

ENABLER OF CO-DESIGN



Unified Communication X (UCX)

UCF Consortium Project

ISC 2019

■ Mission:

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

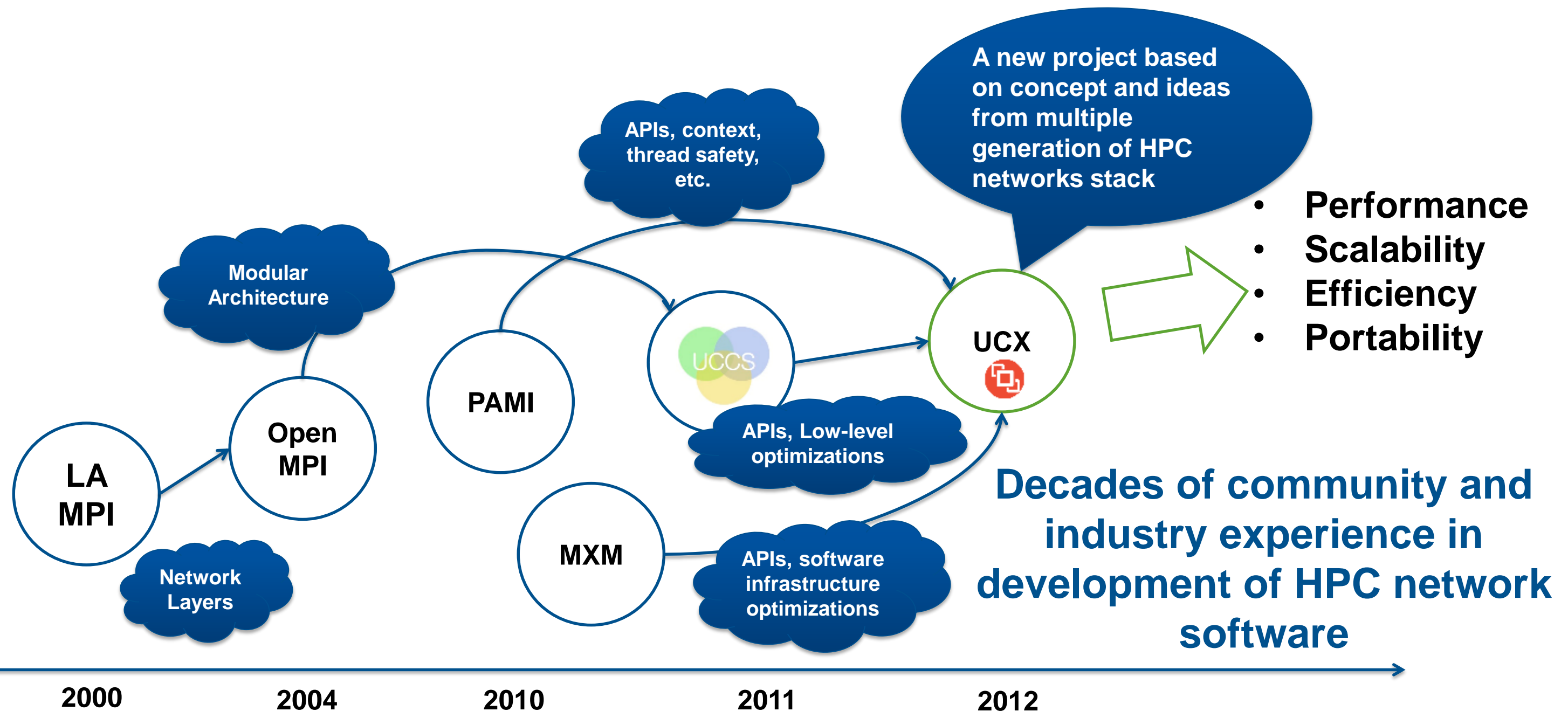
■ Projects

- UCX – Unified Communication X
- Open RDMA

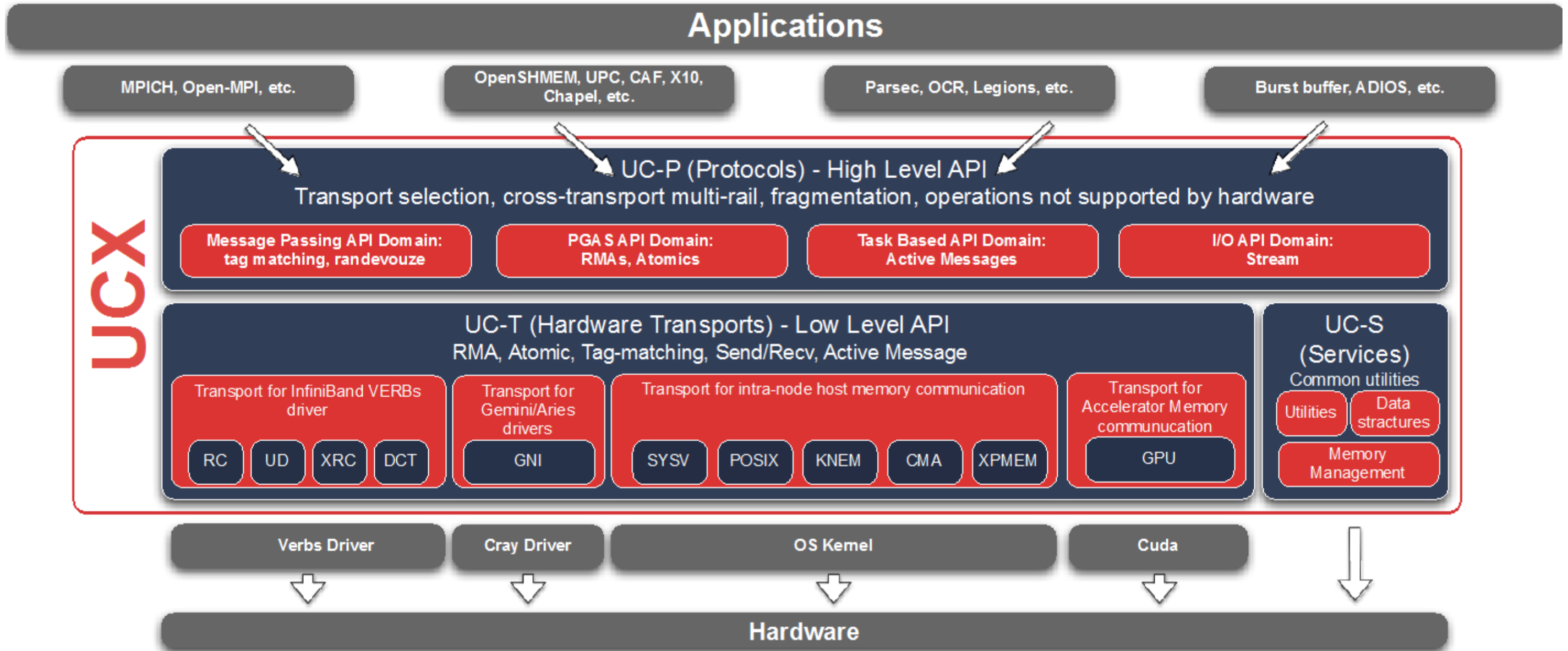
■ Board members

- **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)





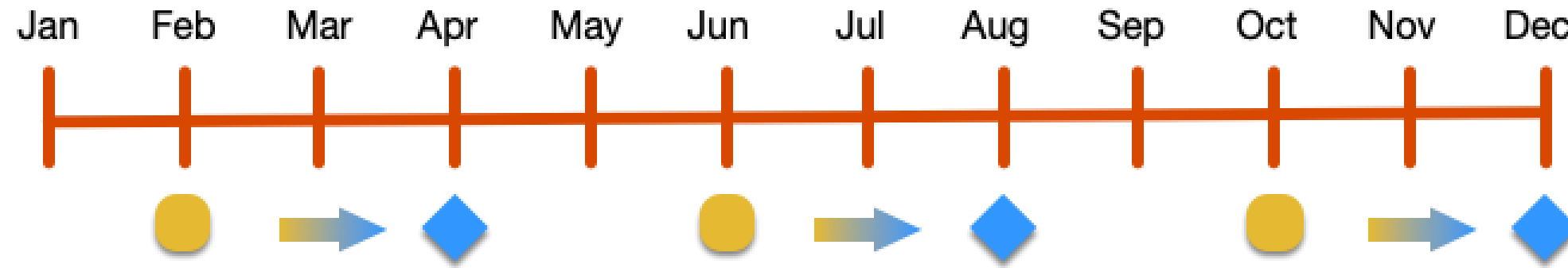
UCX High-level Overview



- UCX is a framework for network APIs and stacks
- UCX aims to unify the different network APIs, protocols and implementations into a single framework that is portable, efficient and functional
- UCX doesn't focus on supporting a single programming model, instead it provides APIs and protocols that can be used to tailor the functionalities of a particular programming model efficiently
- When different programming paradigms and applications use UCX to implement their functionality, it increases their portability. As just implementing a small set of UCX APIs on top of a new hardware ensures that these applications can run seamlessly without having to implement it themselves


UCX Development Status

UCX annual release schedule



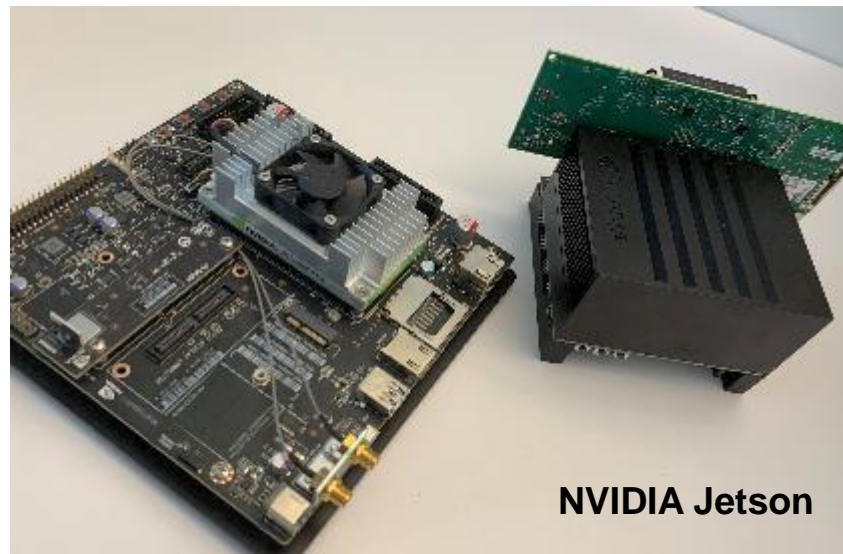
- v1.6.0 - April '19
- v1.7.0 - August '19
- v1.8.0 - December '19

 Major release

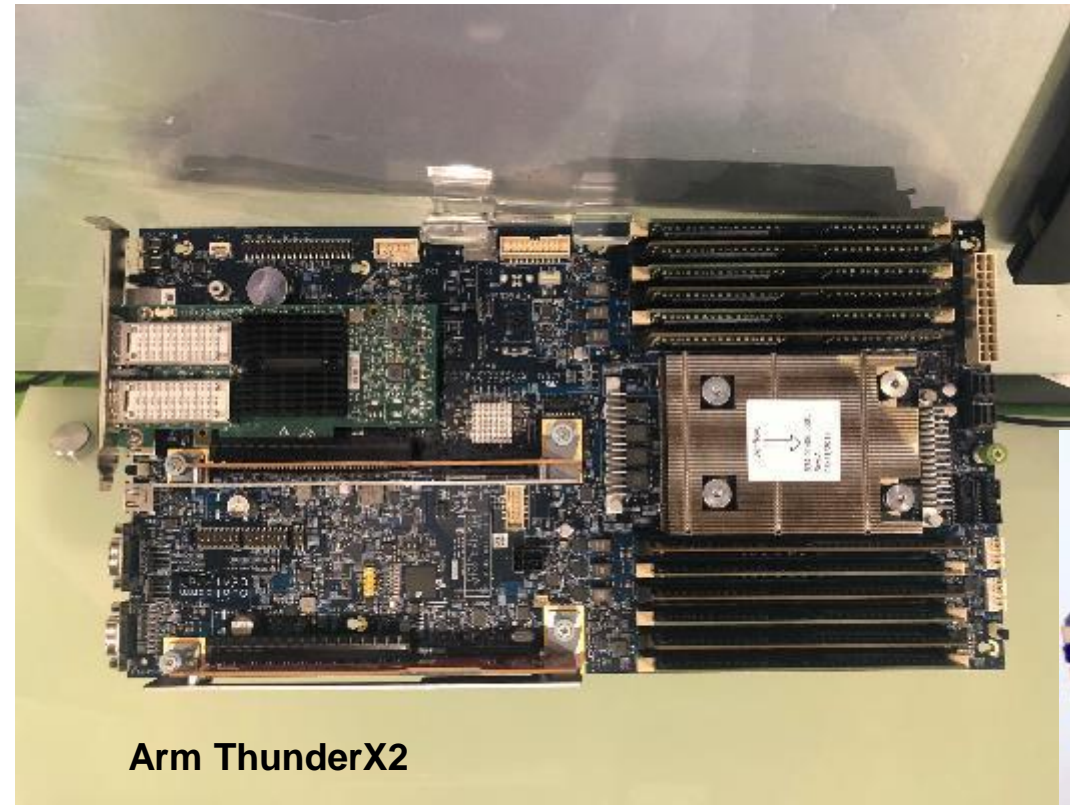
 Feature freeze
(release branch fork)



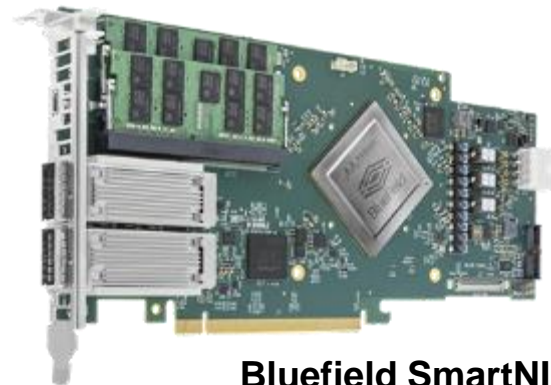
- Support for x86_64, Power 8/9, Arm v8
- Runs on Servers, Raspberry PI like platforms, SmartNIC, Nvidia Jetson platforms, etc.



NVIDIA Jetson



Arm ThunderX2



Bluefield SmartNIC



Odroid C2

- UCP emulation layer (atomics, rma)
- Non-blocking API for all one-sided operations
- Client/server connection establishment API
- Malloc hooks using binary instrumentation instead of symbol override
- Statistics for UCT tag API
- GPU-to-Infiniband HCA affinity support based on locality/distance (PCIe)
- GPU - Support for stream API and receive side pipelining

- AMD GPU ROCm transport re-design: support for managed memory, direct copy, ROCm GDR
- Modular architecture for UCT transports
- Random scheduling policy for DC transport
- OmniPath support (over verbs)
- Optimized out-of-box settings for multi-rail
- Support for PCI atomics with IB transports
- Reduced UCP address size for homogeneous environments

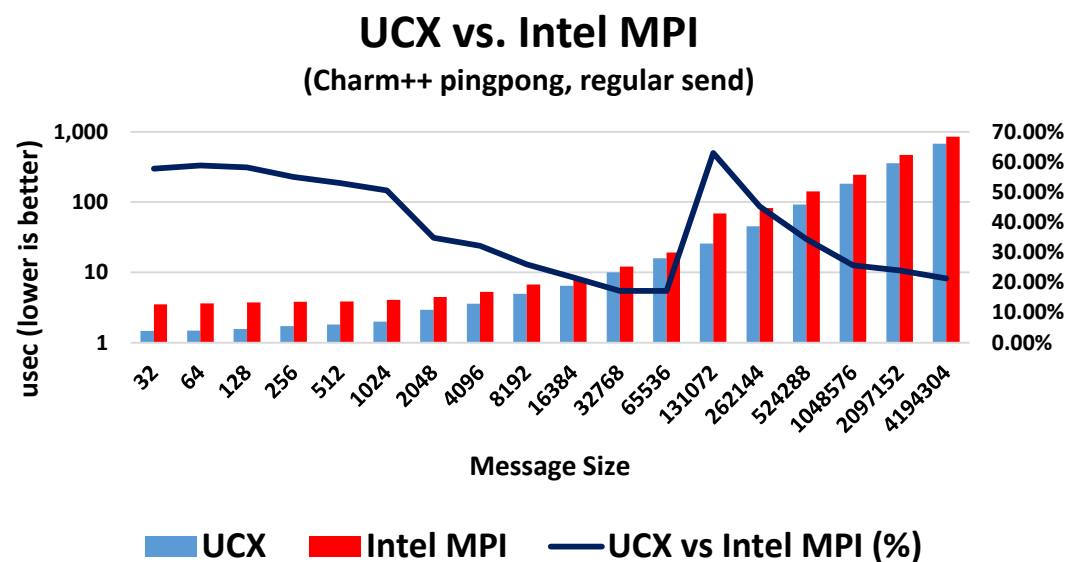
- **Python** bindings ! + Integration with **Dask** and **Rapids AI**
- **Java** bindings ! + Integration with **SparkRDMA**
- **iWARP** support
- **GasNET** over UCX
- **Collectives**
- **FreeBSD** support
- **MacOS** support
- Moving CI to **Jenkins** and **Azure Pipelines**
- **TCP performance** optimizations
- **Hardware tag matching** optimizations

UCX Machine Layer in Charm++

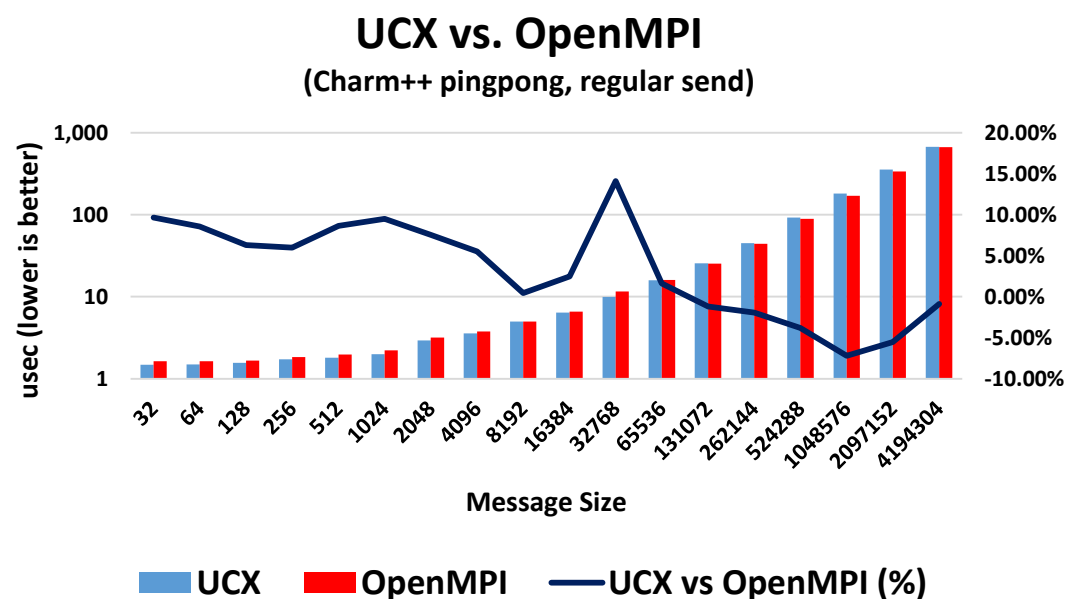
- Charm++ is an object-oriented and message-driven parallel programming model with an adaptive runtime system that enables high-performance applications to scale.
- The Low-level Runtime System (LRTS) is a thin software layer in the Charm++ software stack that abstracts specific networking functionality, which supports uGNI, PAMI, Verbs, MPI, etc.
- UCX is a perfect fit for Charm++ machine layer:
 - Just ~1000 LoC is needed to implement all LRTS APIs (MPI takes ~2700 LoC, Verbs takes ~8000LoC)
 - UCX provides ultra low latency and high bandwidth sitting on top of RDMA Verbs stack
 - UCX provides much less intrusive and close-to hardware API for one-sided communications than MPI

Charm++ over UCX (Performance Evaluations)

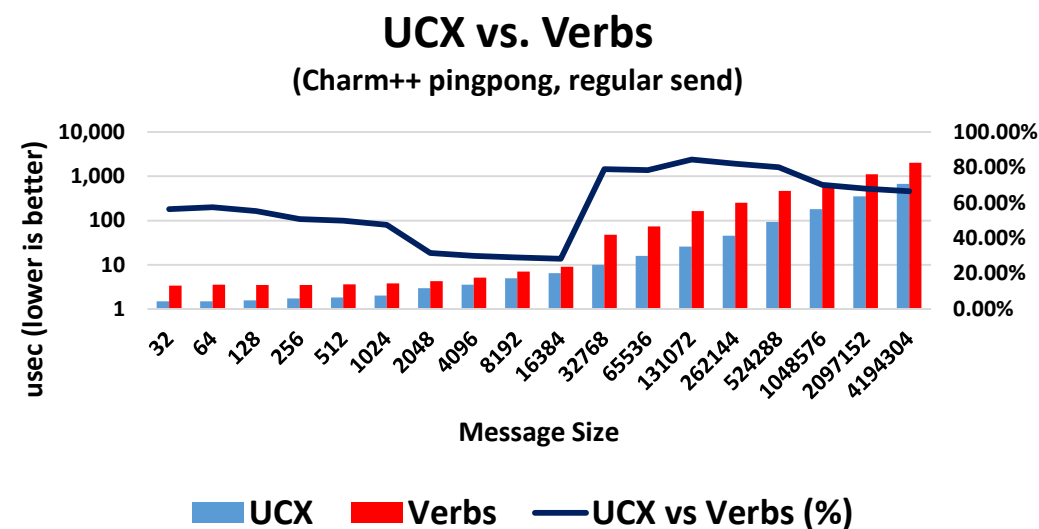
- Up to 63% better than Intel MPI



- Up to 15% better than Open MPI (thru UCX pml)

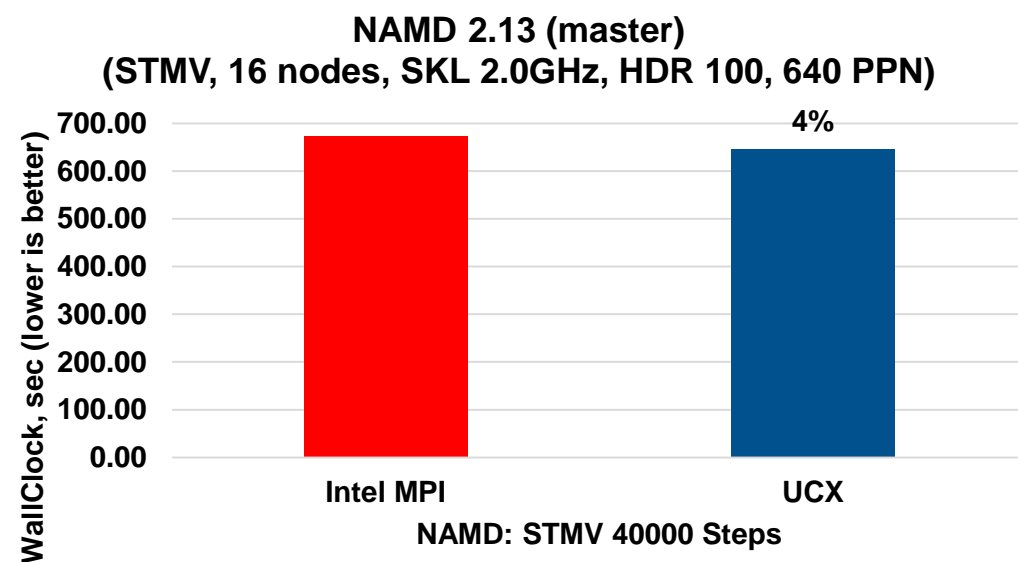


Charm++ over UCX (Performance Evaluations)



- Up to 85% better than Verbs

- 4% improvement over Intel MPI with NAMD (STMV public input, 40000 steps)



UCX Support in MPICH

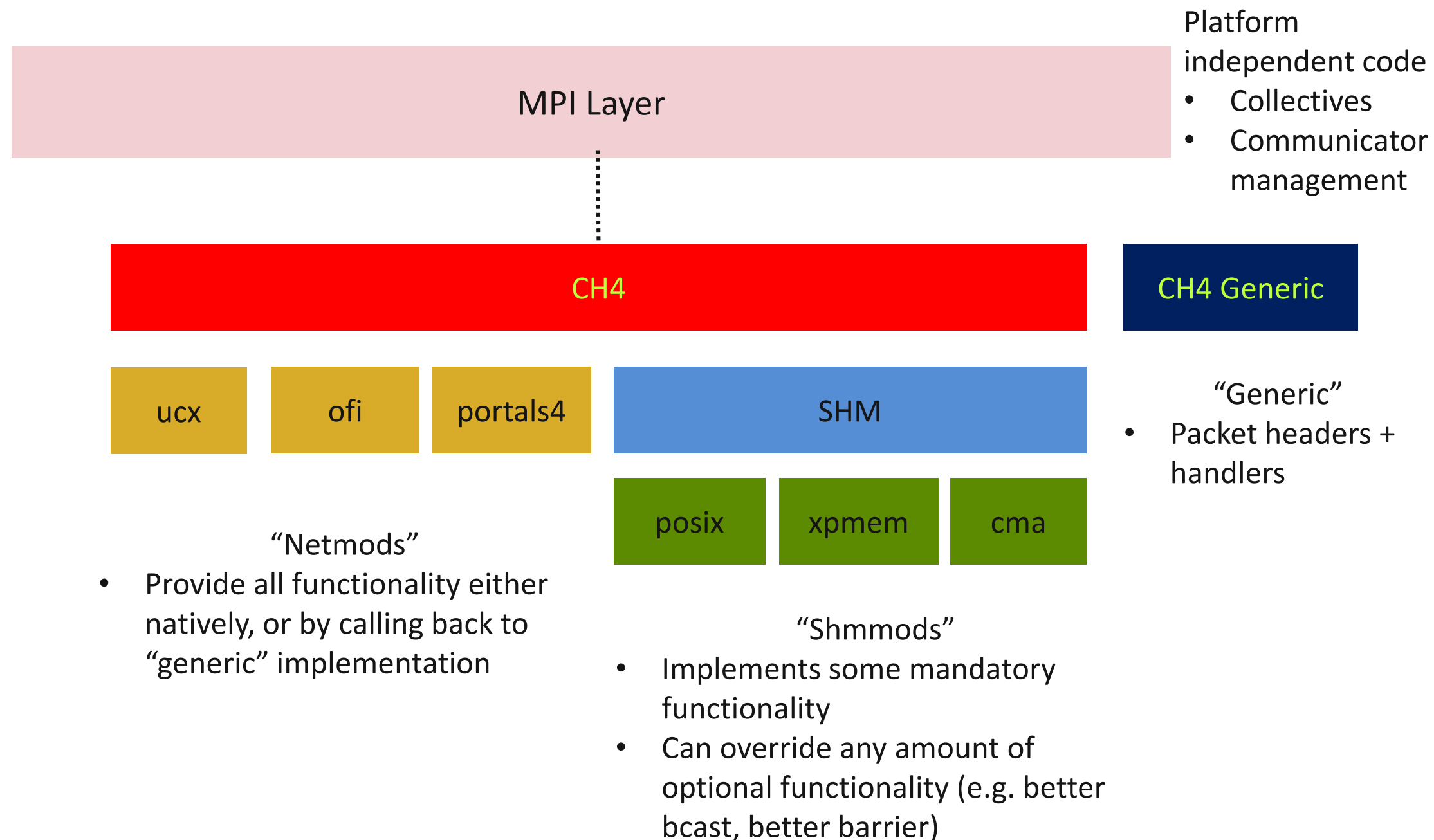
Yanfei Guo

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

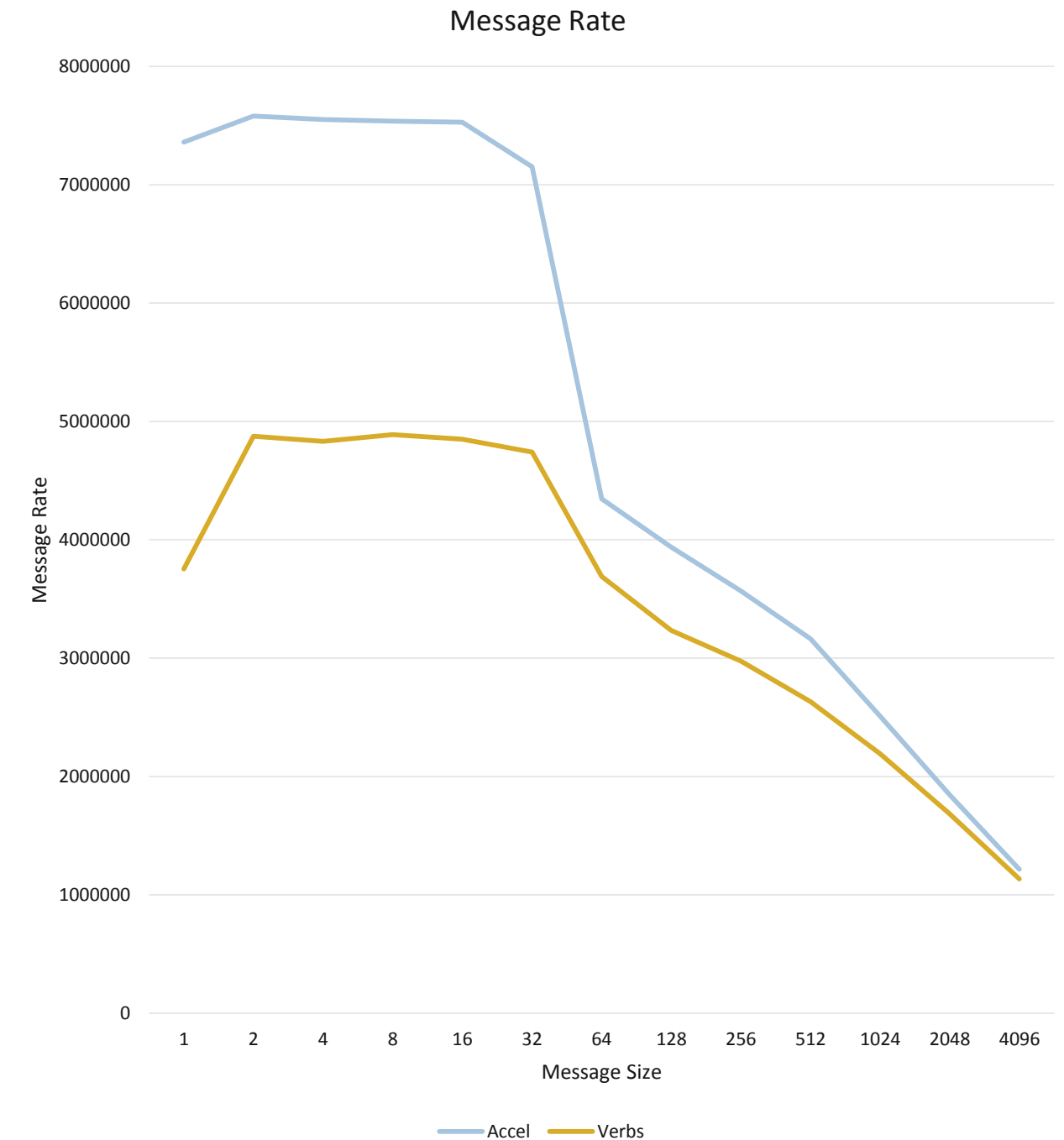
- UCX_TLS=rc_mlx5,cm
- Lower overhead
 - Low latency
 - Higher message rate

OSU Latency: **0.99us**

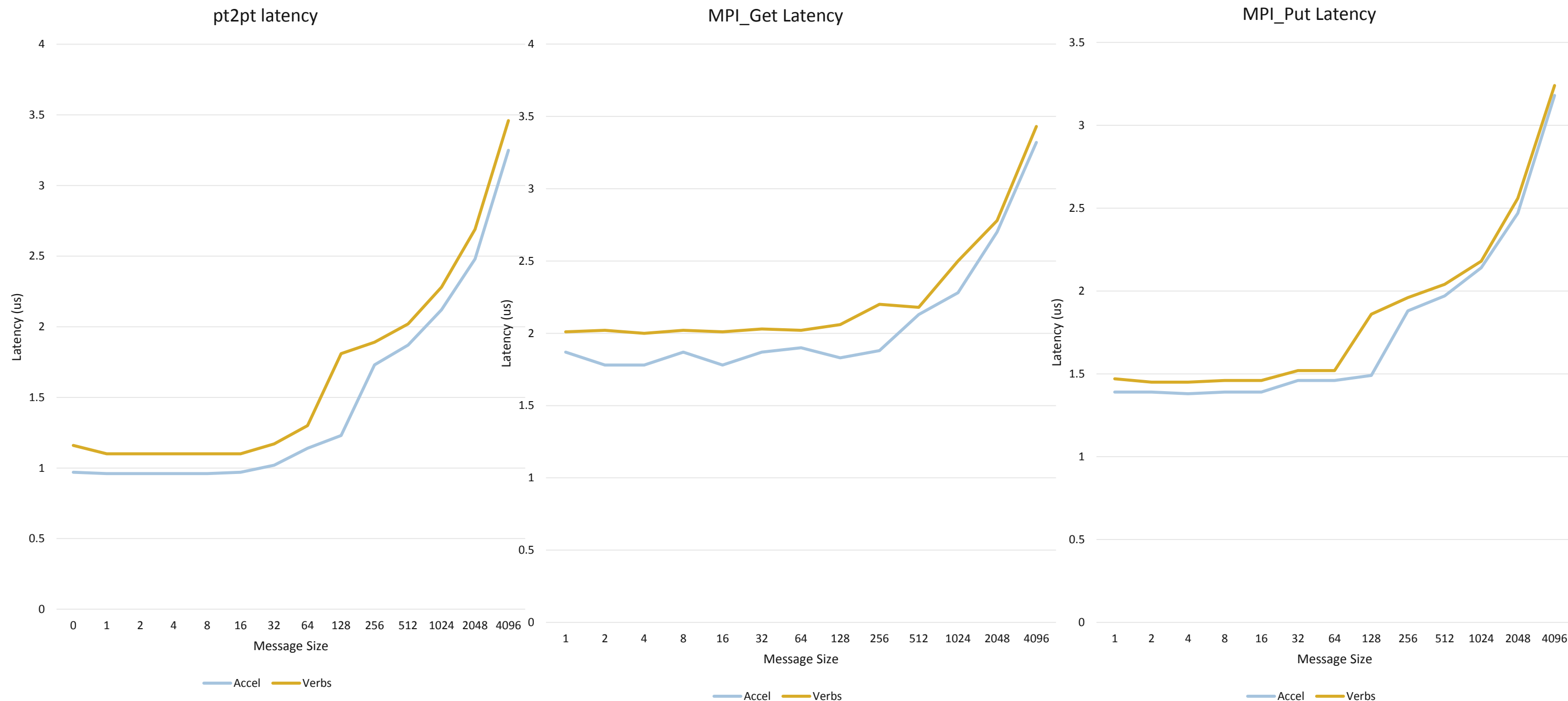
OSU BW: **12064.12 MB/s**

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



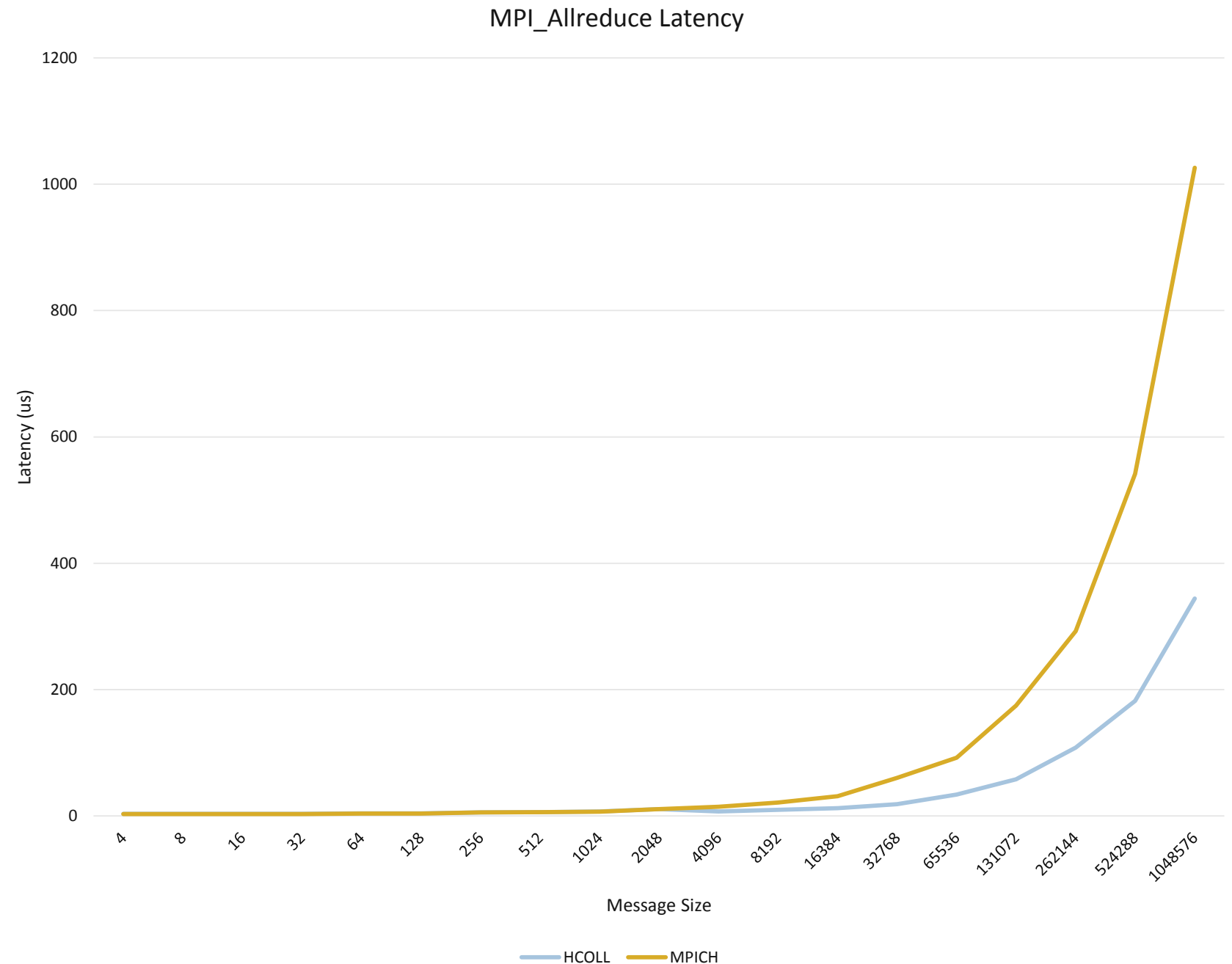
MPICH/UCX with Accelerated Verbs



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 3.3.1 just released
 - Includes an embedded UCX 1.5.0
- 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 hackathons with Mellanox and NVIDIA

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



Delivering science and technology
to protect our nation
and promote world stability



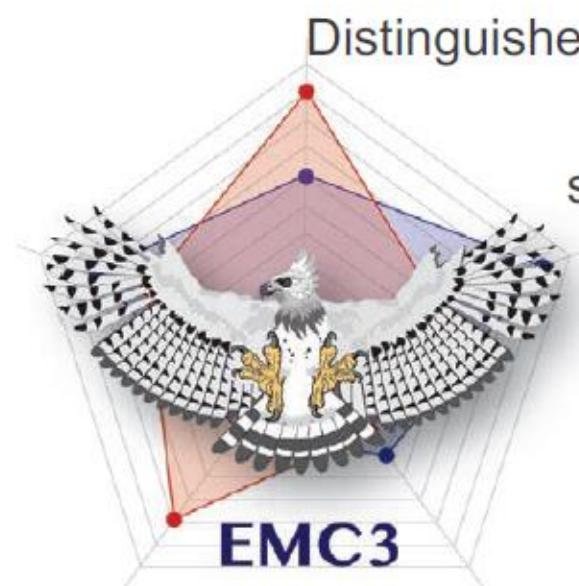
Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

ISC-2019-OpenUCX

ISC 2019
Frankfurt Germany

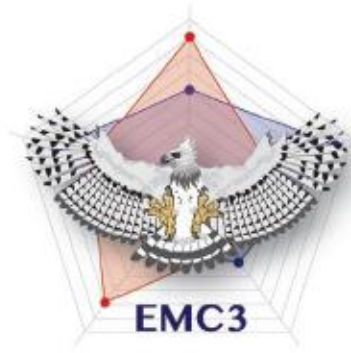
Stephen Poole

Distinguished Senior Scientist
Chief Architect
swpoole@lanl.gov



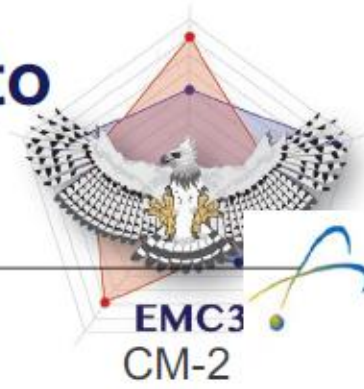
LA-UR-19-25420

Very Rough Outline



- **Brief history of computing at LANL**
 - We have an entire history project on Computing @LANL.
 - [Http://history.lanl.gov](http://history.lanl.gov)
- **What does LANL/Environment look like today**
- **LANL's Mission (What/How?)**
- **Why would we care about UCX?**

Eight Decades of Production Weapons Computing to Promote Science and Keep the Nation Safe



Maniac



IBM Stretch



CDC



Cray 1



Cray X/Y



EMC3
CM-2



CM-5



SGI Blue Mountain



DEC/HP Q



IBM Cell Roadrunner



Cray XE Cielo



Cray Intel KNL Trinity



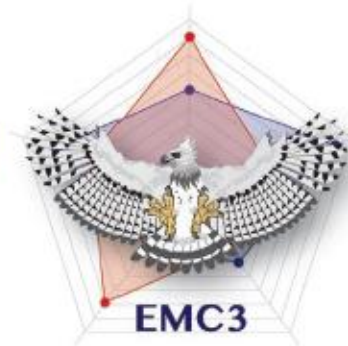
Ising DWave



Cross Roads



Los Alamos has been at the forefront of developing high-speed network interconnects for decades



- Los Alamos developed one of the first networks, named Hydra, to allow common access to the five CDC machines, 1972
- High-Speed Parallel Interface (HSPI) for inter-computer communication, 50 Mbit/s, 1979-1982
- High-Performance, Parallel Interface (HIPPI), the first gigabit network, 1987
- Gigabyte System Network - GSN 1990's
- Infiniband interconnect came out of ASCI work in the late 1990's
- Optical interconnects started by ASCI ~2000
 - Analysis of optical switches
- QKD – Early 2000's (Spun off)
 - Free space quantum optics
- In Situ using COTS
 - Prime candidate for UCX

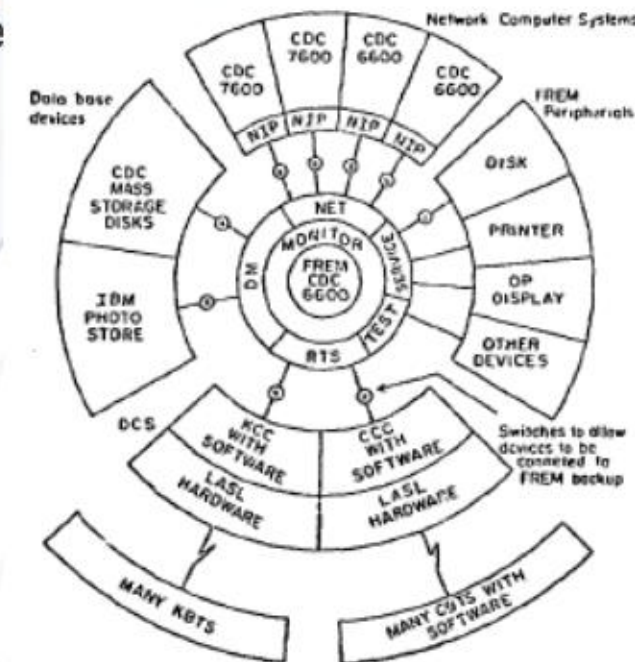
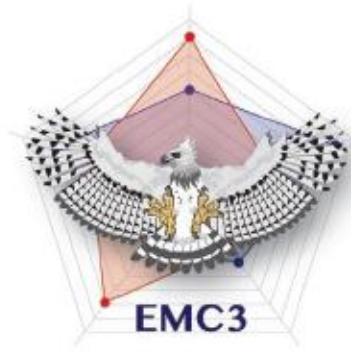


Fig. 1. Hydra system diagram.

Hydra network design, 1972

High-speed interconnects are mostly a standard commercial off-the-shelf technology now

Sustain & Extend ASCI/ASC Application Investments Made During Post-Testing Era



Multi-Physics Codes

- Eulerian Application Project (EAP) Codes
 - Direct Eulerian / Adaptive Mesh Refinement (AMR)
- Lagrangian Application Project (LAP) Codes
 - Lagrange / Arbitrary Eulerian-Lagrangian (ALE)

Single-Physics Codes

- Transport Codes: PARTISN & IMC

Weapons Science Codes

- Plasma Kinetic Code: VPIC

VPIC (Vector Particle-in-Cell)

- 3D Explicit, Relativistic, Charge-Conserving Electromagnetic Particle-in-Cell (PIC) Code
 - Single Precision, Structured Meshes

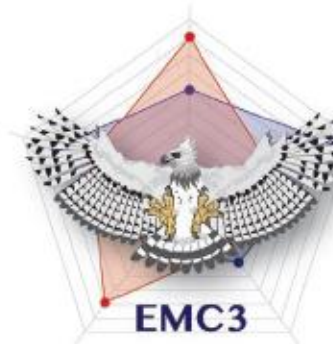
First US Test: Trinity 1945



Last US Test:
Divider
1992



HPC Simulation Background

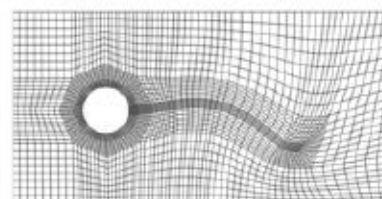
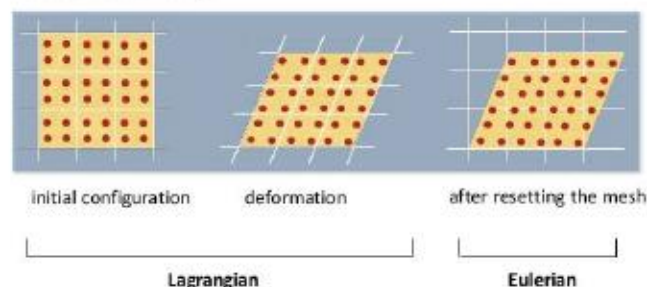


Link scales

<http://eng-cs.syr.edu/research/mathematical-and-numerical-analysis>

Methods

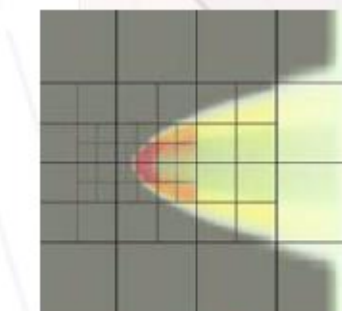
in each calculation step :



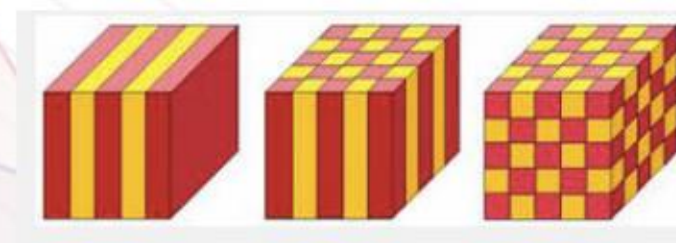
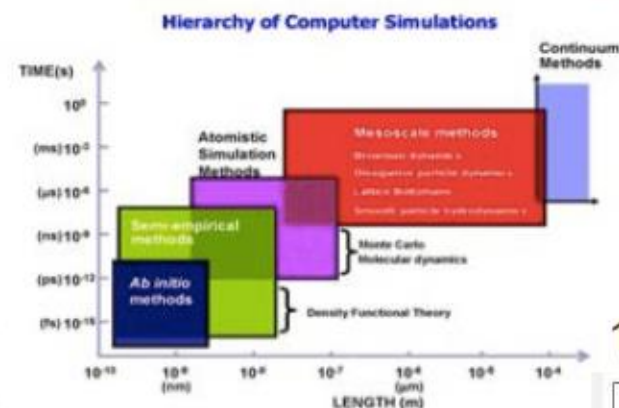
<http://media.archnumsoft.org/10305/>

ALE

http://web.cs.ucdavis.edu/~ma/VolVis/amr_mesh.jpg

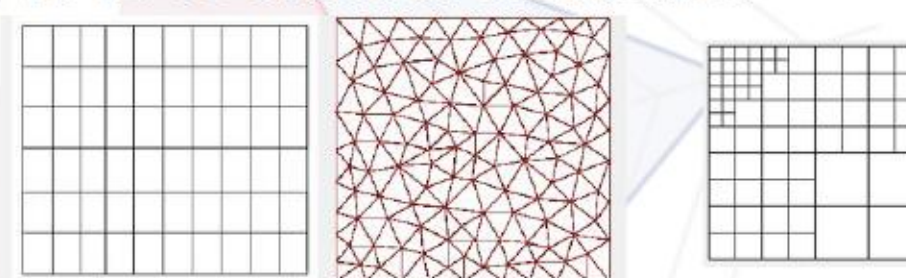


Eulerian AMR



Meshes

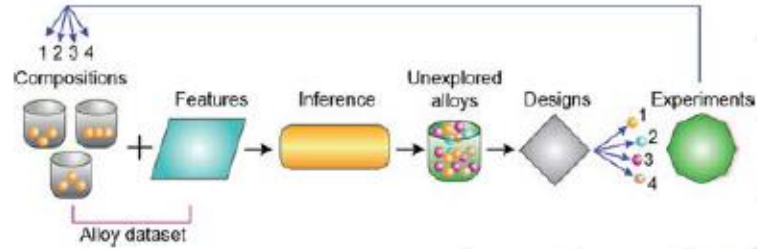
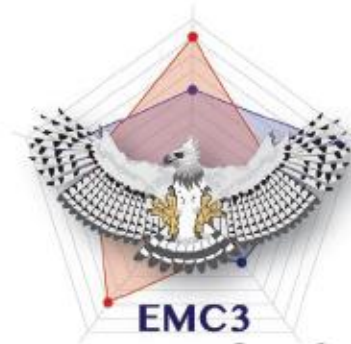
1D 2D 3D Structured Unstructured



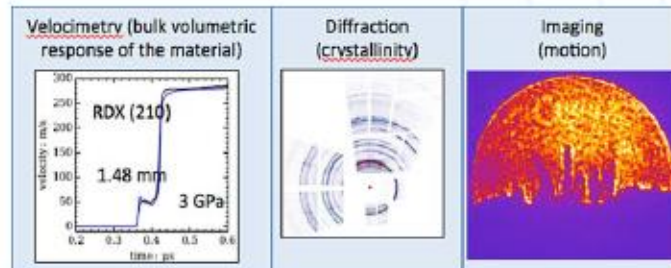
The “simultaneous” challenge:

- Multi-Physics, Multi-Scales
- Multi-Resolution, Multi-method
- Spread across >> 1M memories in multiple levels of memory
- On >>> M's of cores, including heterogeneous cores, that are clocking up and down to save power/heat
- On machines that have a mean time to interrupt of few hours headed towards tens of minutes
- For 6-18 months
- Be aware of numerical instabilities
- Maybe (future) inexact?
- Run ML/DA on simulations for future improvements

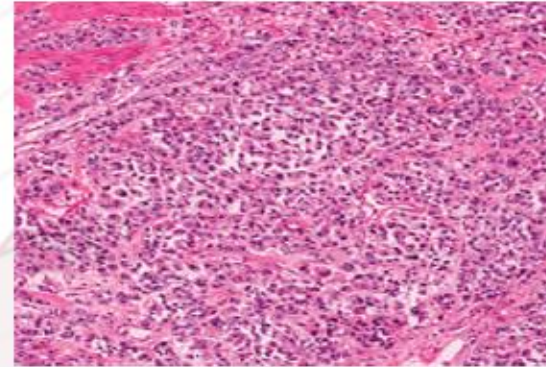
Data analytics is emerging organically and rapidly across many programs



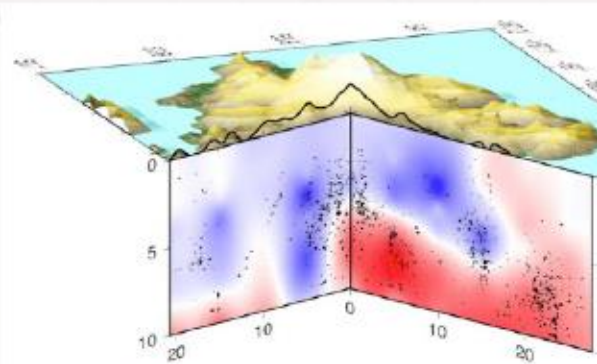
Machine Learning
Accelerating Discovery
of New Materials, old
Materials



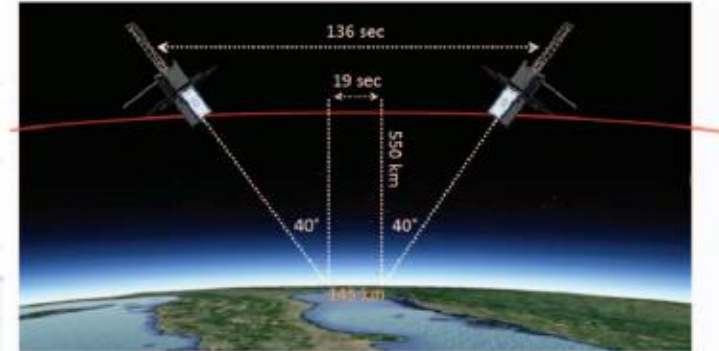
Real-time Adaptive
Acceleration of Dynamic
Experimental Science



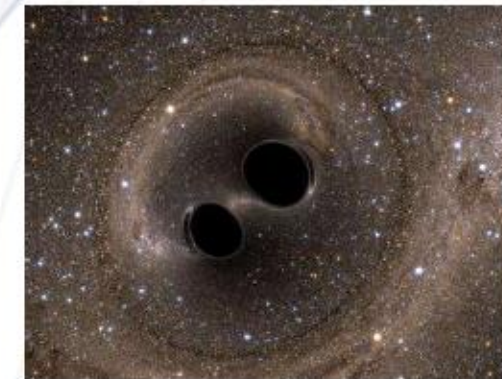
Bioinformatics/
Emergent Diseases



Critical Stress
in Subsurface
Energy Dynamics

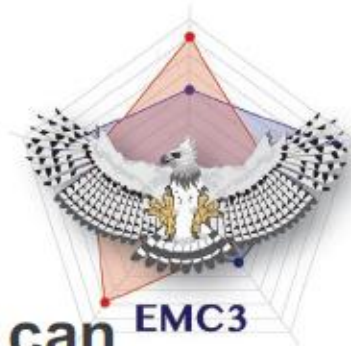


Constellation of CubeSats,
Carrying Ultra-Compact
Spectral Sensors

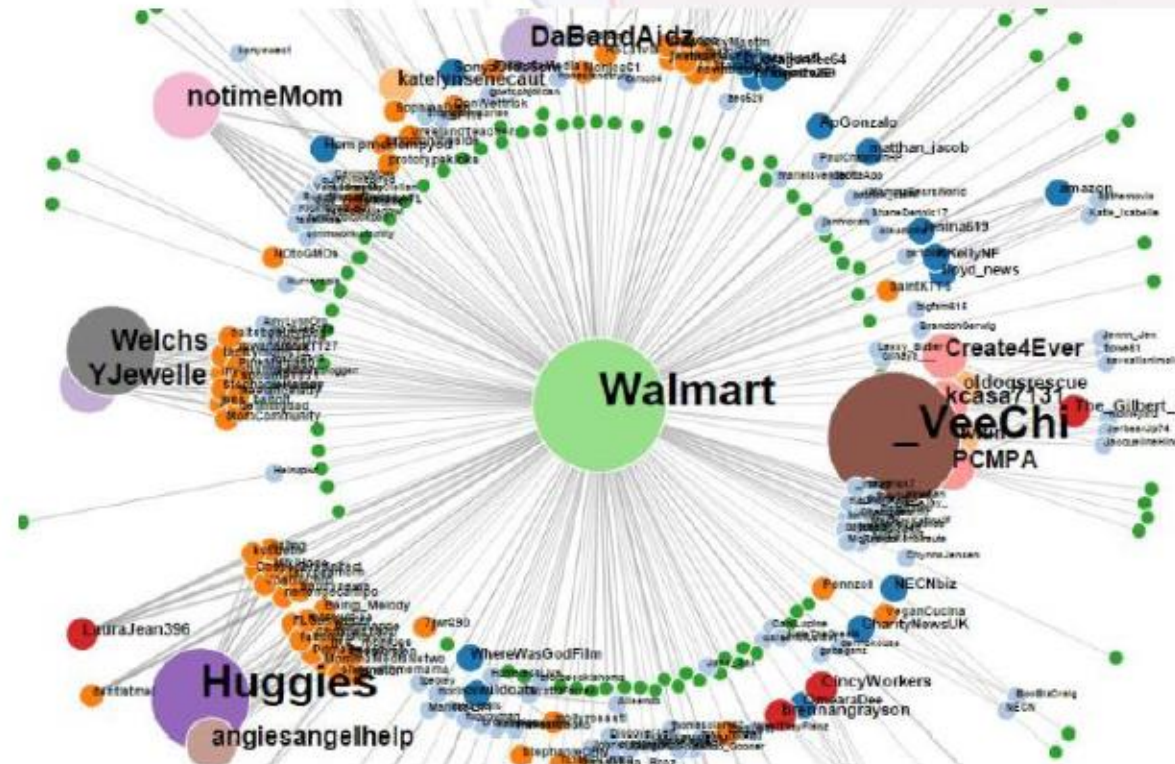


Gravitational Wave
Emissions from
Colliding Black Holes
(LIGO)

Graph Analytics and Event Simulation

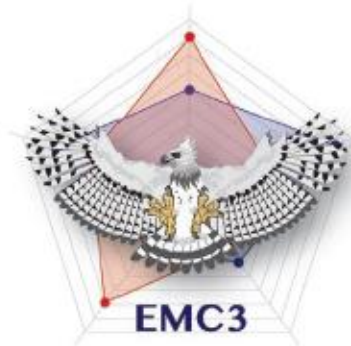


- Communications intensive, event/data driven, working set can exceed main and often have short words, mostly read. (GUPS)
 - Worst case is every lookup is not cached, in some random memory location somewhere in the cluster
 - Non-cached latency is memory latency ~40-100 cycles
 - Worse is across a low latency network ~1000 cycles (non-deterministic)



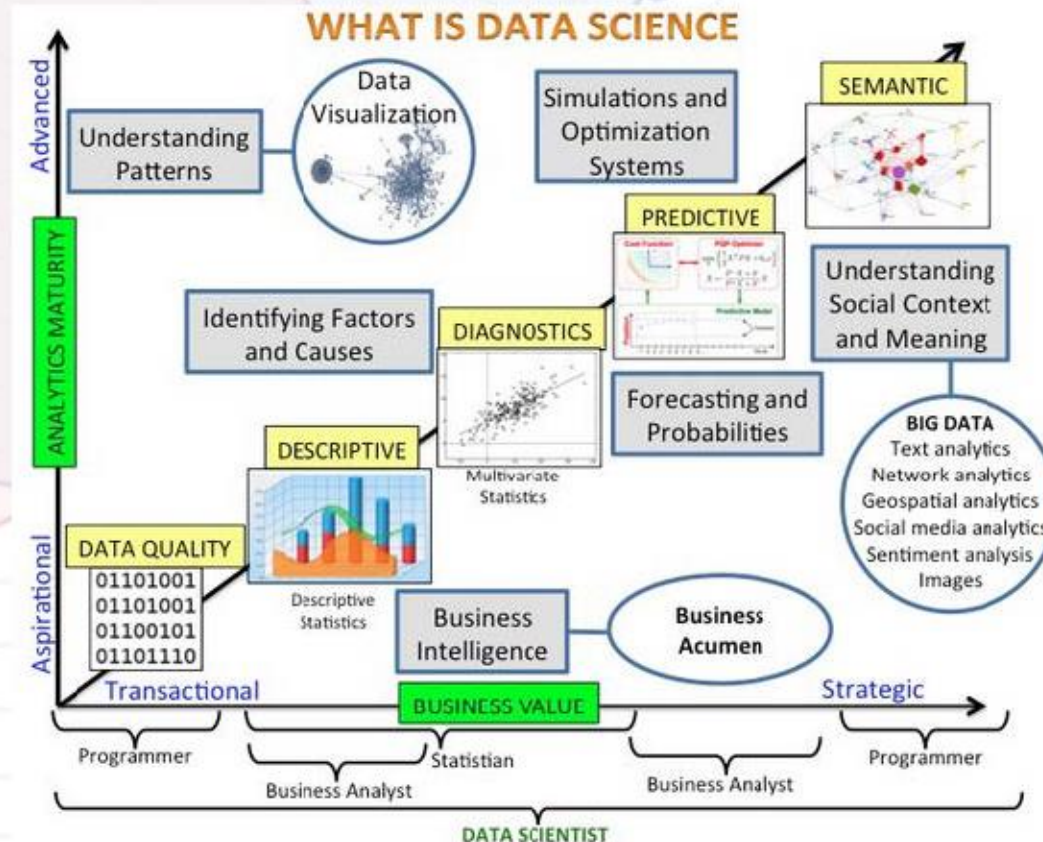
Follow relationships,
needles in haystacks, etc.

Many forms of Data Science: Analytics, Streaming Analytics, and some forms of Machine Learning/Deep Learning/AI

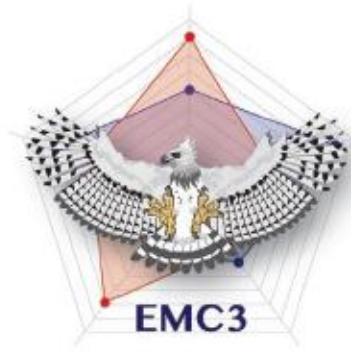


- Often IO Intensive, Data Parallel usually works very well, often with small word sizes, mostly read
 - Usually parallelizes well
 - IO and Network Bandwidth are sometimes limiters
 - We are working in these areas, so HW has to address issues.

Machine Learning basics

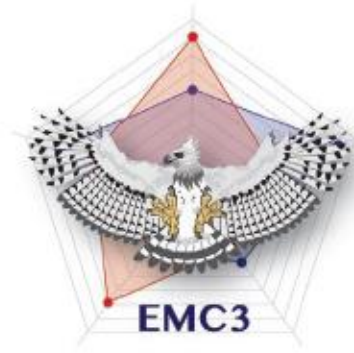


Why does LANL care about UCX?



- LANL is actively engaged in developing “in-network computing”.
- All of the previously mentioned apps will be able to take advantage of UCX
- UCX offers a continuum of deployment spaces.
 - NIC/HCA
 - Standard applications
 - OpenSHMEM
 - MPI
 - PGAS
- UCX offers portability across a very diverse portfolio of systems
- UCX is easily adapted to new and development HW
- We are currently porting some of the LANL mini-apps to UCX
- LANL is supporting UCX for external developers
- LANL is working with O&G customers wrt UCX
- LANL is part of UCF as well as OpenUCX
- LANL is fully supportive of an Open SmartNIC API

Contributors



- Bill Archer
- Jerry Brock
- AnnMarie Cutler
- Gary Grider
- Paul Henning
- Wendy Poole

ASC PM
Dep. ASC PM
Graphics
All Things I/O
Algorithms
Analytics





MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

Enhancing MPI Communication using Accelerated Verbs and Tag Matching: The MVAPICH Approach

Talk at UCX BoF (ISC '19)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

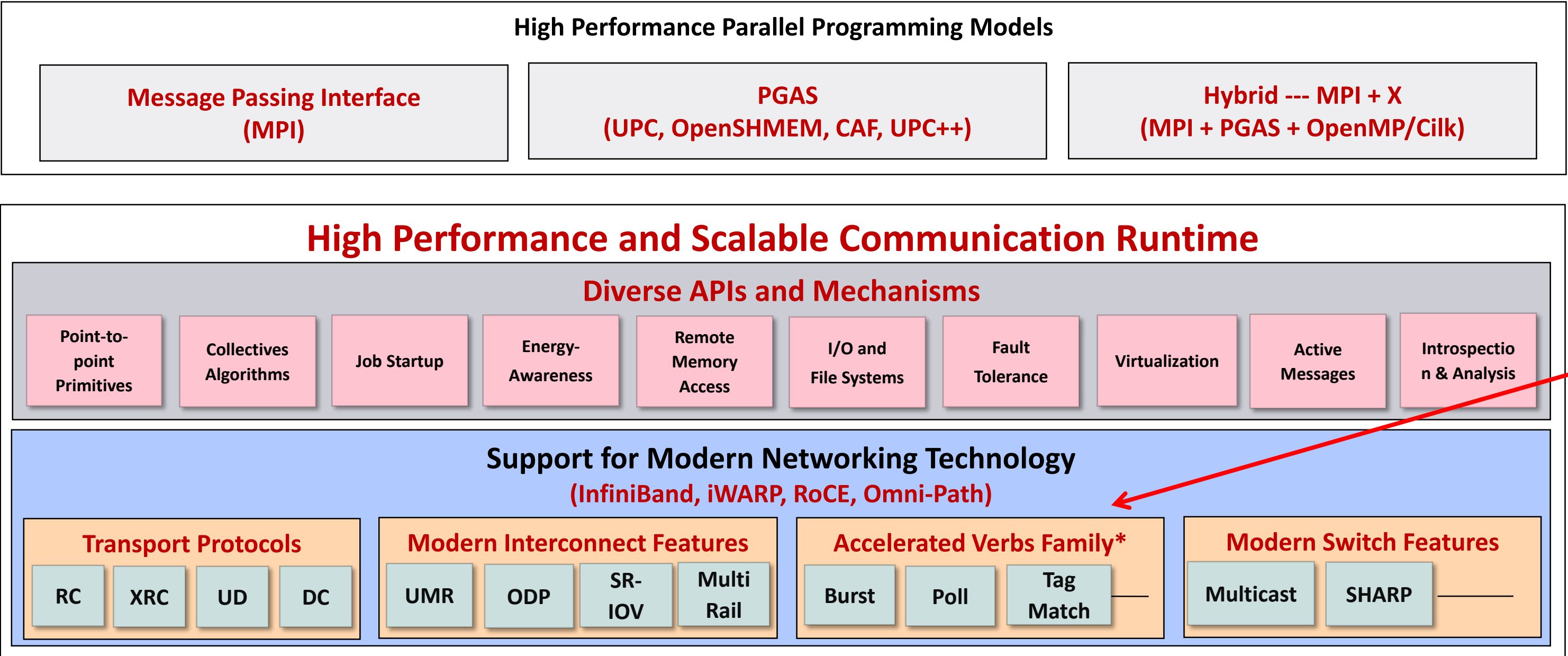
E-mail: panda@cse.ohio-state.edu

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

Introduction, Motivation, and Challenge

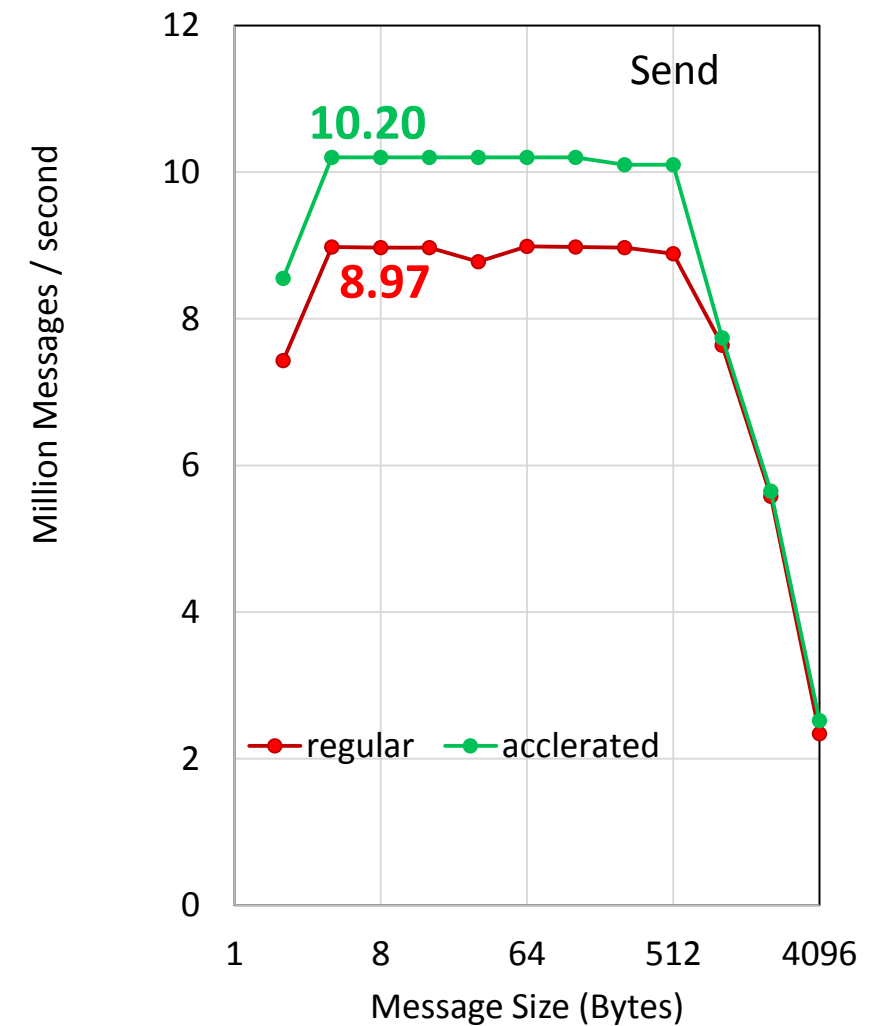
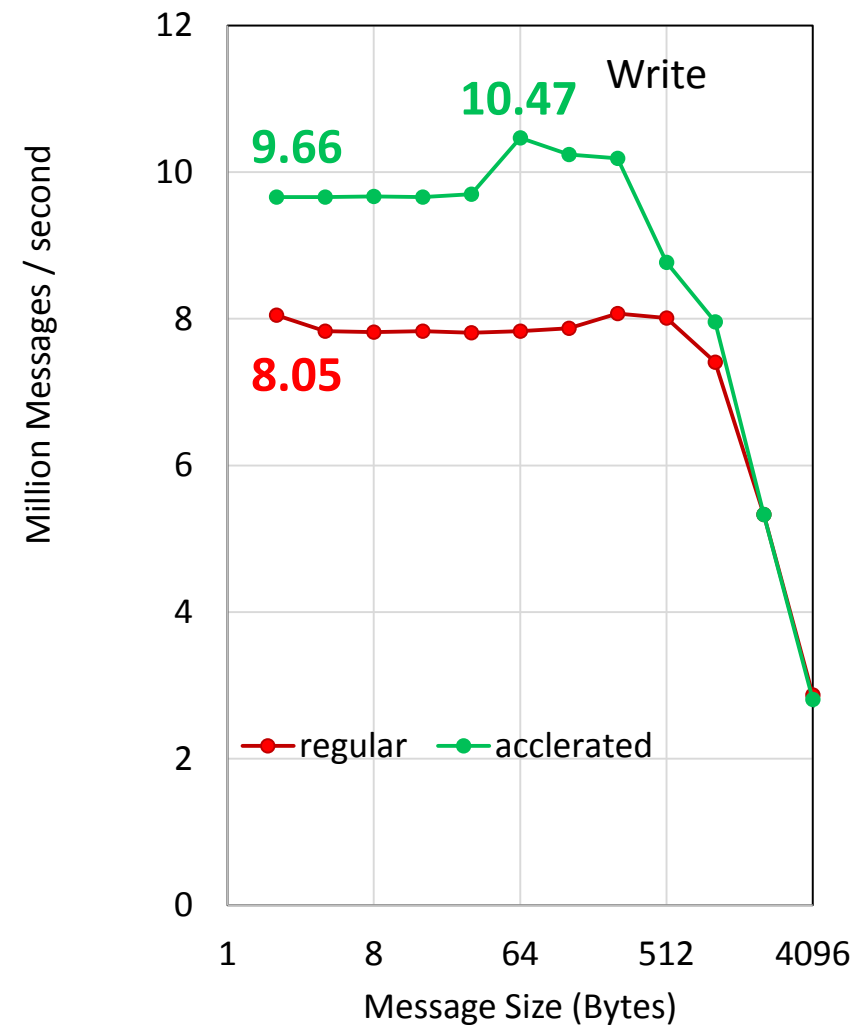
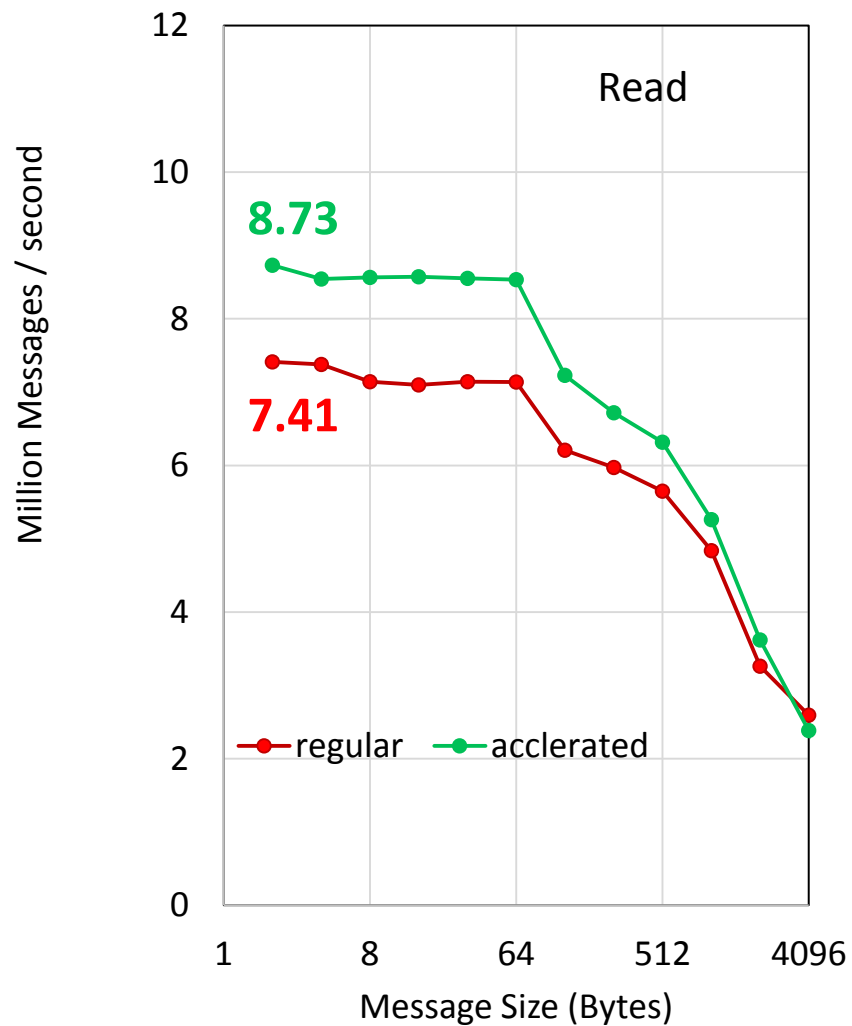
- HPC applications require high-performance, low overhead data paths that provide
 - Low latency
 - High bandwidth
 - High message rate
- Hardware Offloaded Tag Matching
- Different families of accelerated verbs available
 - Burst family
 - Accumulates packets to be sent into bursts of single SGE packets
 - Poll family
 - Optimizes send completion counts
 - Receive completions for which only the length is of interest
 - Completions that contain the payload in the CQE
- Can we integrate accelerated verbs and tag matching support in UCX into existing HPC middleware to extract peak performance and overlap?

The MVAPICH Approach



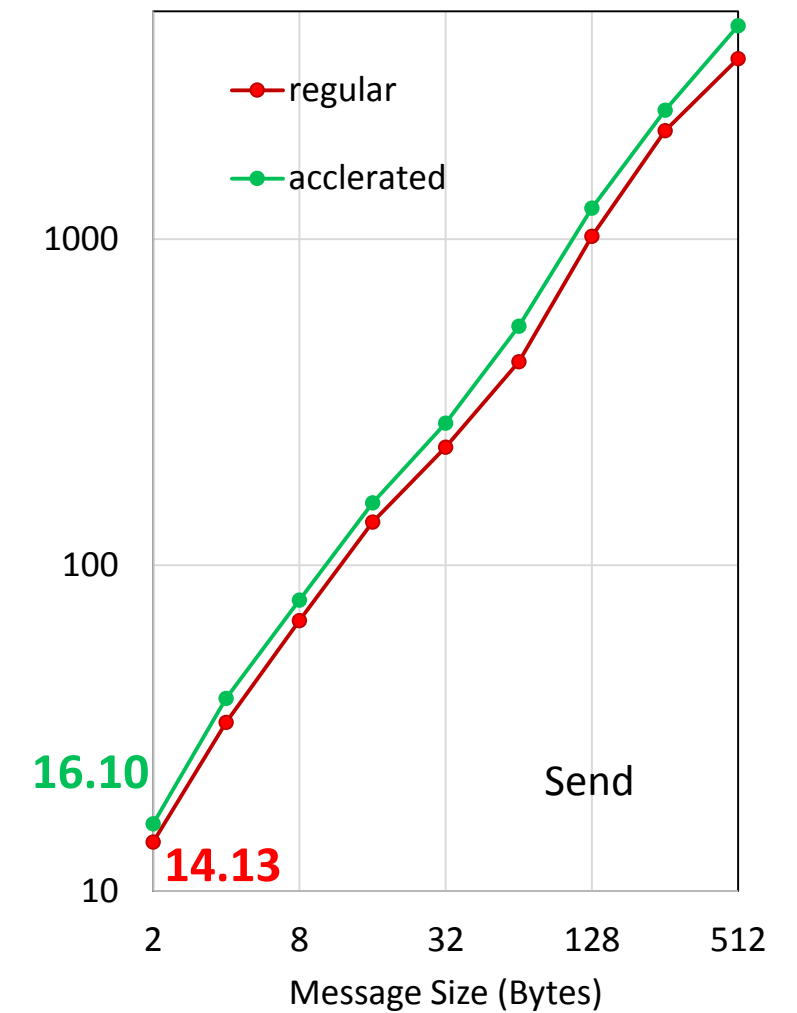
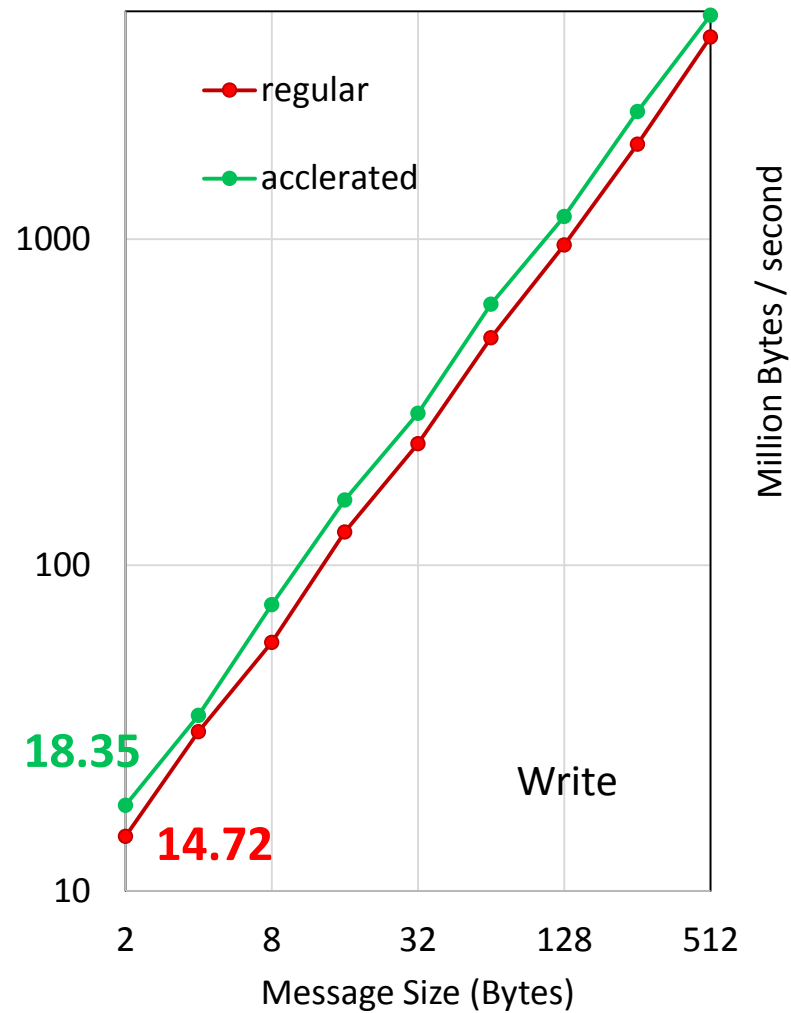
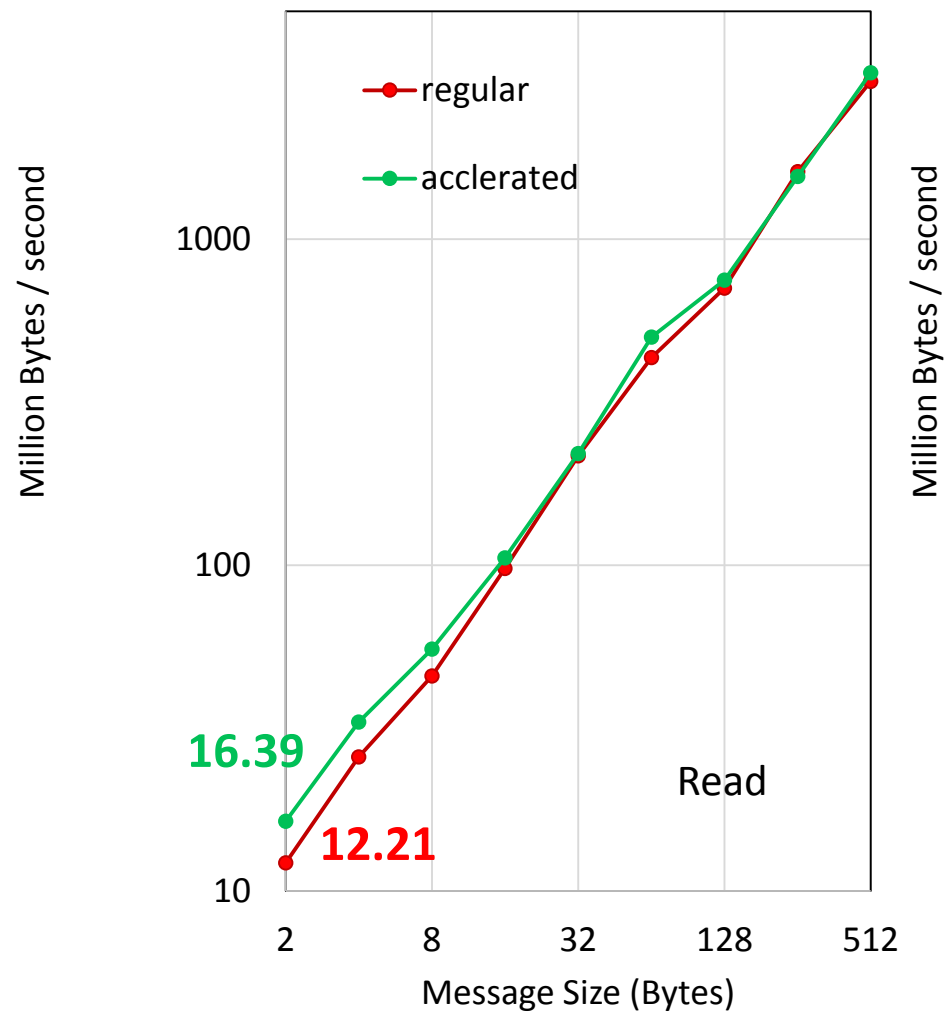
* Upcoming

Verbs-level Performance: Message Rate



ConnectX-5 EDR (100 Gbps), Intel Broadwell E5-2680 @ 2.4 GHz
MOFED 4.2-1, RHEL-7 3.10.0-693.17.1.el7.x86_64

Verbs-level Performance: Bandwidth

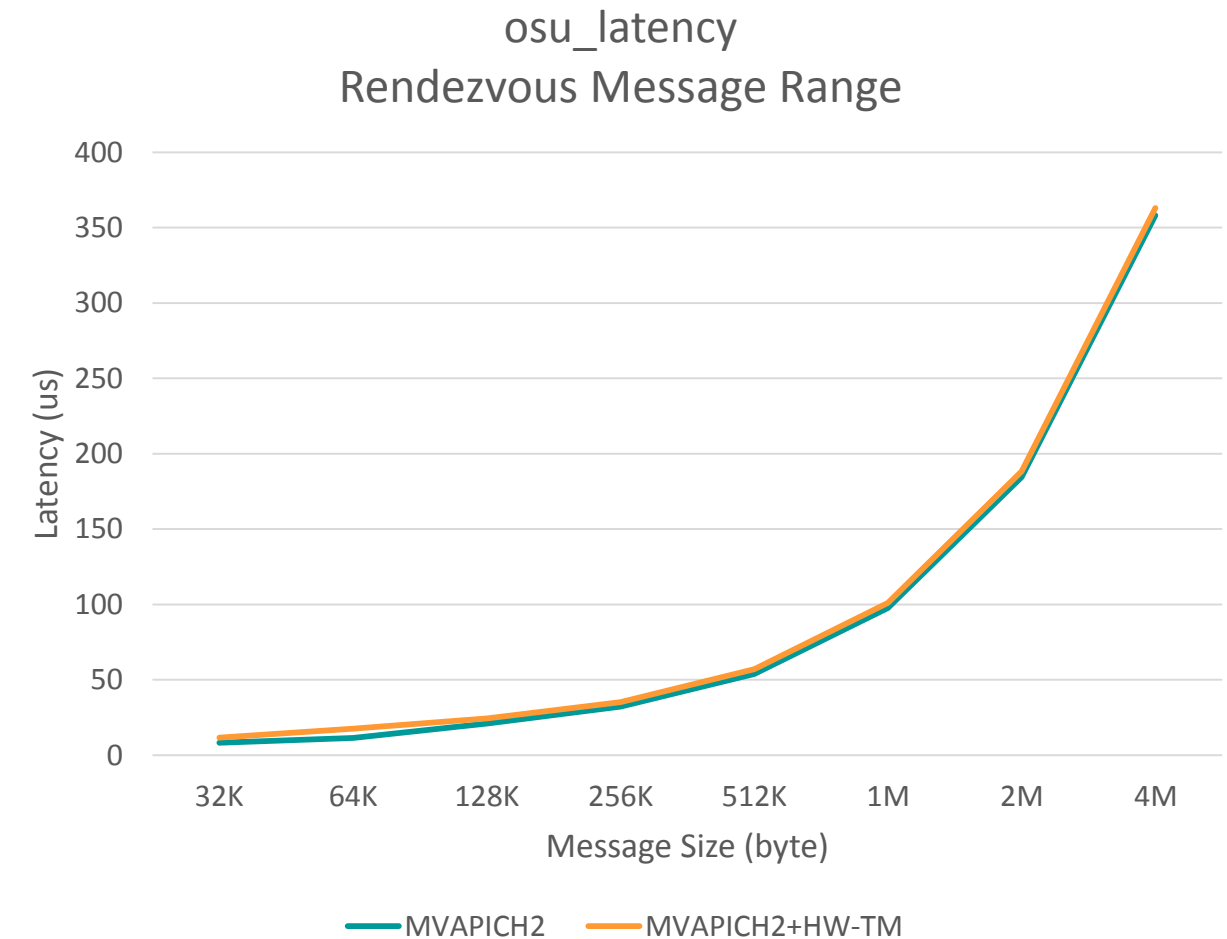
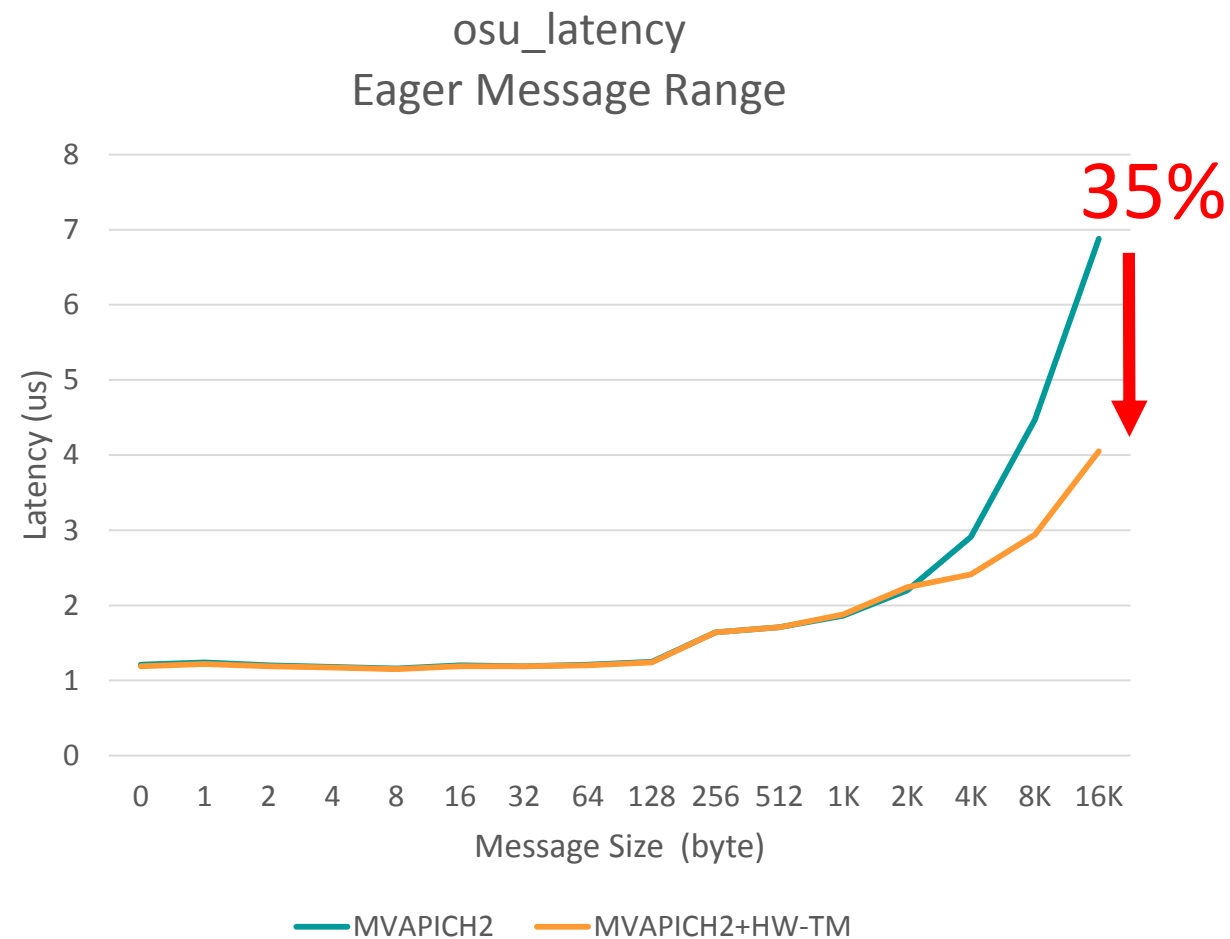


ConnectX-5 EDR (100 Gbps), Intel Broadwell E5-2680 @ 2.4 GHz
MOFED 4.2-1, RHEL-7 3.10.0-693.17.1.el7.x86_64

Hardware Tag Matching Support

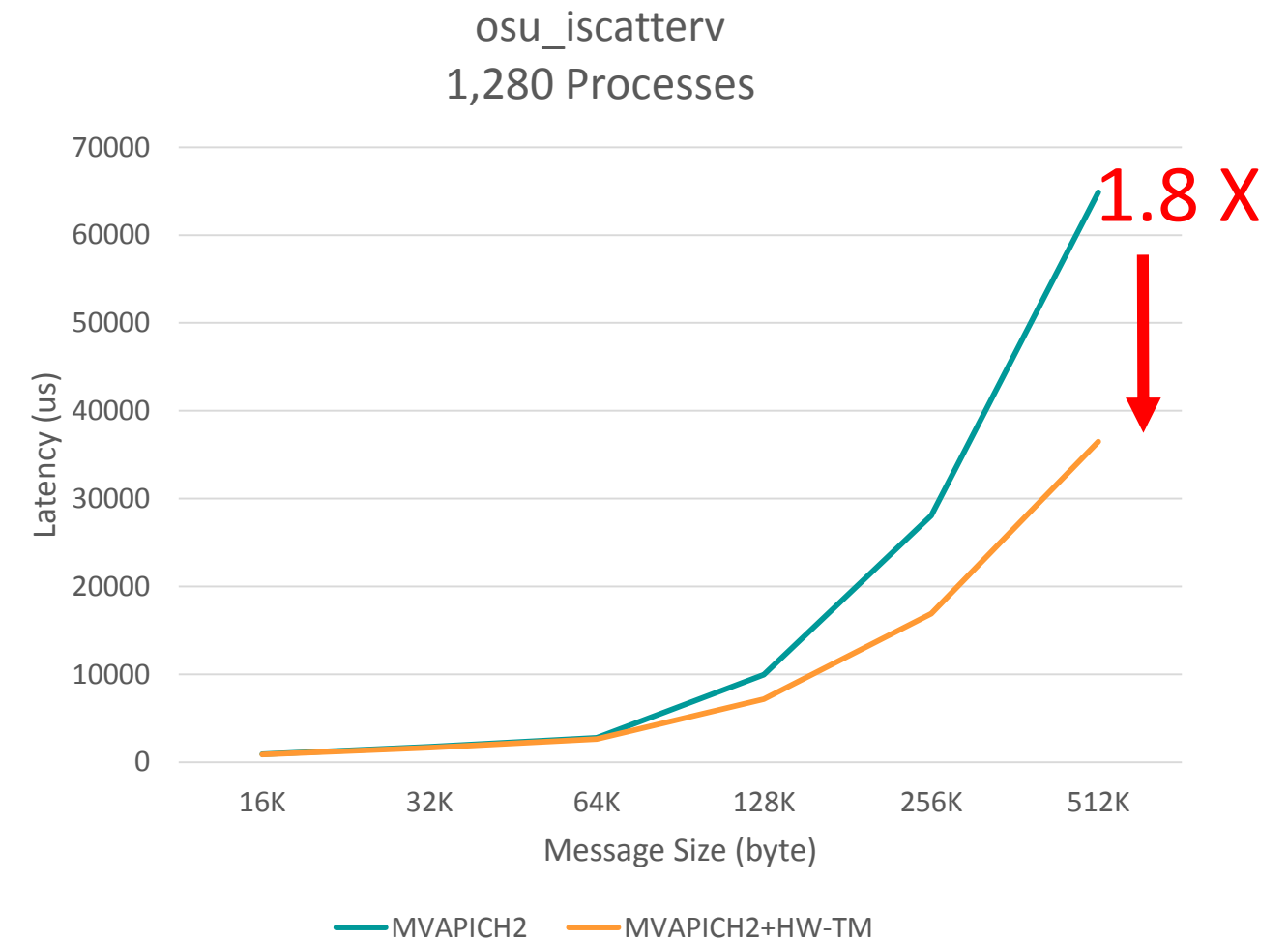
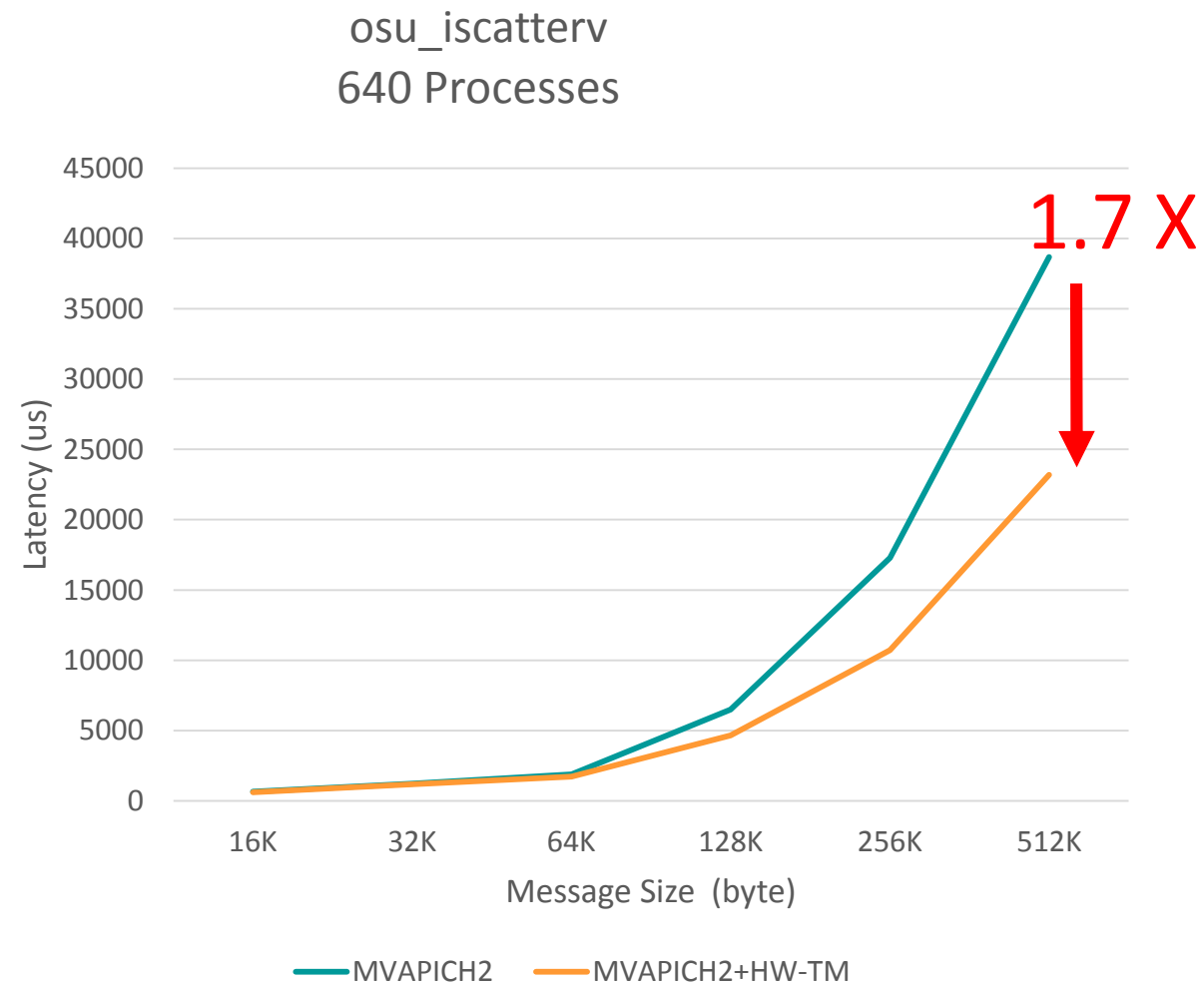
- Offloads the processing of point-to-point MPI messages from the host processor to HCA
- Enables zero copy of MPI message transfers
 - Messages are written directly to the user's buffer without extra buffering and copies
- Provides rendezvous progress offload to HCA
 - Increases the overlap of communication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

Impact of Zero Copy MPI Message Passing using HW Tag Matching



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

Impact of Rendezvous Offload using HW Tag Matching



The increased overlap can lead to 1.8X performance improvement in total latency of osu_iscatterv

Future Plans

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

UCX CUDA ROADMAP UPDATE

June, 2019



LEVERAGING CUDA & GPU DIRECT

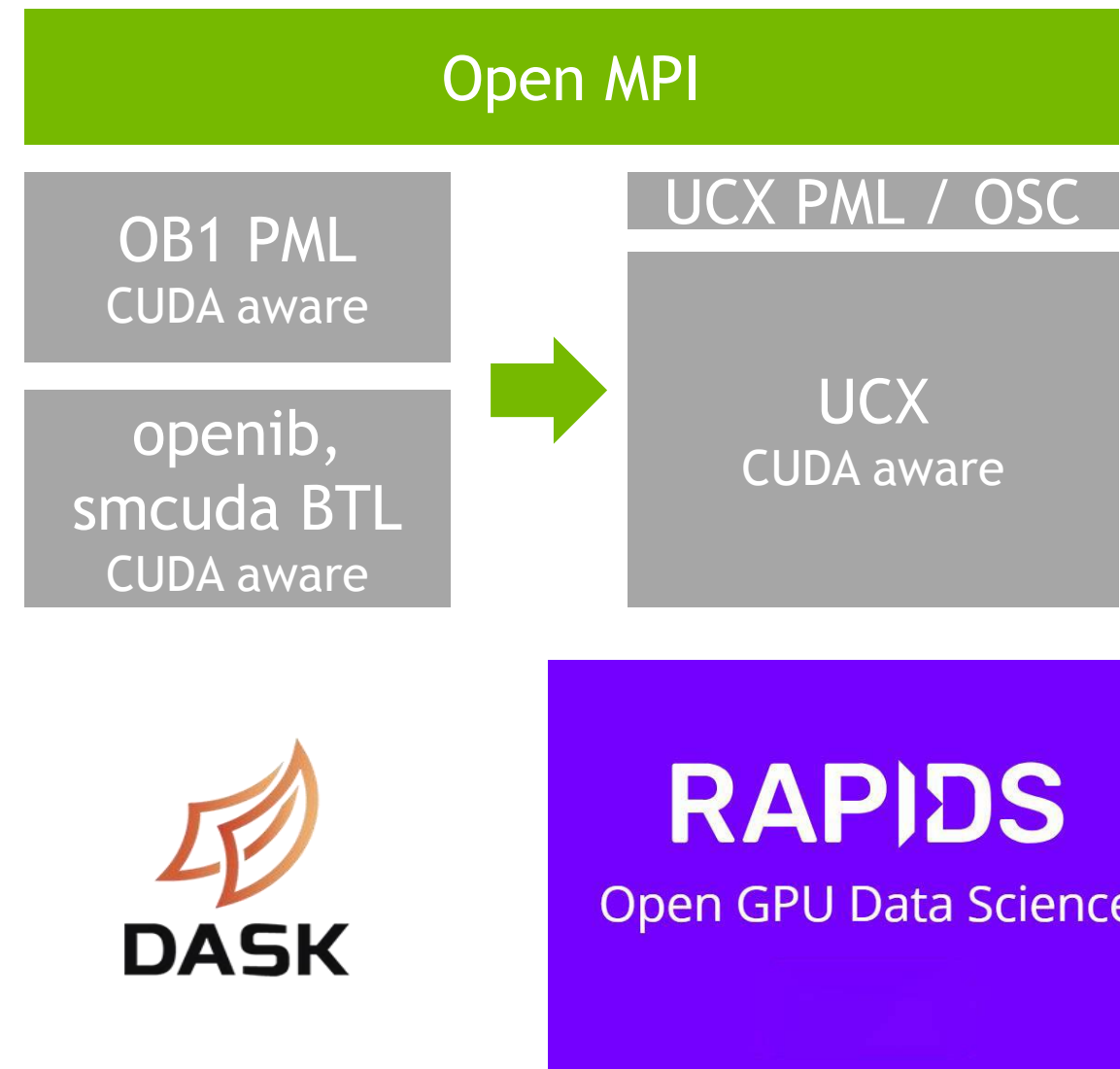
CUDA-awareness in UCX

Mellanox MPI effort is PML UCX

MPICH and OpenMPI use UCX

Move CUDA-related features to UCX

GPU-accelerated Data Science projects
under RAPIDS starting to leverage UCX
directly



CURRENT SUPPORT

Code contributions from Mellanox and NVIDIA

GPUDirectRDMA + GDRCopy for inter-node transfers

CUDA-IPC + GDRCopy for intra-node transfers

Pointer cache through interception mechanisms; CUDA-IPC mapping cache

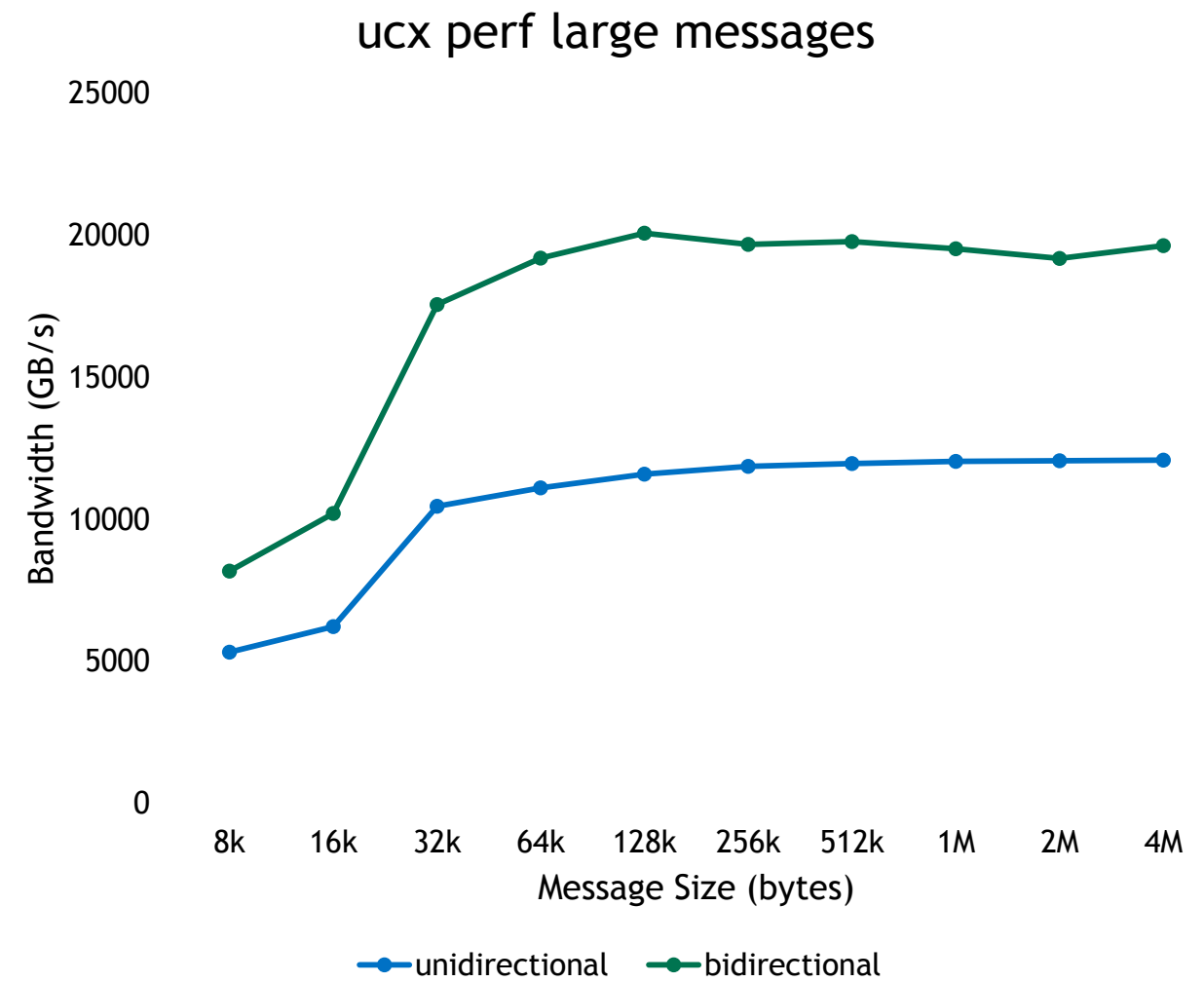
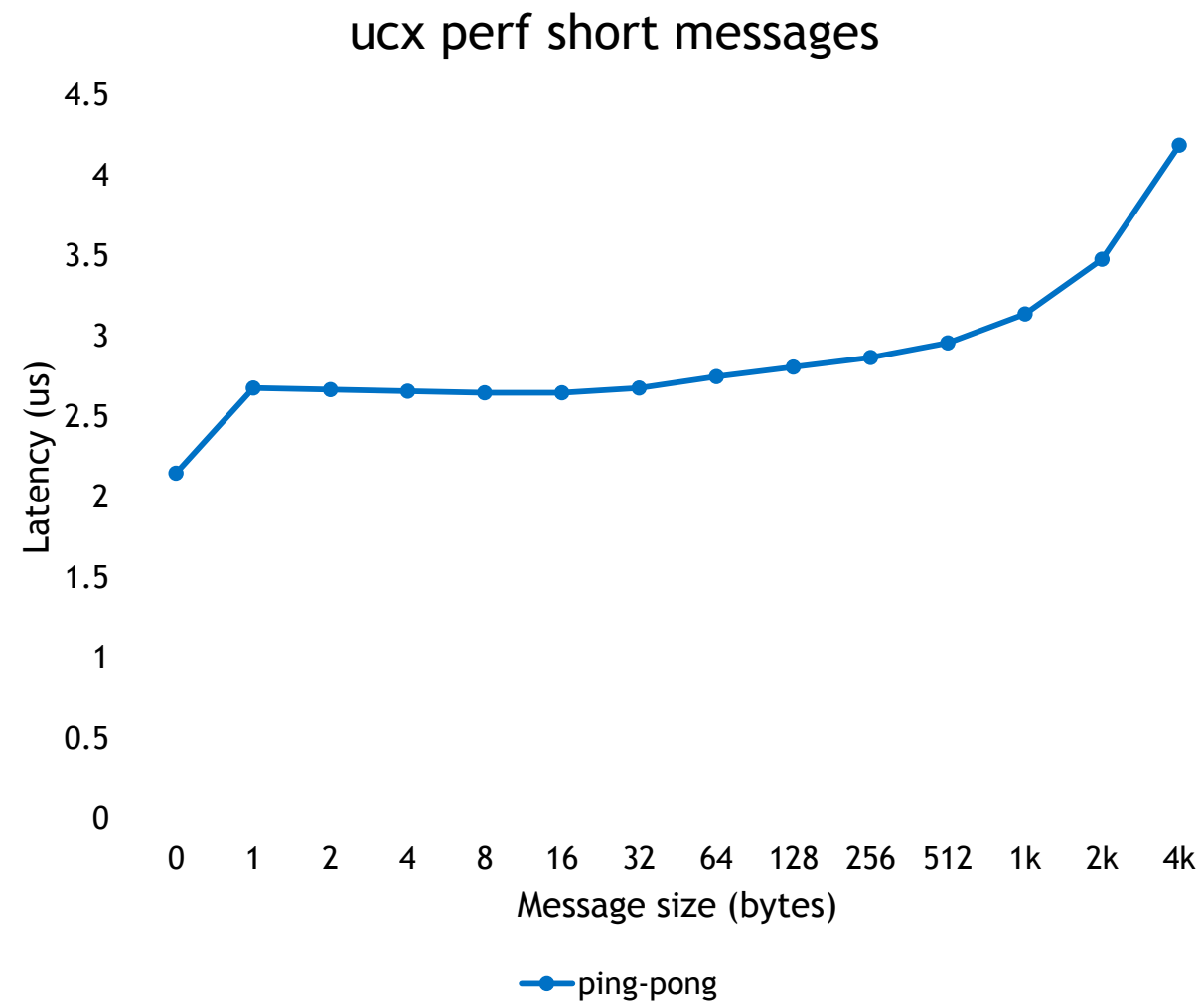
Managed memory support

Automatic HCA selection based on GPU-affinity (UCX 1.5 release)

Python-bindings for UCX (ucx-py, <https://github.com/rapidsai/ucx-py>)

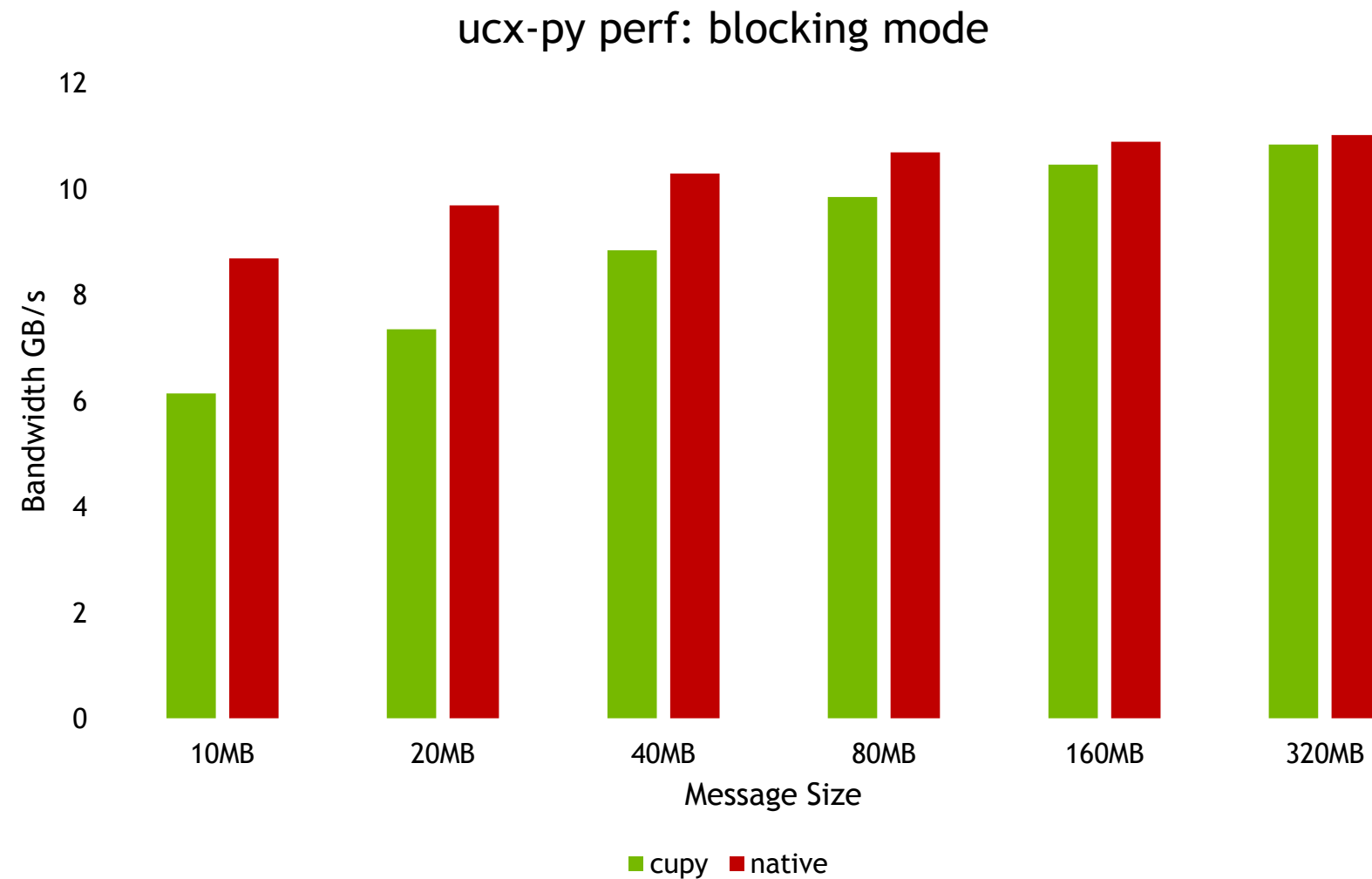
UCX-CUDA PERFORMANCE

Results with 2 DGX-1 nodes: approach peak with big enough buffers



UCX-PY PERFORMANCE

Results with 2 DGX-1 nodes: approach peak with big enough buffers



UPCOMING TARGETS

Short term:

- 3-stage pipeline optimizations (for imbalanced GPU-HCA configurations)

- Pipelining over NVLINK path (for managed memory; memory footprint)

- Automatic HCA selection based on GPU-affinity (general availability)

Long term:

- Persistent request support: memoize xfer info for repeated use

- Non-contig Datatypes optimizations

- One-sided UCX-CUDA; stream-based UCX operations

Join the UCX Community

Save the date !

- UCX F2F meeting is planed on the week of December 9
- 3 days meeting
- Austin, TX





Unified Communication - X Framework

WEB:

www.openucx.org

<https://github.com/openucx/ucx>

Mailing List:

<https://elist.ornl.gov/mailman/listinfo/ucx-group>

ucx-group@elist.ornl.gov

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.