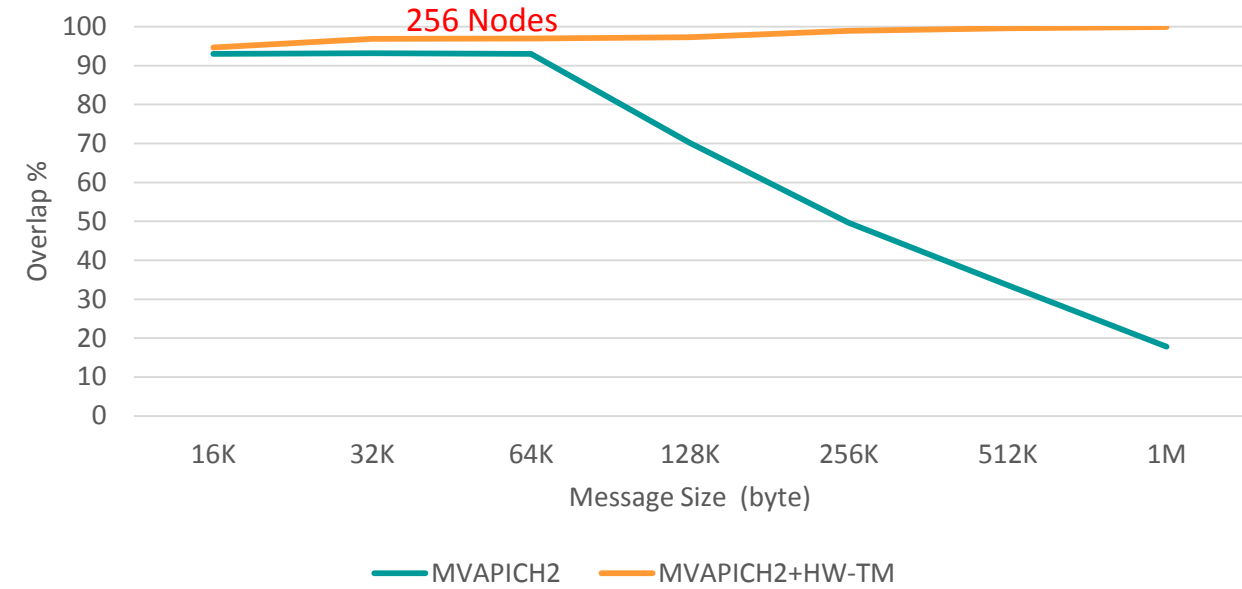
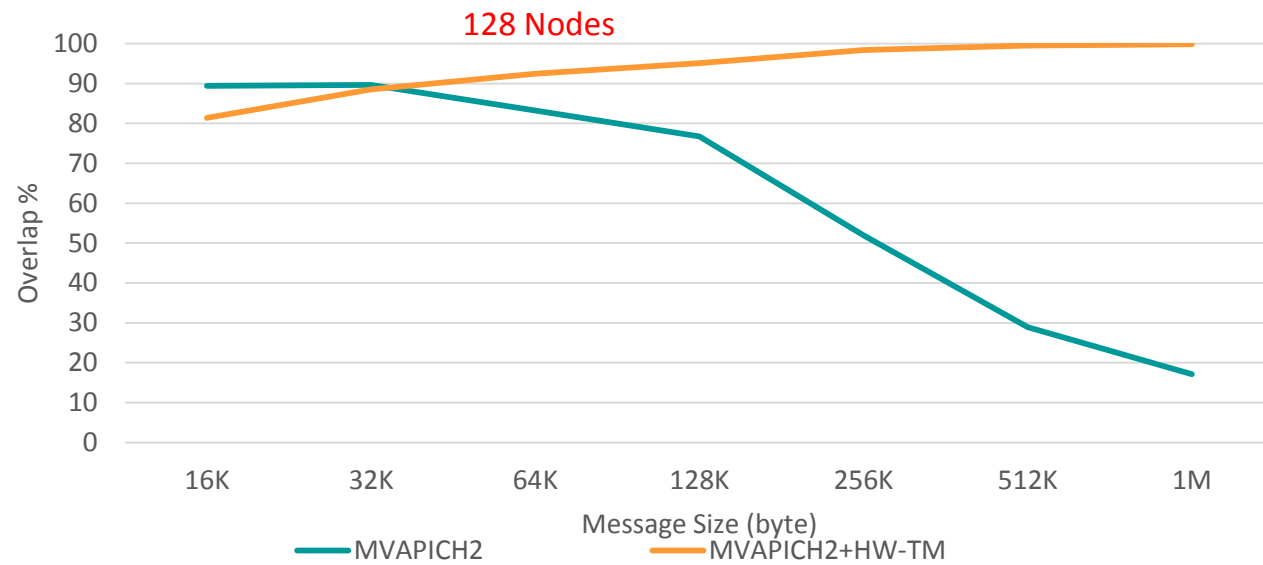
- Up to 1.8x Performance Improvement
- Sustained benefits as system size increases

Performance of MPI_alltoall using HW Tag Matching on Frontera



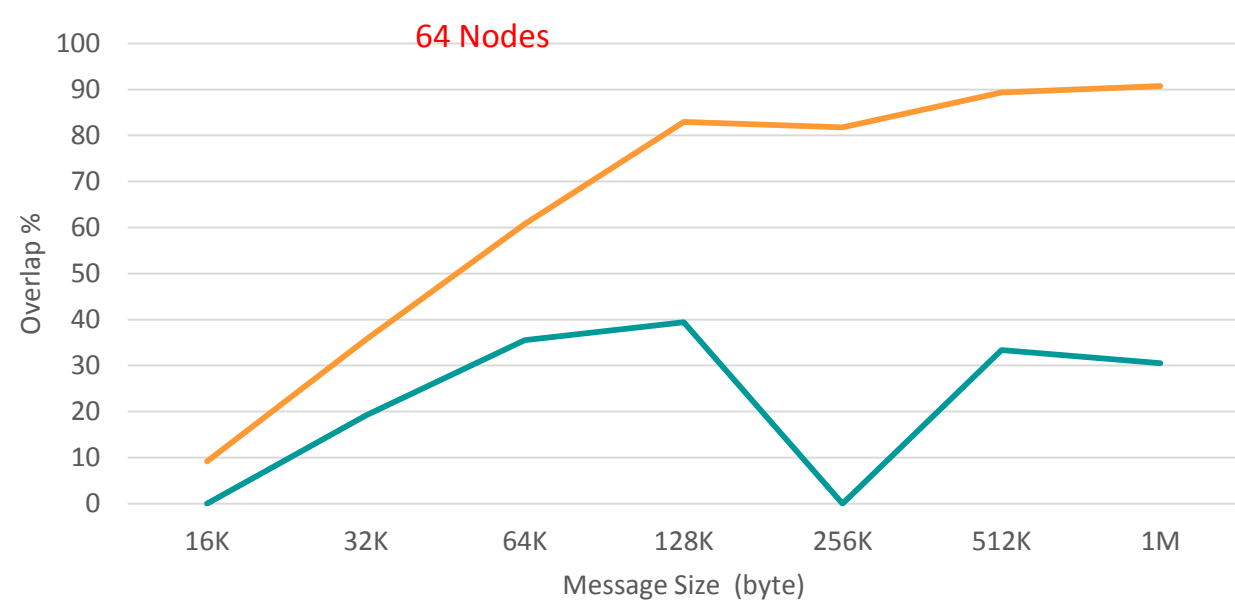
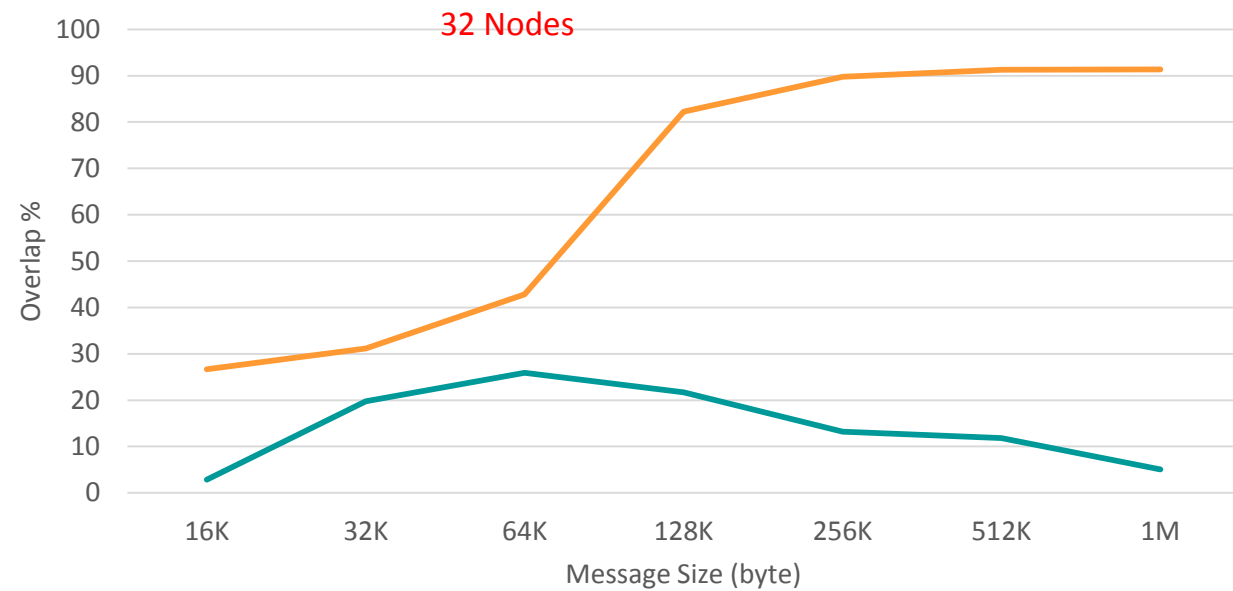
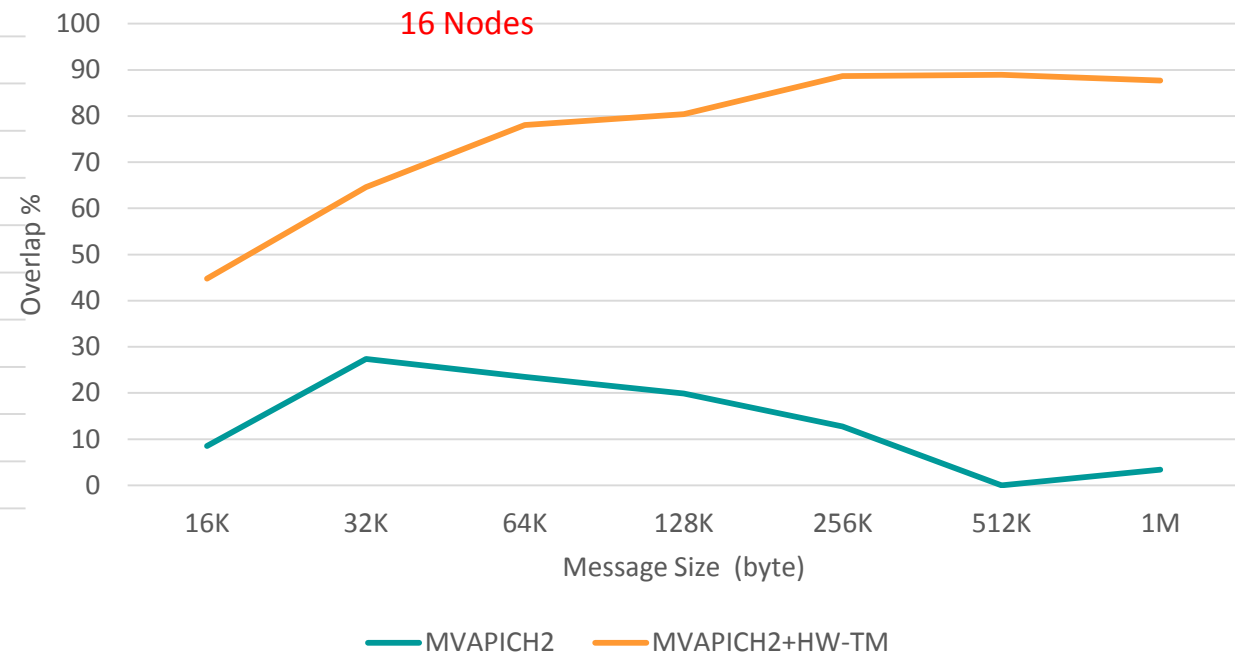
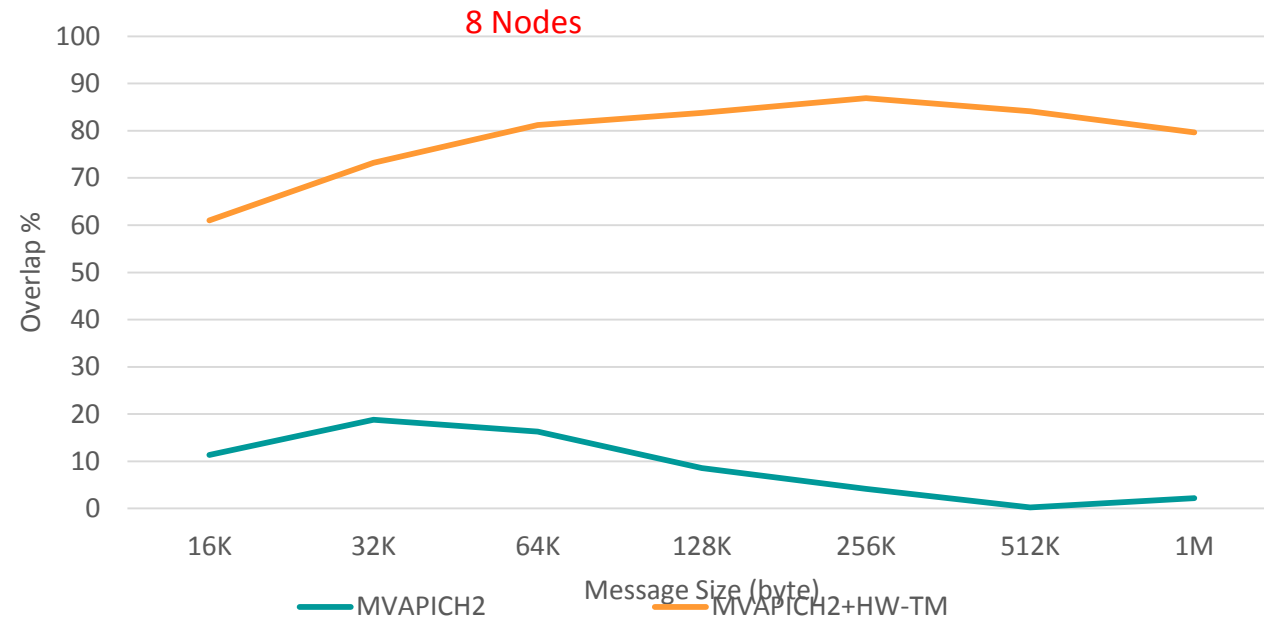
- Up to 1.8x Performance Improvement
- Sustained benefits as system size increases

Overlap with MPI_Isscatterv using HW Tag Matching on Frontera



- Maximizing the overlap of communication and computation
- Sustained benefits as system size increases

Overlap with MPI_lalltoall using HW Tag Matching on Frontera



- Maximizing the overlap of communication and computation
- Sustained benefits as system size increases

Future Plans

- Complete designs are being worked out
- Will be available in the future MVAPICH2 releases

Multiple Events at SC '19

- Presentations at OSU Booth (#2094)
 - Members of the MVAPICH, HiBD and HiDL members
 - External speakers
- Presentations at SC main program (Tutorials, Workshops, BoFs, Posters, and Doctoral Showcase)
- Presentation at many other booths (Mellanox, Intel, Microsoft, and AWS) and satellite events
- Complete details available at

<http://mvapich.cse.ohio-state.edu/conference/752/talks/>

ENABLER OF CO-DESIGN



Thank You

The UCF Consortium is a collaboration between industry, laboratories, and academia to create production grade communication frameworks and open standards for data centric and high-performance applications.