



# Smarter Networks, Smarter MPI

Improving MPI User Experience & Cluster Performance

**Virtual HPC User Forum Special Event**

February 3, 2022

1:00pm to 2:30pm ET



THE OHIO STATE UNIVERSITY



Queen's  
UNIVERSITY

rockport.

# HPC User Forum



- Mission: promote the health of the global HPC industry and address issues of common concern to users
- Steering Committee: voluntary, multinational steering committee of 28 leading HPC experts from government, academia and industry
- Events
  - Two forums annually in the United States
  - Two forums annually internationally
  - Special Events throughout the year

# Upcoming Event



- Virtual Spring 2022 HPC User Forum
  - North America stream:
    - Mar 28 – Mar 30
    - 10:00AM – 3:00PM CDT (Central Daylight Time)
  - EMEA / APAC stream:
    - Mar 29 – Mar 31
    - 10:00 – 15:00PM BST (British Summer Time)
- Agenda and registration forthcoming

# Presenters & Panelists



**Mark Nossokoff**  
Senior Analyst  
Hyperion Research



**Dr. John McCalpin**  
Research Scientist  
Texas Advanced  
Computing Center



**Dr. Dhabaleswar K.  
(DK) Panda**  
Professor  
Ohio State University



**Dr. Ryan Grant**  
Assistant Professor  
Queen's University



**Matthew Williams**  
CTO  
Rockport Networks



# Agenda



Speaker	Organization	Topic	Start	Duration
Mark Nossokoff	Hyperion Research	Introduction	1:00	5 min
Dr. John McCalpin	TACC	Defining the MPI/Resource Challenge	1:05	20 min
Dr. DK Panda	OSU	MVAPICH Perspective	1:25	10 min
Dr. Ryan Grant	Queen's University	Broader MPI Perspective	1:35	10 min
Matt Williams	Rockport	Advances in interconnect/MPI integration	1:45	15 min
All		Q&A	2:00	30 min





HYPERION RESEARCH

# Smarter Networks, Smarter MPI

## Hyperion Research Perspective

February 3, 2022

Mark Nossokoff, Research Director

[www.HyperionResearch.com](http://www.HyperionResearch.com)  
[www.hpcuserforum.com](http://www.hpcuserforum.com)

# Hyperion Research

*www.hyperionresearch.com*



HYPERION RESEARCH

[Home](#)

[Services](#) ▾

[Team](#)

[Sample Projects](#) ▾

[Events](#)

[Careers](#)

[Contact](#)



[LOGIN](#)

**Hyperion Research helps organizations make effective decisions and seize growth opportunities by providing research and recommendations in both high performance computing and emerging technology areas.**

[Sample Projects](#) ▾

[Hyperion In The News](#)

[HPC Market Update During SC21](#)

[Purchase Documents](#)

# Parallel Programming Models Used

*Over 50% of sites utilized MPI*

	Overall Percent	Industry Percent	Government Percent	Academia Percent
C/C++ (all types)	79.4%	68.3%	91.3%	97.2%
Python	73.8%	80.5%	56.5%	69.4%
CUDA	52.5%	37.8%	73.9%	72.2%
<b>MPI</b>	<b>51.8%</b>	<b>30.5%</b>	<b>73.9%</b>	<b>86.1%</b>
OpenMP	48.2%	31.7%	60.9%	77.8%
Fortran (all types)	47.5%	28.0%	69.6%	77.8%
R	41.8%	41.5%	26.1%	52.8%
MATLAB	39.0%	31.7%	52.2%	47.2%
Java	29.8%	31.7%	34.8%	22.2%
OpenCL	24.1%	17.1%	26.1%	38.9%
Mathematica	18.4%	11.0%	26.1%	30.6%
Pthreads	17.0%	12.2%	26.1%	22.2%
Scala	14.2%	20.7%	0.0%	8.3%
Ruby	12.8%	15.9%	4.3%	11.1%
Julia	10.6%	8.5%	8.7%	16.7%
SHMEM	10.6%	4.9%	26.1%	13.9%
Coarray Fortran	9.9%	4.9%	21.7%	13.9%
PGAS	9.9%	4.9%	21.7%	13.9%
PVM	2.8%	0.0%	8.7%	5.6%
Cilk	2.1%	1.2%	4.3%	2.8%
Other	5.7%	3.7%	0.0%	13.9%
n = 141; 82; 23; 36 (respectively)				
Source: Hyperion Research, 2021				

- **141 global sites**
- **2,006 HPC systems**
- **Multiple responses allowed**

# Barriers to Expanding On-Prem HPC

*Sizable percentages related skillset and operational issues*

- **32% felt that lack of staff or skills were keeping them from buying more HPC**
  - 40% in Industry
  - 21.7% in Government
  - 19.4% in Academia
- **28% were limited by the difficulties in scaling their workloads**
  - 32% in Industry
  - 22% in Government
  - 25% in Academia
- **24% were limited by programming hurdles with hybrid environments**
  - 18% in Industry
  - 61% in Government
  - 14% in Academia
- **16% were limited by ease-of-use issues**
  - 21% in Industry
  - 9% in Government
  - 11% in Academia

**For more information...**

**Questions or comments are welcome.**



**[mnossokoff@hyperionres.com](mailto:mnossokoff@hyperionres.com)**





TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU



TEXAS

The University of Texas at Austin

# Some Challenges with MPI

Everything is harder than it looks....

PRESENTED BY:

John D. McCalpin, PhD

Research Scientist

Texas Advanced Computing Center

[mccalpin@tacc.utexas.edu](mailto:mccalpin@tacc.utexas.edu)

# Economics has driven us to one primary model

- Loosely-coupled clusters of small nodes
  - 1-4 multicore processors
  - Optional “accelerators”
  - IO-based interconnect networks
- Performance improvements are obtained primarily by parallelizing over more hardware
  - Hardware has increasing layers of hierarchy
- Software is increasingly complex
  - MPI-1.x specifications: ~240 pages
  - MPI-2.x specifications: ~600 pages
  - MPI-3.x specifications: ~850 pages
  - MPI-4.0 specification: 1139 pages

# Nodes have become immensely more complex

- Multiple NUMA domains
  - Sometimes aligned with sockets, sometimes not (in either direction)
  - Local vs Remote BW & latency properties vary significantly across products
  - Interconnect often has preference for local socket
- Multiple shared caches
  - sometimes aligned with NUMA domains, sometimes subsets
- Multiple levels of cache
  - L1, L2 usually private
  - L3 usually shared, but many possible configurations
  - DRAM caches in some systems
    - HBM as cache for DDR (Intel Knights Landing & possible future products)
    - DDR as cache for Persistent Memory (Intel Cascade Lake & later)

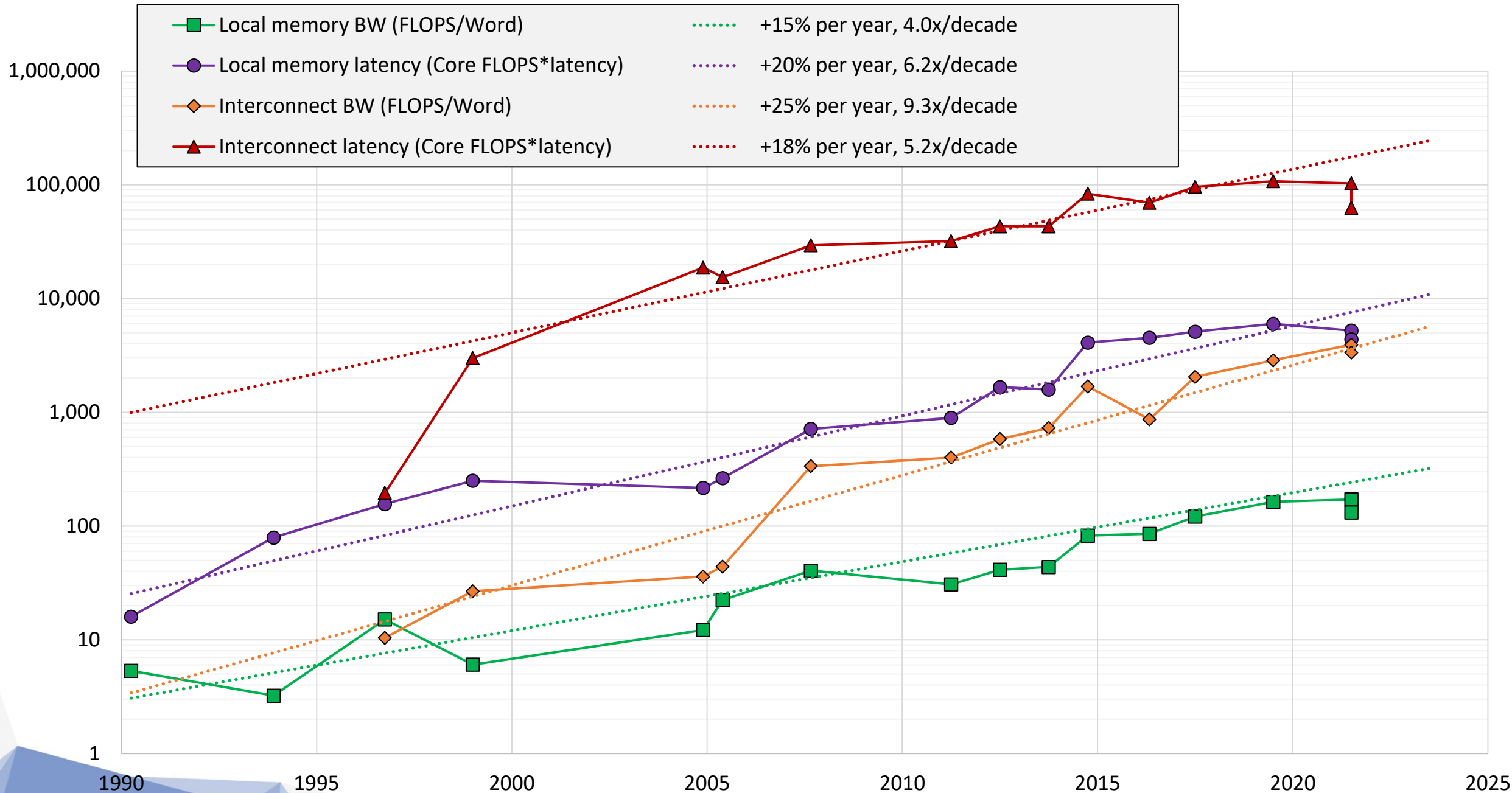
# Bandwidth Increases, but Latency constant

This increases the required concurrency according to Little's Law

$$\textit{Concurrency} = \textit{Latency} \times \textit{Bandwidth}$$

Example:

- Xeon Platinum 8380
  - 3<sup>rd</sup> gen Intel Scalable Processor (aka Ice Lake Xeon)
- Local DDR4 memory latency: 85 ns
- Peak DDR4 bandwidth (1 socket): 204.8 GB/s
- Required Concurrency to tolerate local memory latency:
  - 17408 bytes = 272 cache lines
  - This corresponds to 6.8 cache lines per core on all 40 cores

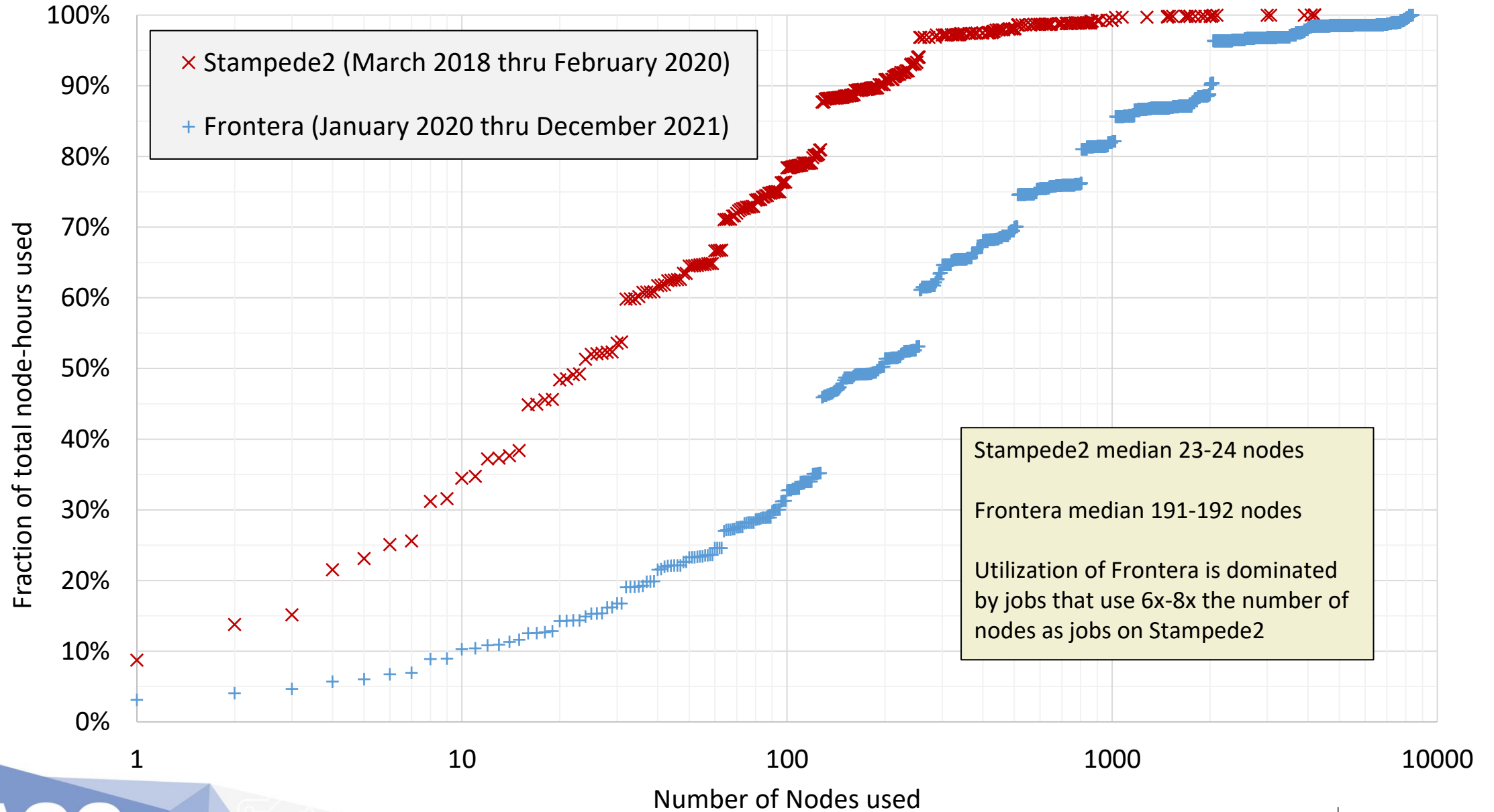


**Cost of data motion in terms of 64-bit Floating Point arithmetic over time**

# Processor core performance is almost stagnant

- Dennard Scaling was the observation that frequency increases and power density is fixed as gate size decreases
  - Allowed scaling of frequencies from MHz to GHz from the 1980's to early 2000's
- Processor frequencies have been roughly constant since ~2003 (?)
- Memory bandwidth per core has been roughly constant since the early 2000's
- Cache sizes per core are not systematically increasing
- Special features such as wider SIMD arithmetic have limited breadth of applicability (and reduce average frequency when used)
- So... the only way to get more performance is to parallelize over more hardware....

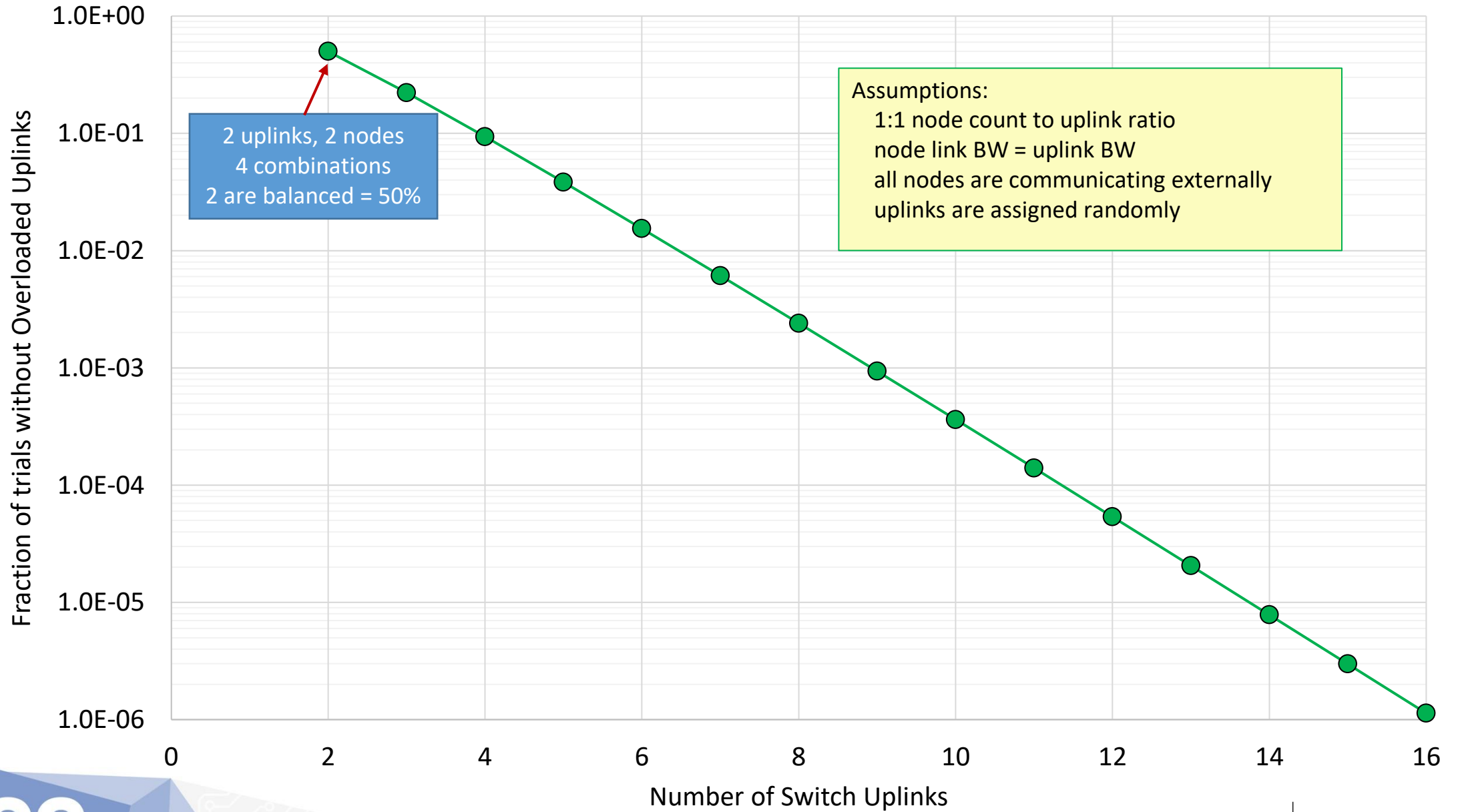
# TACC Frontera and Stampede2: Cumulative Utilization by Job Node Count



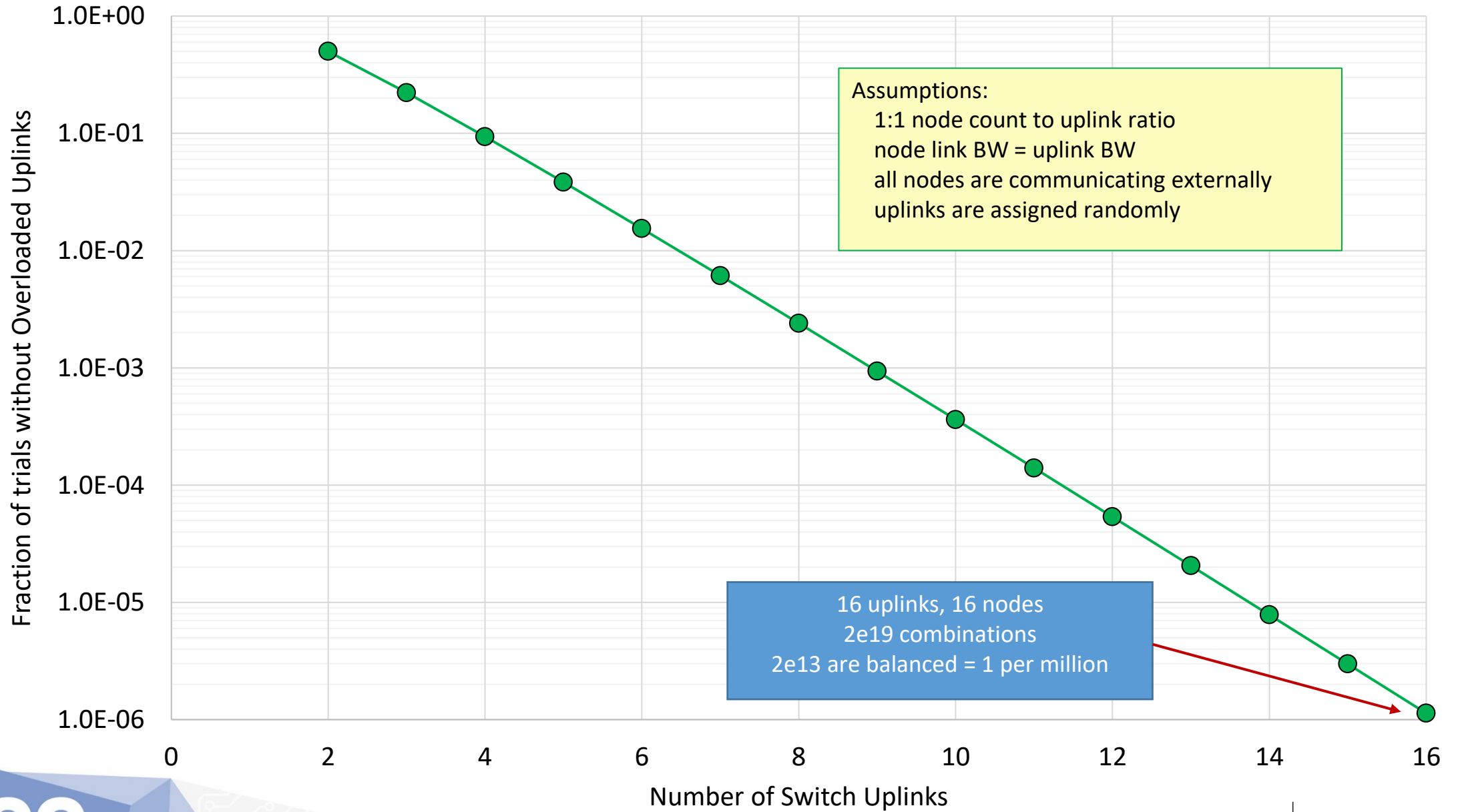
# Static Routing and Uplink overloading

- A “full-bisection bandwidth” network configuration has enough links to allow any  $\frac{1}{2}$  of the nodes to communicate to the other  $\frac{1}{2}$  of the nodes without overloading any links
- This means that it is *possible* to get full bandwidth – it does not mean that it is *likely* that it will occur
- For a production system like Frontera, the nodes allocated to a job are effectively random
  - Nodes are clustered into chassis switches when possible, but the sizes and chassis locations are effectively random
- The uplink used to send to a remote node is a static function of the target node – effectively a random value
- Randomly selected uplinks will very often conflict, causing overloading and reduction in per-node bandwidth

# Randomly-generated conflict-free routes are improbable at high radix



# Randomly-generated conflict-free routes are improbable at high radix



The TACC Frontera main compute cluster consists of 192 chassis, each with an InfiniBand HDR100/200 switch.

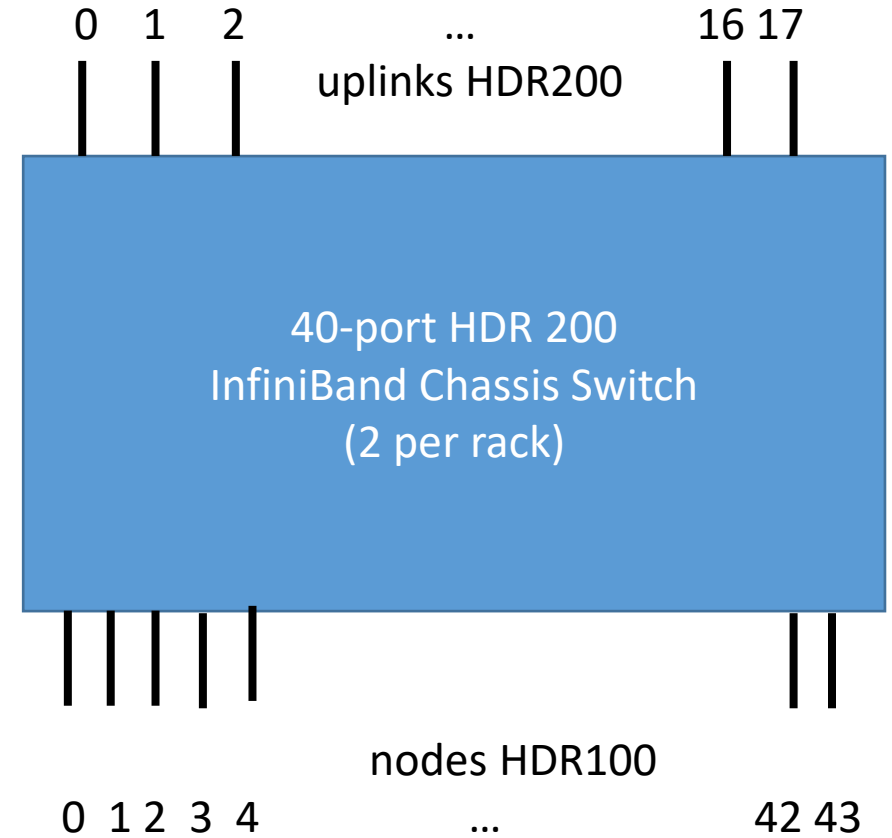
In each chassis:

- 44 nodes with HDR100 connections use up 22 of the HDR200 ports

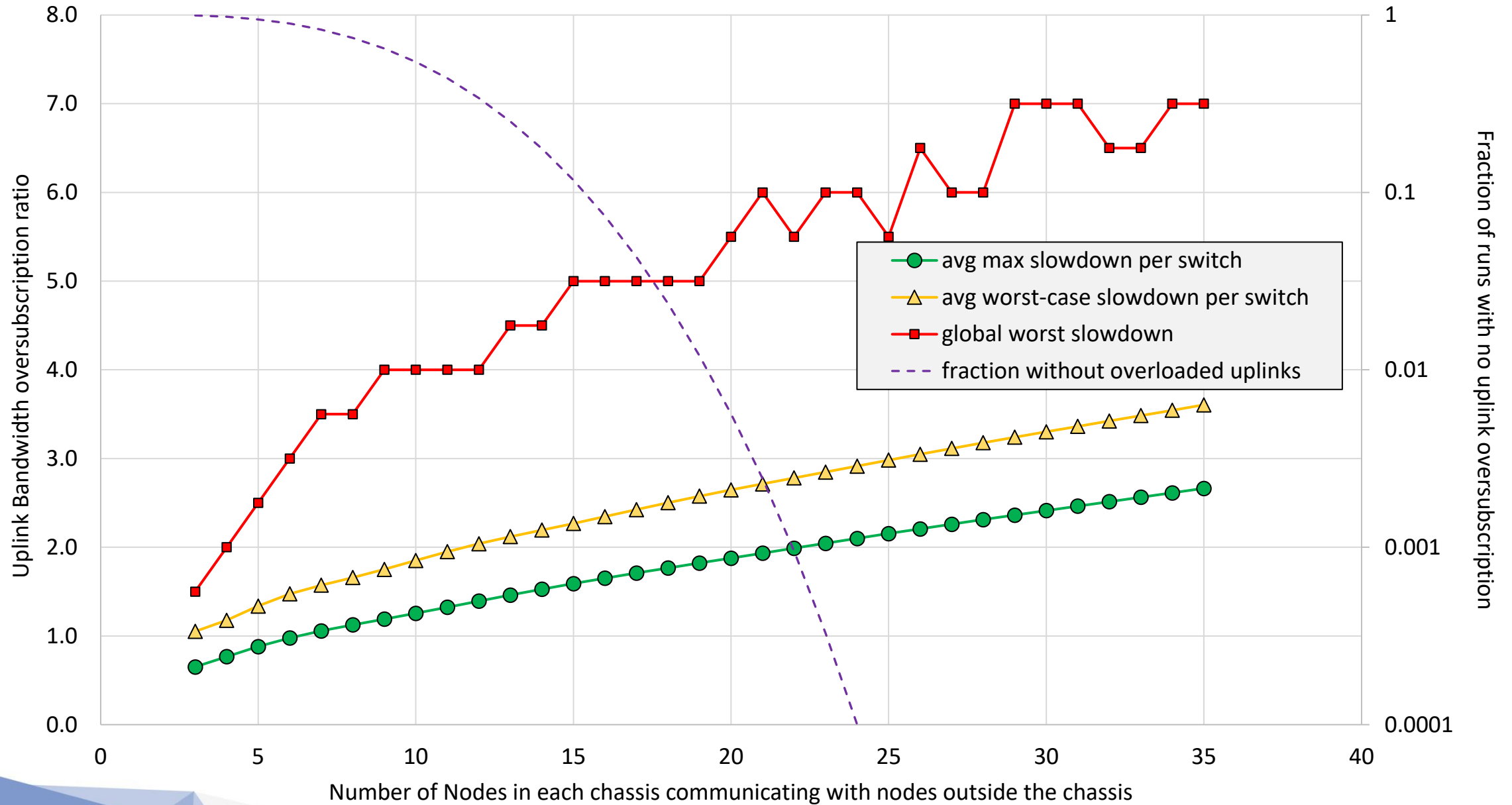
- 18 uplinks provide 3 HDR200 links to each of the 6 400-port central switches

Each uplink can handle the full bandwidth from any two nodes communicating in the same direction, so the “slowdown factor” is  $\frac{1}{2}$  the number of nodes simultaneously using a link.

NOTE that static routing conflicts do not require oversubscription – only massive *undersubscription* or adaptive routing will help.



Static Routing uplink overloads for 18 chassis on Frontera-like system



# How do these static routing conflicts hurt?

- Static routing conflicts are variable – depending on the interaction of the application’s communication pattern with the specific set of nodes assigned and the system’s current set of routing tables
- Suppose most runs have 2x slowdown on some links due to oversubscription, but some runs have 3x slowdowns
  - If the “ideal” execution should spend 20% of its time in communication, the most common runs will be 20% slow (the new “normal”), and some runs will be 40% slow.
  - Slowdowns that happen on most jobs reduce the center’s effectiveness and cause us to over-spend on the underperforming network hardware/
  - Unexpected additional 20% slowdowns can cause job timeouts and interfere with how much science can get done in a project.
  - The user’s preferred adaptation is to decrease the size of the jobs – fewer nodes, less communication time, less variability
  - This can also mean less science...

# Summary

- The MPI programming model is familiar, but it is being mapped onto increasingly complex systems
  - Some of the complexity is visible to the users, some is not
- Vendor secrecy often prevents users from seeing details
  - Even without secrecy, the number of interacting components is frightening
- Despite decades of development, most current InfiniBand and OmniPath networks are limited by static routing
  - Effectively impossible to use all the wires at once!
- Good News – there is still low-hanging fruit
  - Adaptive routing
  - Exploiting QoS to minimize contention between independent traffic
- Bad News – this continues to get harder as systems become larger, more heterogeneous, and more heavily shared

# Exploiting Smarter Networks: MVAPICH2 Perspectives

Virtual HPC User Forum (Feb. '22)

by

**Dhableswar K. (DK) Panda**

The Ohio State University

E-mail: [panda@cse.ohio-state.edu](mailto:panda@cse.ohio-state.edu)

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



Follow us on

<https://twitter.com/mvapich>

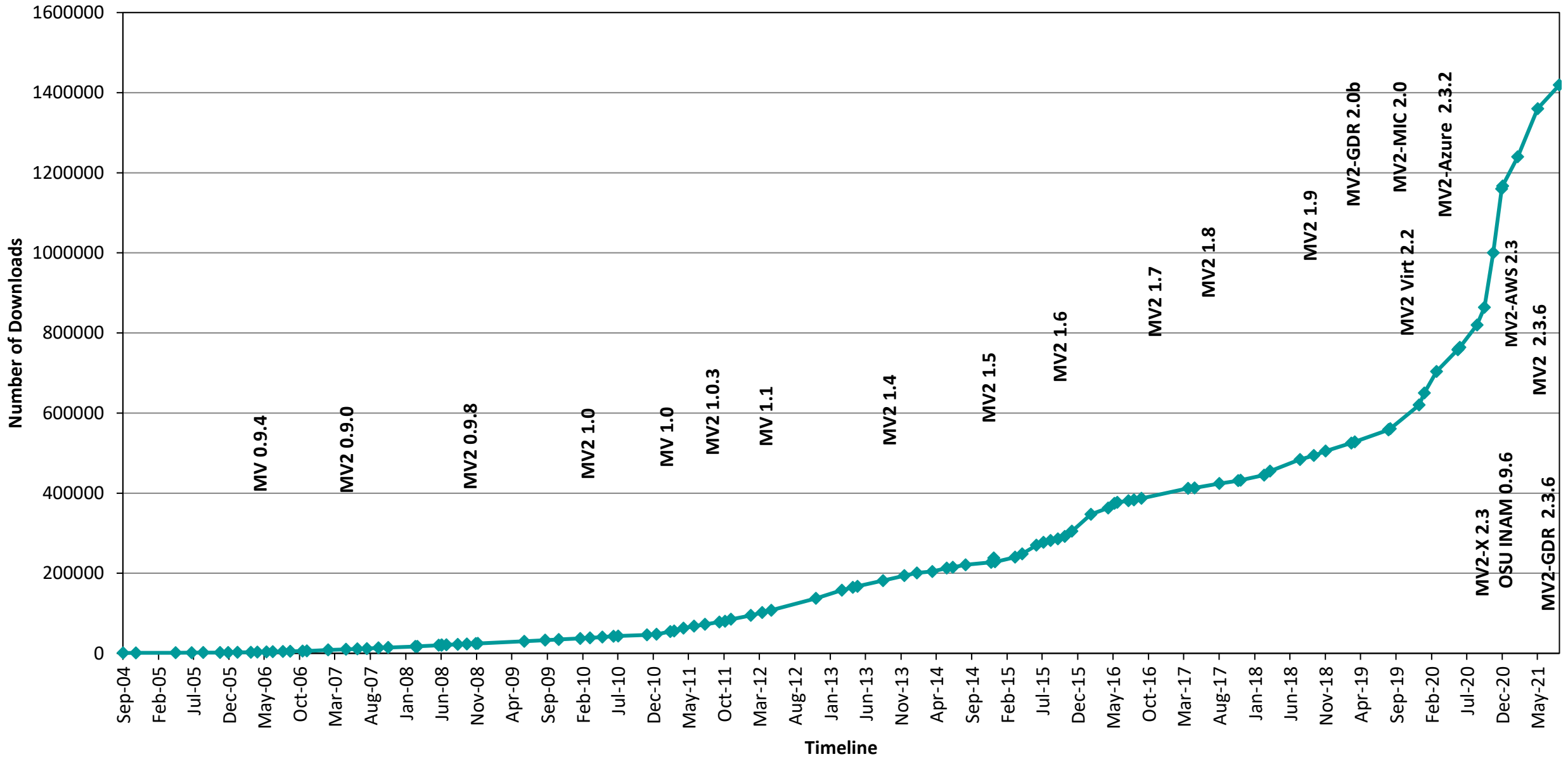
# Overview of the MVAPICH2 Project

- High Performance open-source MPI Library
- Support for multiple interconnects
  - InfiniBand, Omni-Path, Ethernet/iWARP, RDMA over Converged Ethernet (RoCE), AWS EFA, and Rockport Networks
- Support for multiple platforms
  - x86, OpenPOWER, ARM, Xeon-Phi, GPGPUs (NVIDIA and AMD)
- **Started in 2001, first open-source version demonstrated at SC '02**
- Supports the latest MPI-3.1 standard
- <http://mvapich.cse.ohio-state.edu>
- Additional optimized versions for different systems/environments:
  - MVAPICH2-X (Advanced MPI + PGAS), since 2011
  - MVAPICH2-GDR with support for NVIDIA (since 2014) and AMD (since 2020) GPUs
  - MVAPICH2-MIC with support for Intel Xeon-Phi, since 2014
  - MVAPICH2-Virt with virtualization support, since 2015
  - MVAPICH2-EA with support for Energy-Awareness, since 2015
  - MVAPICH2-Azure for Azure HPC IB instances, since 2019
  - MVAPICH2-X-AWS for AWS HPC+EFA instances, since 2019
- Tools:
  - OSU MPI Micro-Benchmarks (OMB), since 2003
  - OSU InfiniBand Network Analysis and Monitoring (INAM), since 2015



- **Used by more than 3,200 organizations in 89 countries**
- **More than 1.56 Million downloads from the OSU site directly**
- Empowering many TOP500 clusters (Nov '21 ranking)
  - **4<sup>th</sup>, 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China**
  - 13<sup>th</sup>, 448, 448 cores (Frontera) at TACC
  - 26<sup>th</sup>, 288,288 cores (Lassen) at LLNL
  - 38<sup>th</sup>, 570,020 cores (Nurion) in South Korea and many others
- Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, OpenHPC, and Spack)
- Partner in the 13<sup>th</sup> ranked TACC Frontera system
- **Empowering Top500 systems for more than 16 years**

# MVAPICH2 Release Timeline and Downloads



# Architecture of MVAPICH2 Software Family for HPC, DL/ML, and Data Science

## High Performance Parallel Programming Models

Message Passing Interface  
(MPI)

PGAS  
(UPC, OpenSHMEM, CAF, UPC++)

Hybrid --- MPI + X  
(MPI + PGAS + OpenMP/Cilk)

## High Performance and Scalable Communication Runtime

### Diverse APIs and Mechanisms

Point-to-point  
Primitives

Collectives  
Algorithms

Job Startup

Energy-  
Awareness

Remote  
Memory  
Access

I/O and  
File Systems

Fault  
Tolerance

Virtualization

Active  
Messages

Introspectio  
n & Analysis

### Support for Modern Networking Technology

(InfiniBand, iWARP, RoCE, Omni-Path, Elastic Fabric Adapter)

#### Transport Protocols

RC

SRD

UD

DC

#### Modern Features

UMR

ODP

SR-  
IOV

Multi  
Rail

### Support for Modern Multi-/Many-core Architectures

(Intel-Xeon, OpenPOWER, Xeon-Phi, ARM, NVIDIA GPGPU)

#### Transport Mechanisms

Shared  
Memory

CMA

IVSHMEM

XPMEM

#### Modern Features

Optane\*

NVLink

CAPI\*

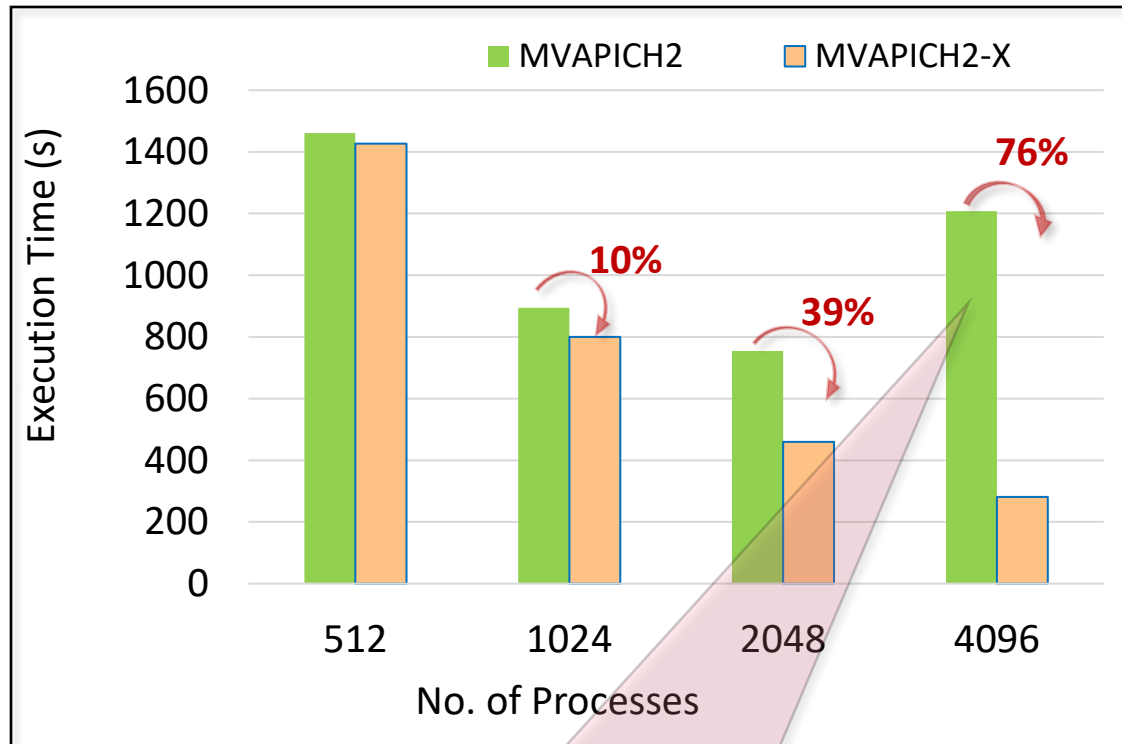
\* Upcoming

# Highlights of some of the MVAPICH2 Designs

- **Direct Connect (DC) Protocol for Scalable inter-node communication with Reduced Memory Footprint**
- **Scalable Collective Communication Support with SHARP In-network Computing**
- **Hardware Tag-Matching Support**
- **Optimized Derived Datatype Support**
- **Non-blocking Collective Support with DPUs**
- **QoS-aware Designs**
- **Exploiting Rockport Network Features**

# Impact of DC Transport Protocol on Neuron

## Neuron with YuEtAl2012



**Overhead of RC protocol for connection establishment and communication**

- Up to **76%** benefits over MVAPICH2 for Neuron using Direct Connected transport protocol at scale
  - VERSION 7.6.2 master (f5a1284) 2018-08-15
- Numbers taken on bbpv2.epfl.ch
  - Knights Landing nodes with 64 ppn
  - ./x86\_64/special -mpi -c stop\_time=2000 -c is\_split=1 parinit.hoc
  - Used “runtime” reported by execution to measure performance
- Environment variables used
  - MV2\_USE\_DC=1
  - MV2\_NUM\_DC\_TGT=64
  - MV2\_SMALL\_MSG\_DC\_POOL=96
  - MV2\_LARGE\_MSG\_DC\_POOL=96
  - MV2\_USE\_RDMA\_CM=0

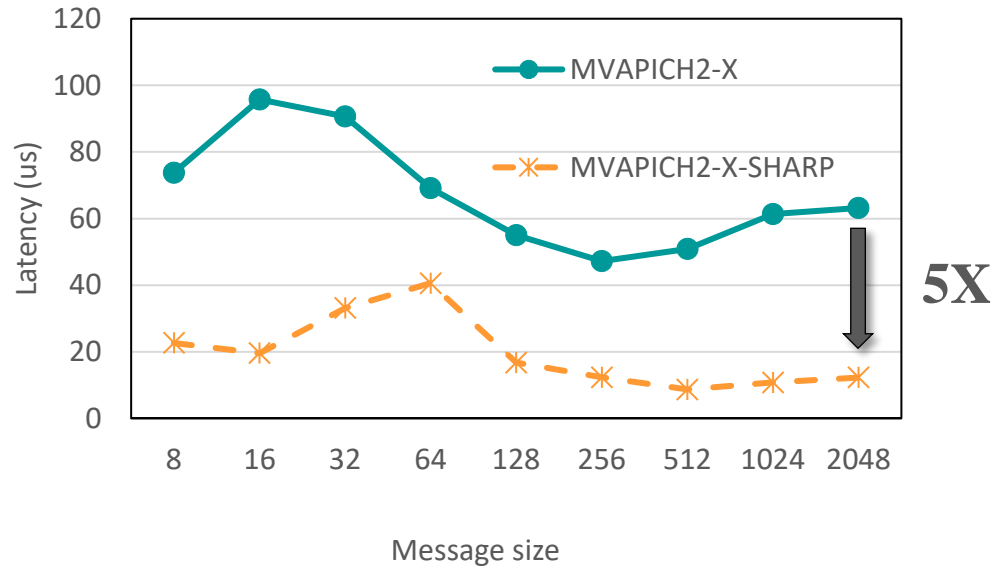
*More details in talk*

*“Building Brain Circuits: Experiences with shuffling terabytes of data over MPI”, by Matthias Wolf at MUG’20*

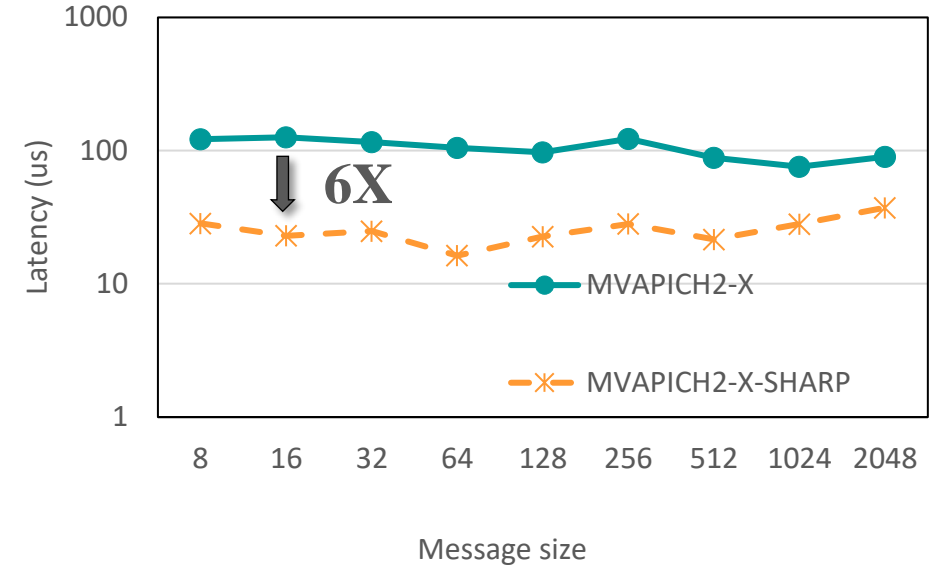
<https://www.youtube.com/watch?v=TFi8O3-Hznw>

# Performance of Collectives with SHARP on TACC Frontera

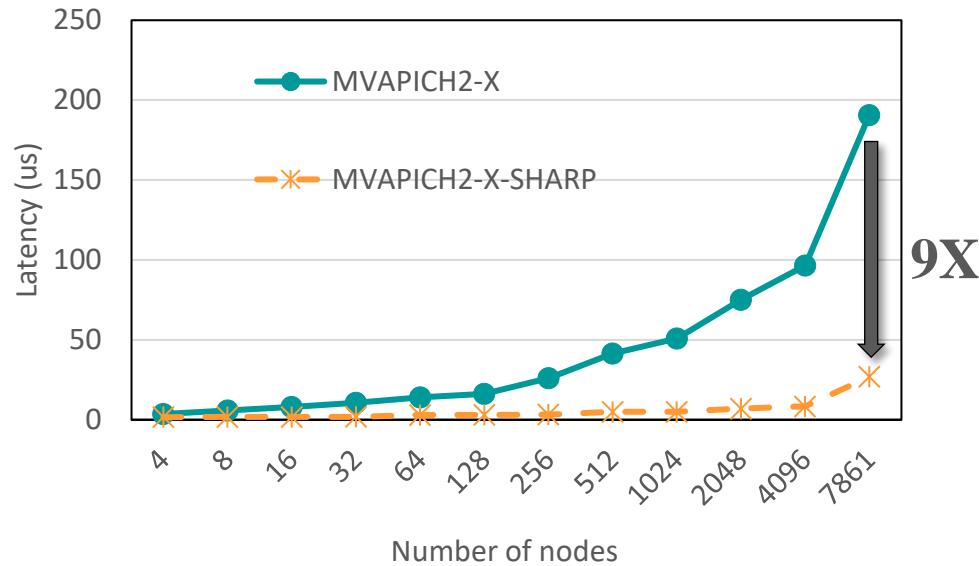
MPI\_Allreduce  
(PPN = 1, Nodes = 7861)



MPI\_Reduce  
(PPN = 1, Nodes = 7861)



MPI\_Barrier



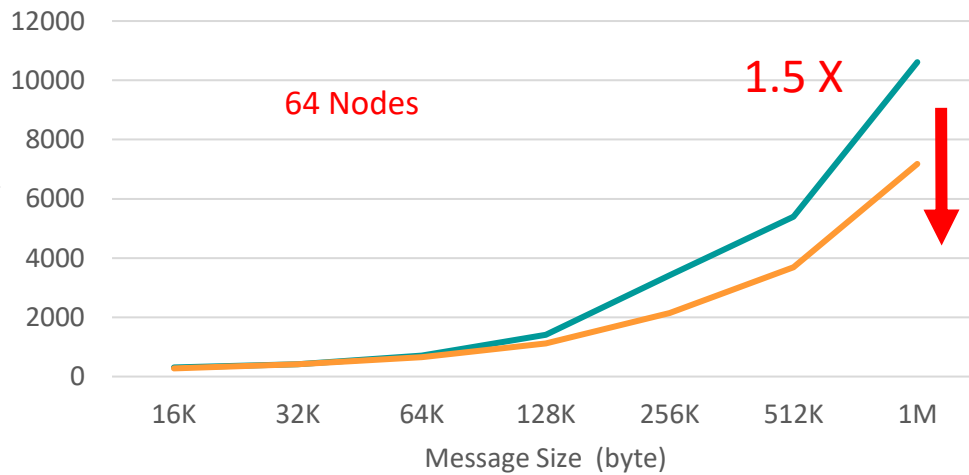
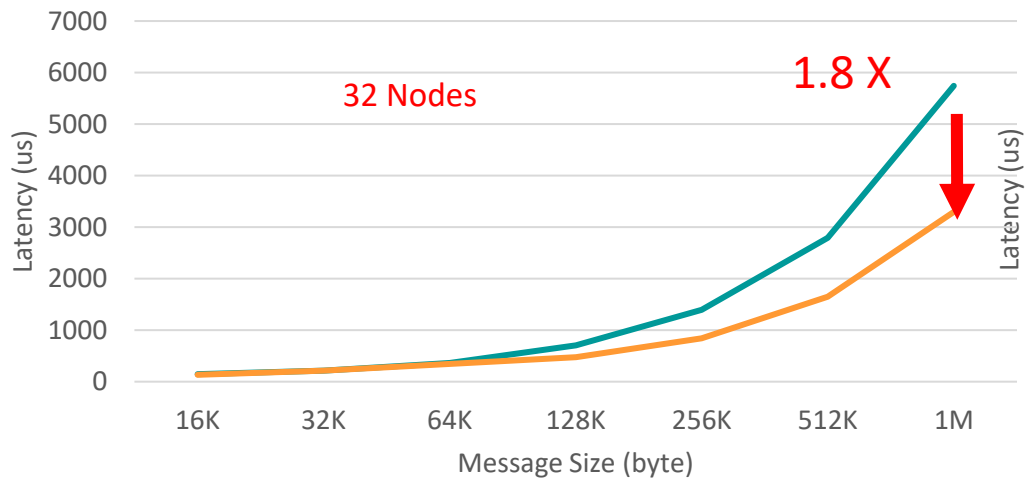
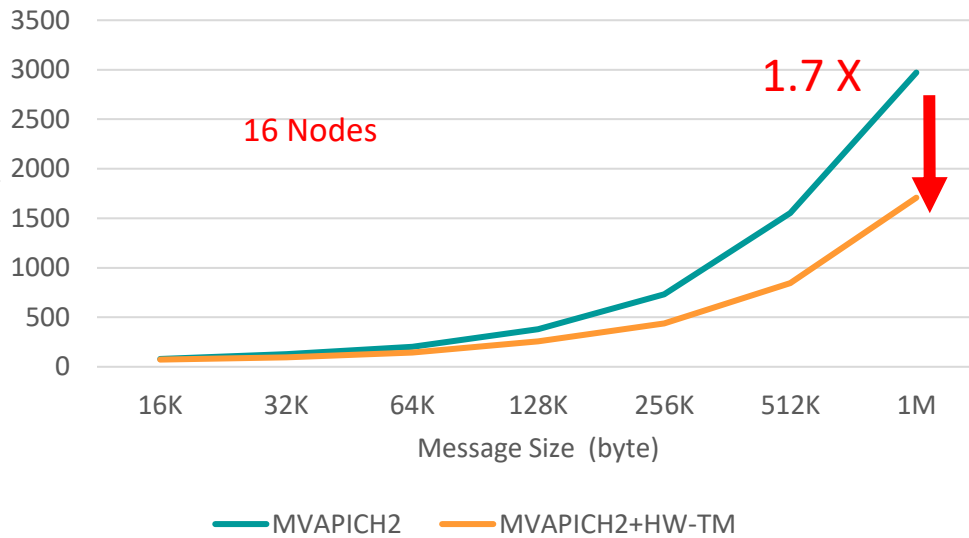
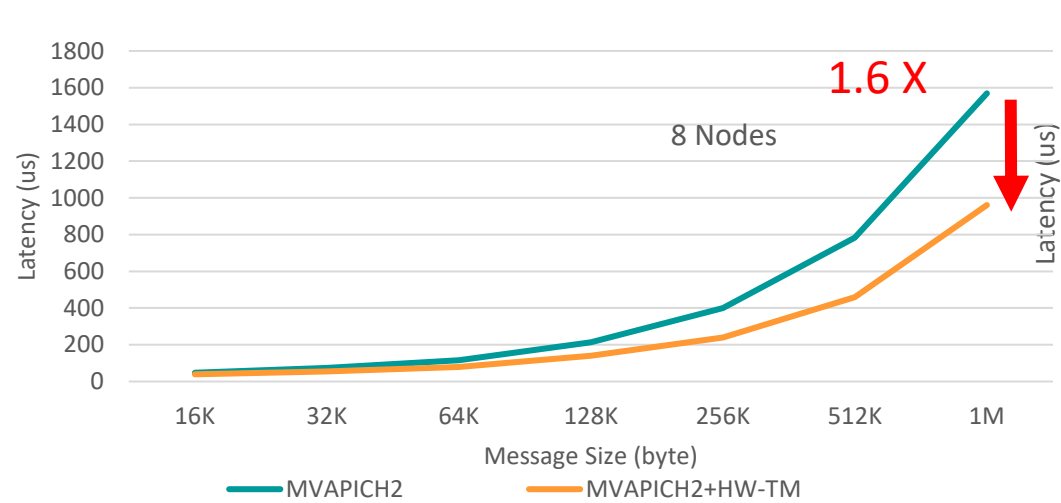
## Optimized SHARP designs in MVAPICH2-X

**Up to 9X** performance improvement with SHARP over MVAPICH2-X default for 1ppn MPI\_Barrier, **6X** for 1ppn MPI\_Reduce and **5X** for 1ppn MPI\_Allreduce

B. Ramesh , K. Suresh , N. Sarkauskas , M. Bayatpour , J. Hashmi , H. Subramoni , and D. K. Panda, Scalable MPI Collectives using SHARP: Large Scale Performance Evaluation on the TACC Frontera System, ExaMPI2020 - Workshop on Exascale MPI 2020, Nov 2020.

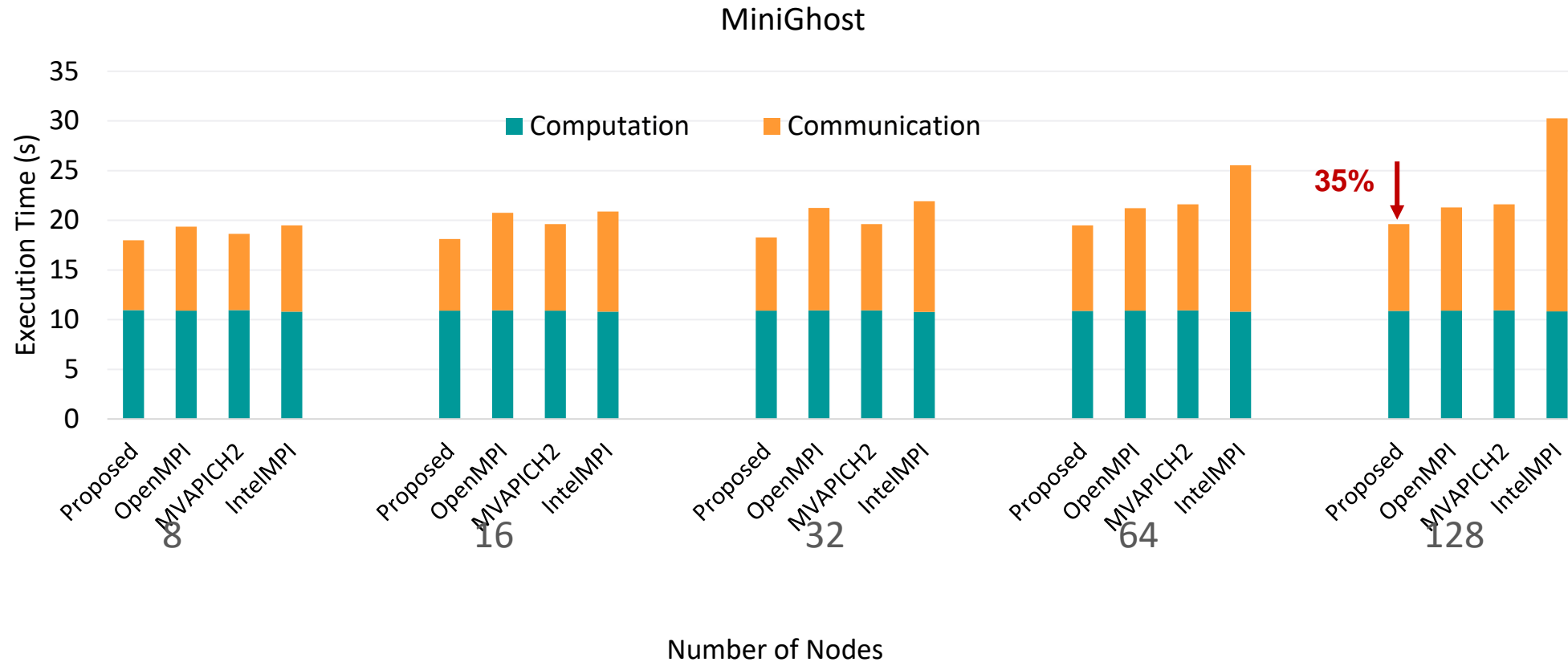
Optimized Runtime Parameters: MV2\_ENABLE\_SHARP = 1

# Performance of MPI\_lalltoall using HW Tag Matching



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

# Optimized Derived Data Type (DDT) Processing



- Execution time of the proposed scheme is up to **35%** better than Intel-MPI, **7.8%** better than OpenMPI, and **9%** better than MVAPICH2 at a scale of 128 nodes (7K cores).

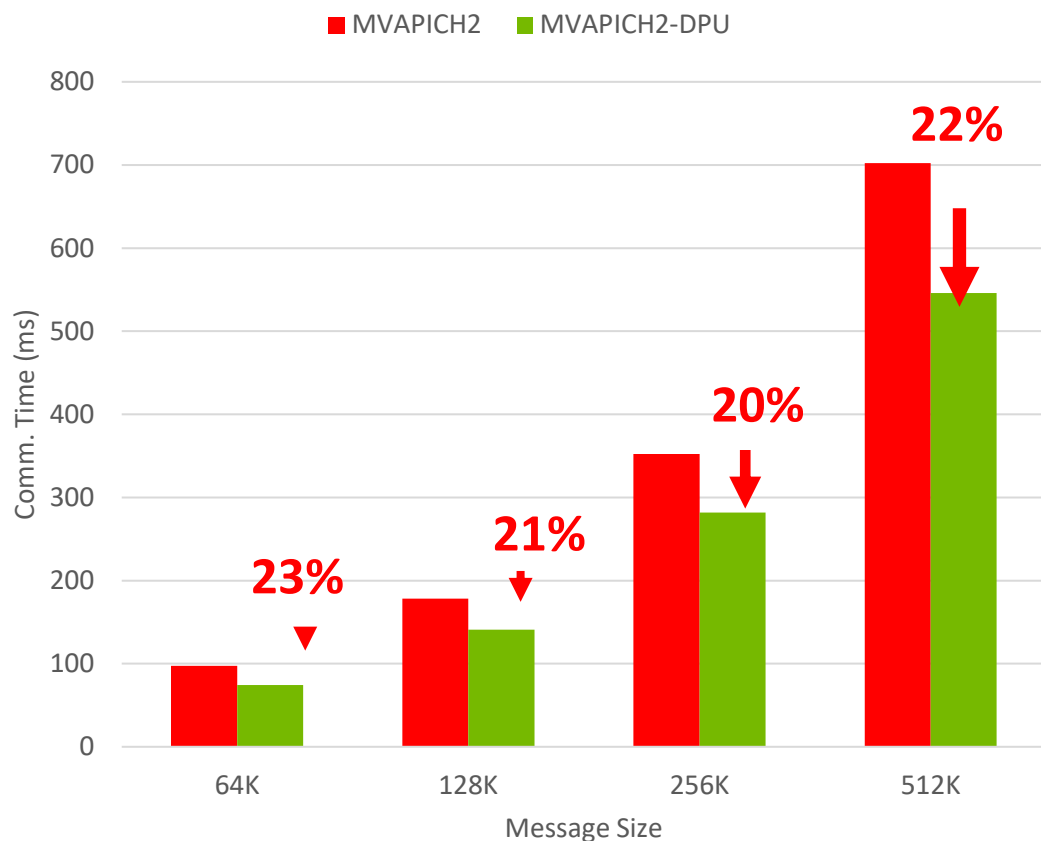
K. Suresh, C. Chen, B. Ramesh, SM Ghazimirsaeed, M. Bayatpour, A. Shafi, H. Subramoni and DK Panda,  
“Layout-aware Hardware-assisted Designs for Derived Data Types in MPI”, HiPC ’21

# Highlights of some of the MVAPICH2 Designs

- Direct Connect (DC) Protocol for Scalable inter-node communication with Reduced Memory Footprint
- Scalable Collective Communication Support with SHARP In-network Computing
- Hardware Tag-Matching Support
- Optimized Derived Datatype Support
- **Non-blocking Collective Support with DPUs**
- **QoS-aware Designs**
- **Exploiting Rockport Network Features**

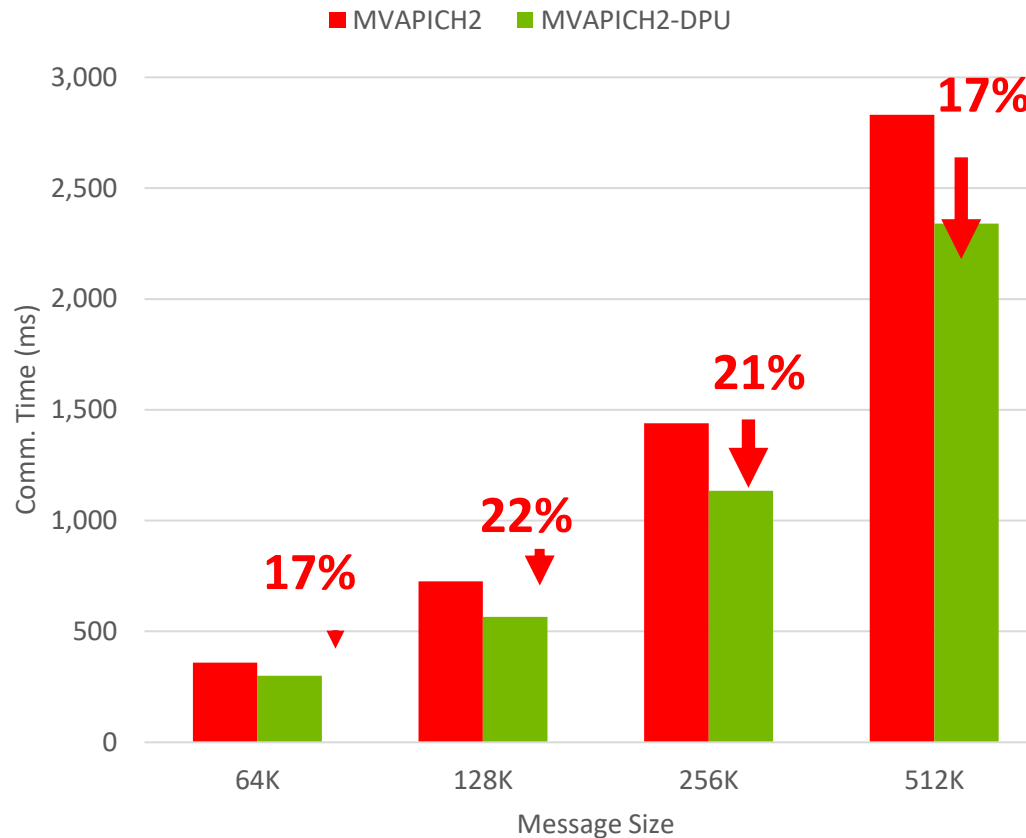
# Total Execution Time with osu\_ialltoall (32 nodes)

Total Execution Time, BF-2 (osu\_ialltoall)



32 Nodes, 16 PPN

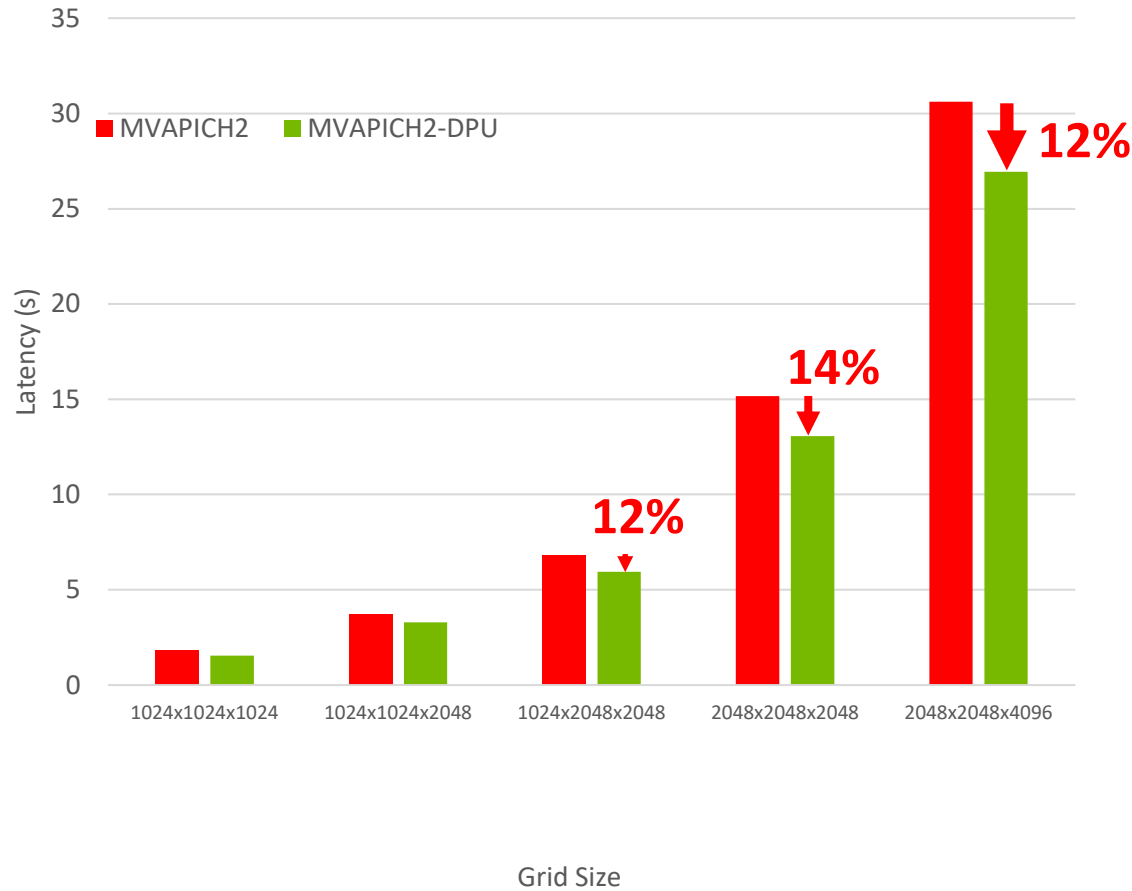
Total Execution Time, BF-2 (osu\_ialltoall)



32 Nodes, 32 PPN

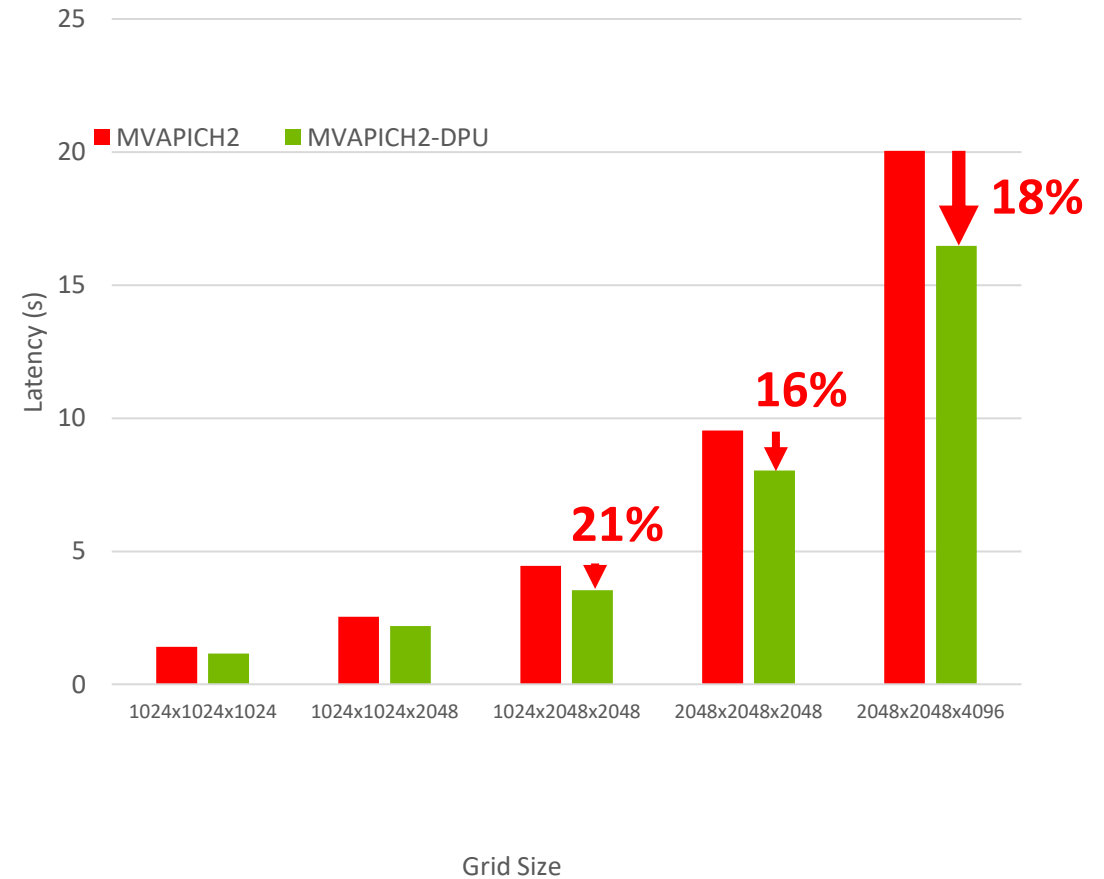
Benefits in Total execution time (Compute + Communication)

# P3DFFT Application Execution Time (32 nodes)



**32 Nodes, 16 PPN**

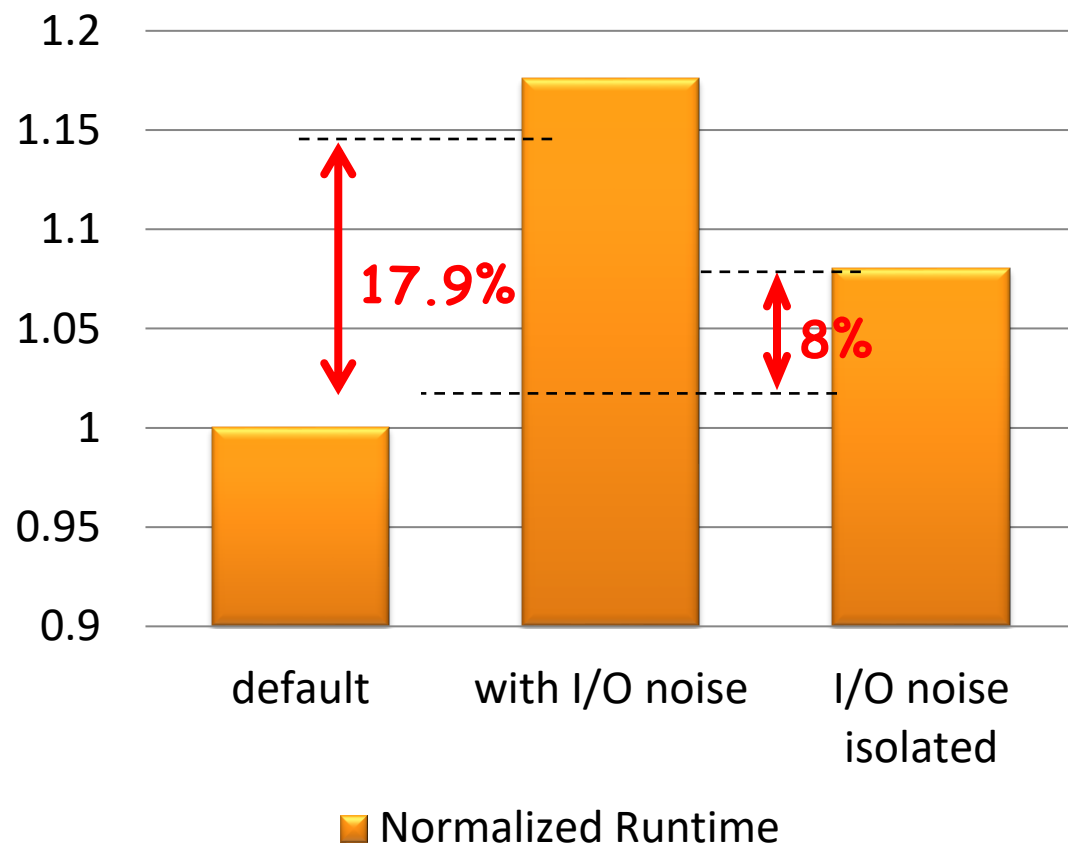
**Benefits in application-level execution time**



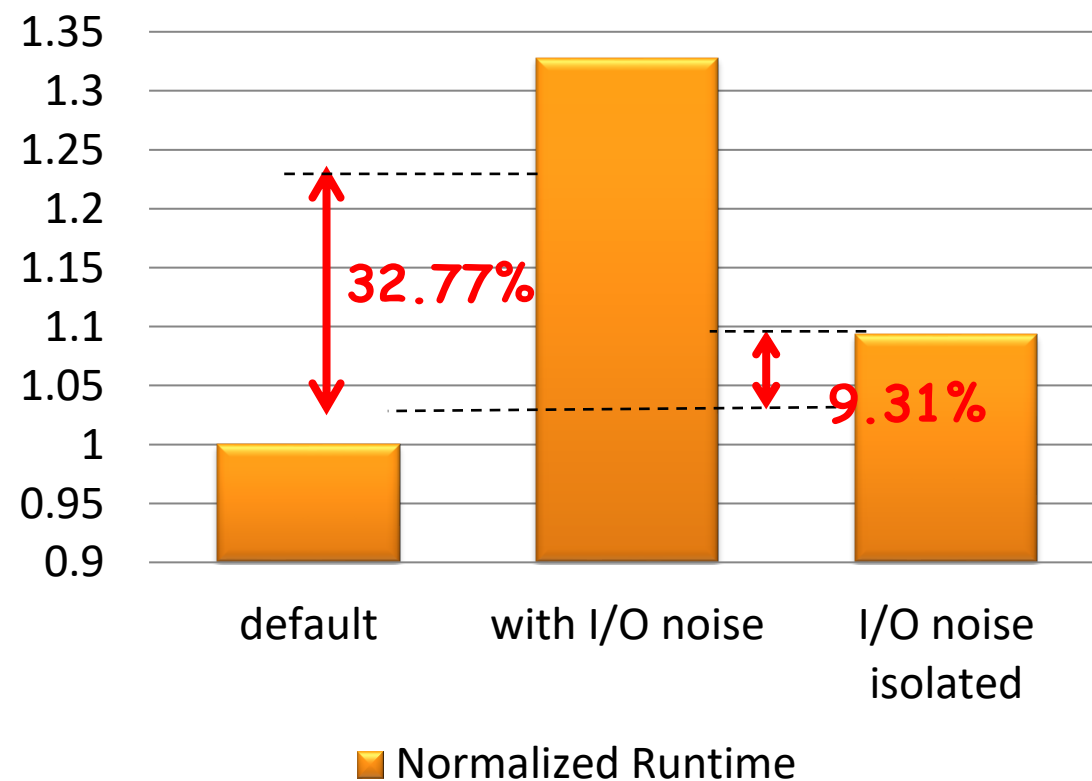
**32 Nodes, 32 PPN**

# Impact Isolating MPI and I/O Traffic on Applications with QoS Support

Anelastic Wave Propagation  
(64 MPI processes)

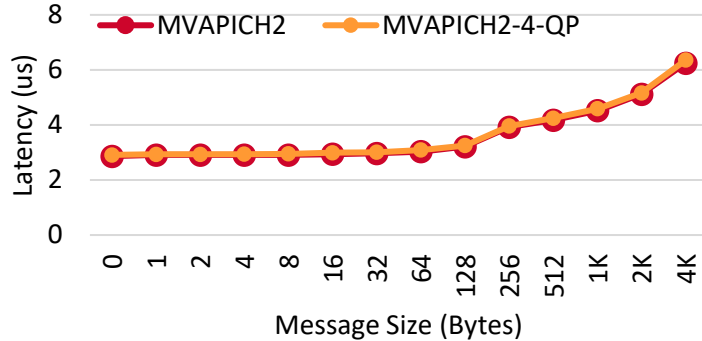


NAS Parallel Benchmark  
Conjugate Gradient Class D  
(64 MPI processes)

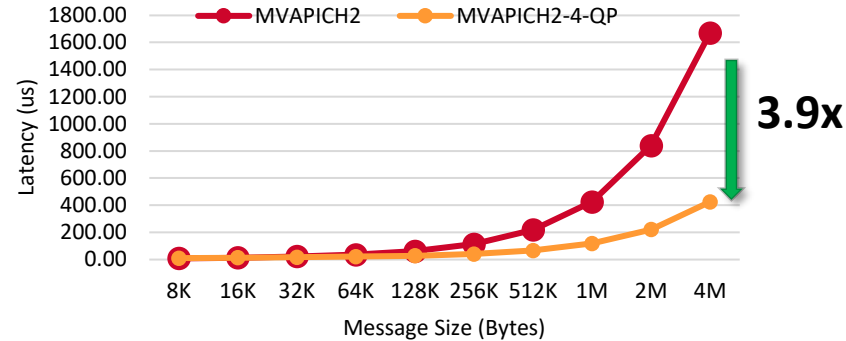


# Inter-node point-to-point Latency and Bandwidth (Rockport Networks with Multiple QPs)

### Small message Latency



### Medium/Large message Latency

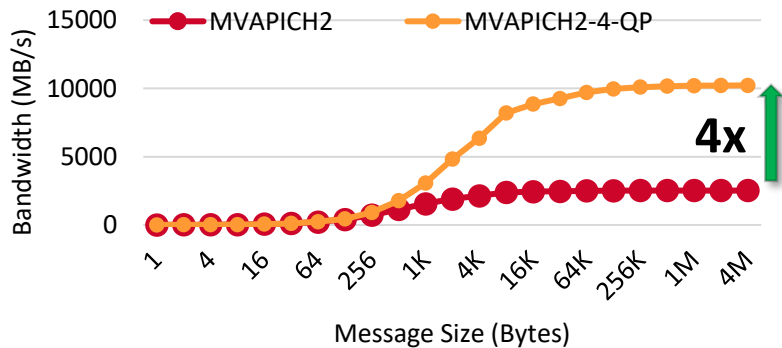


- MVAPICH2 delivers around 3.0 microsec latency for small messages
- Using multiple QPs gives up to 3.9x reduction in latency

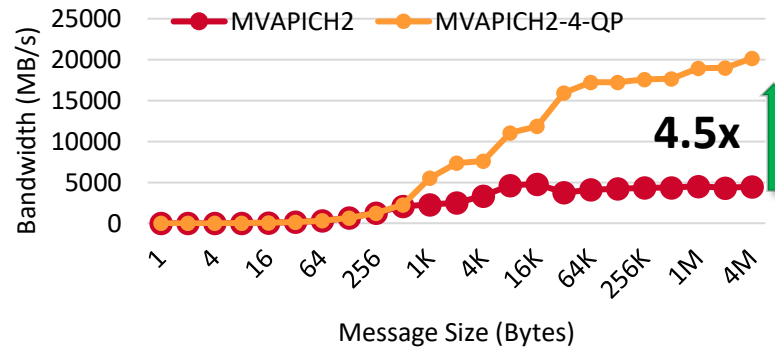
MVAPICH2-4-QP delivers

- 10229 MB/sec peak unidirectional bandwidth
- 20165 MB/Sec peak bi-directional bandwidth
- **Upcoming MVAPICH2 2.3.7 release**

### Uni-directional Bandwidth

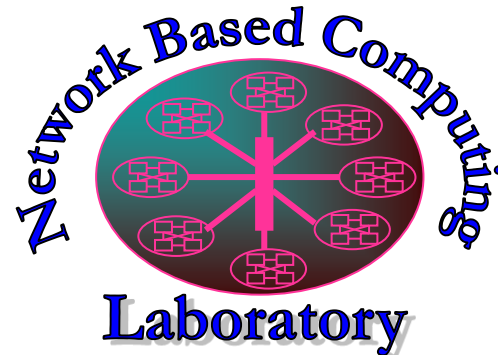


### Bi-Directional Bandwidth



# Thank You!

[panda@cse.ohio-state.edu](mailto:panda@cse.ohio-state.edu)



Follow us on

<https://twitter.com/mvapich>

Network-Based Computing Laboratory

<http://nowlab.cse.ohio-state.edu/>



**MVA PICH**

MPI, PGAS and Hybrid MPI+PGAS Library

The High-Performance MPI/PGAS Project

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



**HiBD**

High-Performance  
Big Data

The High-Performance Big Data Project

<http://hibd.cse.ohio-state.edu/>



**HiDL**

High-Performance  
Deep Learning

The High-Performance Deep Learning Project

<http://hidl.cse.ohio-state.edu/>

# Smarter Networks Smarter MPI

BROAD PERSPECTIVES FROM THE MPI COMMUNITY

# MPI and Network Hardware

Standard has been developed with hardware in mind

- Version 1: How industry was doing things already
- Later versions: Support hardware that is built for MPI performance
  - RMA
  - Message Matching
  - Collectives
  - Concurrency

# Smarter Networks

Networks can adapt to MPI behaviour and support it

- Smart adaptable NICs provide a good research environment
- Quickly adaptable to explore usefulness of offloads
- Prototype solutions for standardization activities

Networks need not integrate with MPI itself to improve it

- Detect traffic patterns and adapt
- General control path improvements can be exposed to MPI

# Separating Control and Data

MPI has tight coupling between control and data

- MPI\_Send/Recv dictate control and data movement
- MPI\_Isend/Irecv still dictate both control and data movement but don't wait for completion before returning

Control path can be heavy weight

- Many proposals to restrict wildcard sources/tags to enable faster message matching

# MPI Momentum

## Huge code legacy with MPI-1

- Not just resistance to change, the two-sided send/recv model is popular
  - Allows easy reasoning about what data is moving when
  - Doesn't hide data movement implicitly behind other calls (you know what's expensive)
- RMA (one-sided) never caught on
  - Hard to program for many domain scientists (not computer scientists)
  - Hard to debug when things don't work
  - Had a non-ideal rollout that impacted perceptions in the community

# Leveraging Smarter Networks

Do whatever you can without needing the applications to change

- Help behind the scenes to optimize performance and “hide” network optimizations

Provide performance with the interface the community wants

- E.g. Partitioned communication – new in MPI 4.0
- Hides RDMA behind the covers under a two-sided like interface

# Hiding the Hard Parts

RDMA programming is much like non-coherent cache

- Hard to reason about where data is available and when
- Not impossible, but a significant departure from past CS curriculums

Let the MPI implementors do the hard work

- They always have before, so they can take on more burden with the complexity of the interface
- Smart Networks help here in allowing implementors to do the right thing performance wise while giving the interface applications want


# Final Thoughts

## Smarter networks benefit everyone

- Performance for no application development cost
- Performance for little or no middleware development cost

## Flexible smart networks drive innovation

- Good prototyping solution
- First to get new features
- Customize to userbase need for facilities



# Advances in Interconnect/MPI Integration

Matthew Williams - Rockport Networks



# Characteristics of a Smarter Network

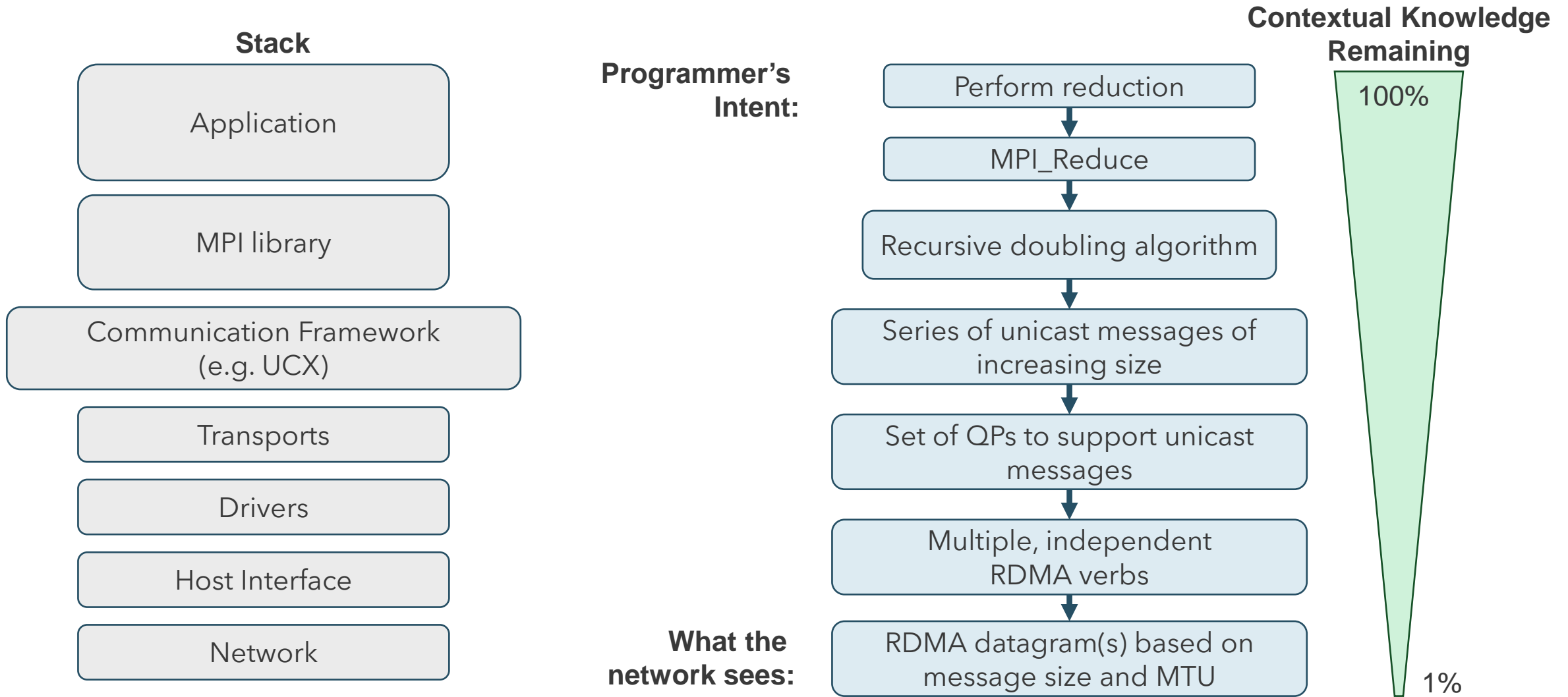
Tighter relationships between applications and the network

Providing application traffic with what it cares about - latency vs BW

Real-time adaptability on a per-packet basis

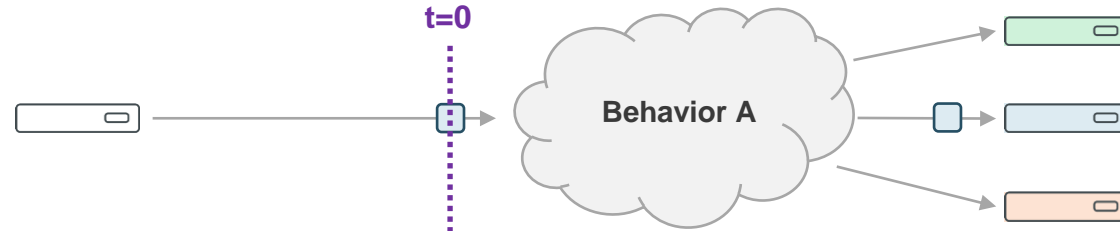
Job independence

# Loss of Knowledge Through the Stack

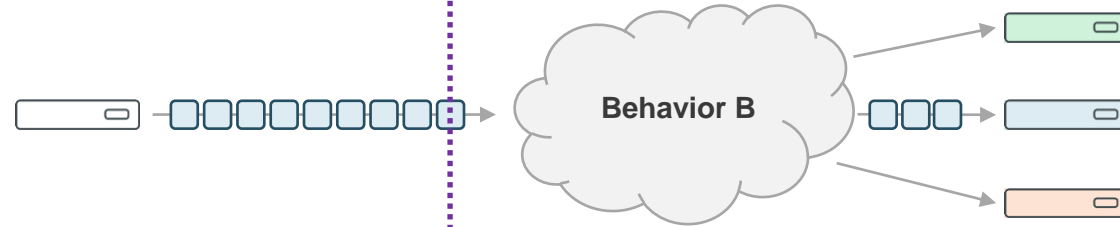


# Traffic Patterns and Optimum Network Behaviors

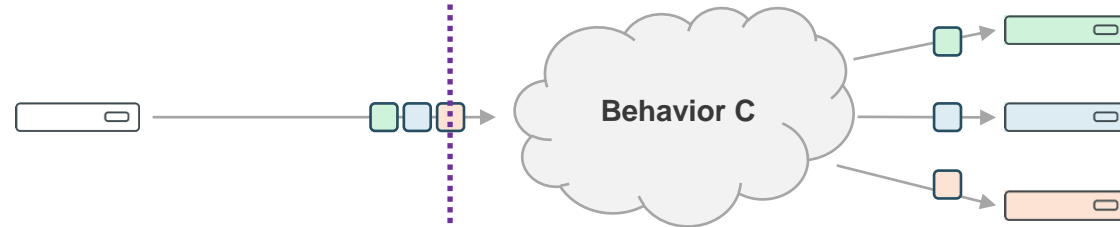
**1-to-1  
small message**



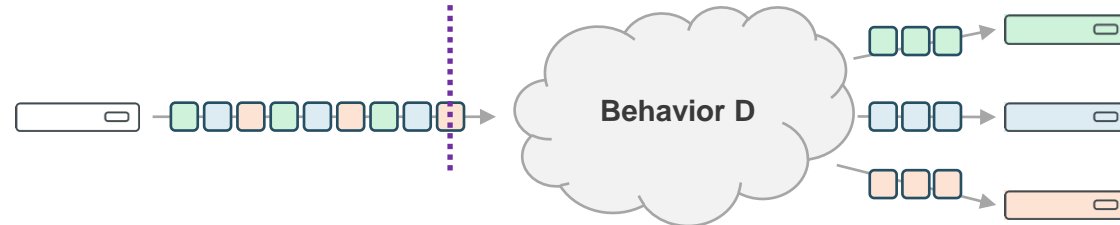
**1-to-1  
large message  
(or storage access)**



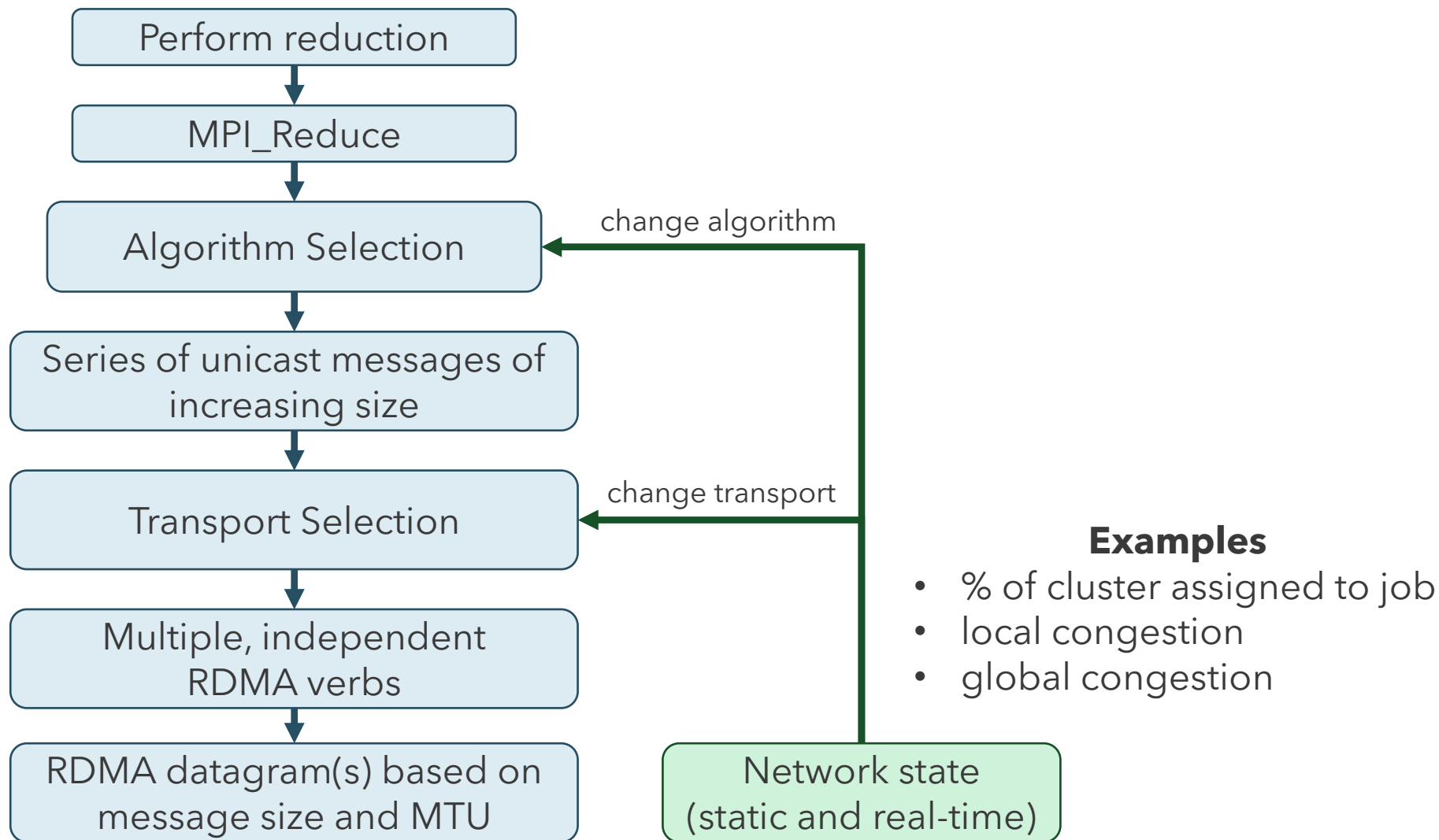
**1-to-many  
small messages**



**1-to-many  
large messages**



# Feeding Network Knowledge to the Stack



# Rethinking Network Performance at Scale

Rockport has reimagined performance networks with an embeddable switchless architecture that delivers the performance at scale needed for HPC, HPDA, and AI/ML/DL.

By distributing the network switching function into each device endpoint,

**the nodes become the network:**

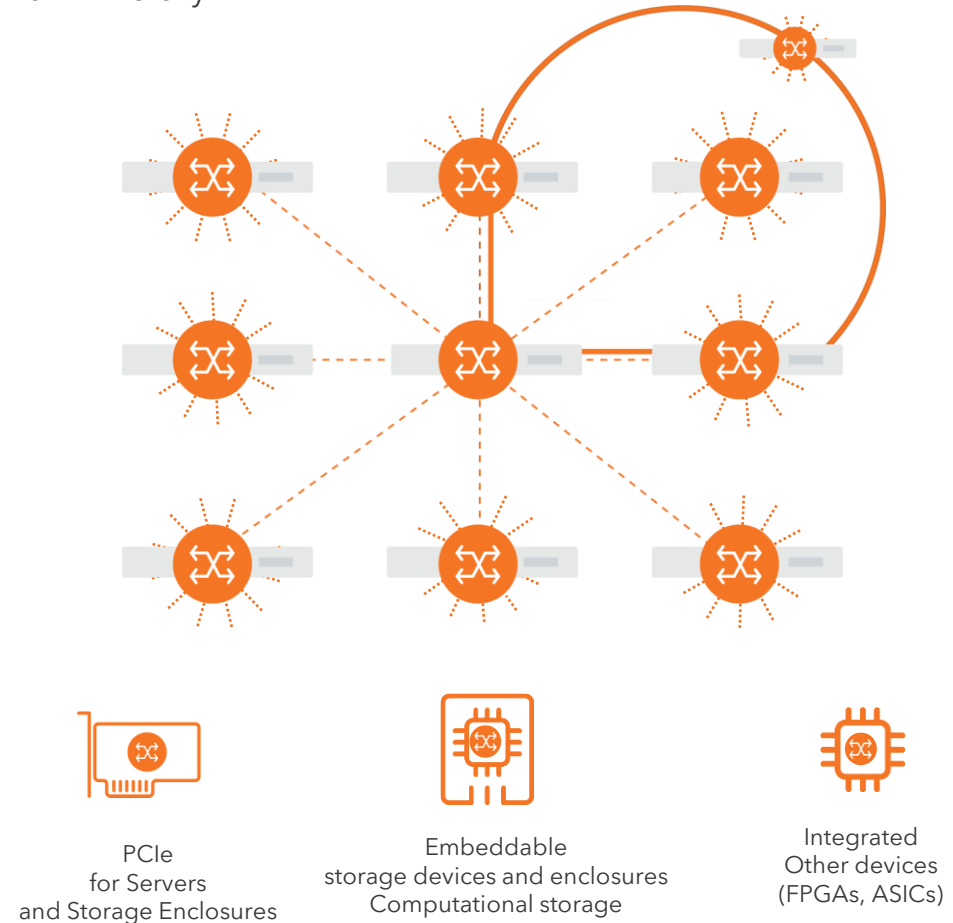
- Standard Ethernet-based host interface (RDMA/RoCEv2 and TCP/UDP)
- Supports all Parallel Programming Models
- Working with HPC community to deliver smarter MPI and smarter networks
- Direct interconnect
- Per-packet adaptive routing
- Advanced congestion control
- Critical messages immune to congestion
- Rich tooling with deep insights
- Linear, elastic scaling

## Rockport Architecture

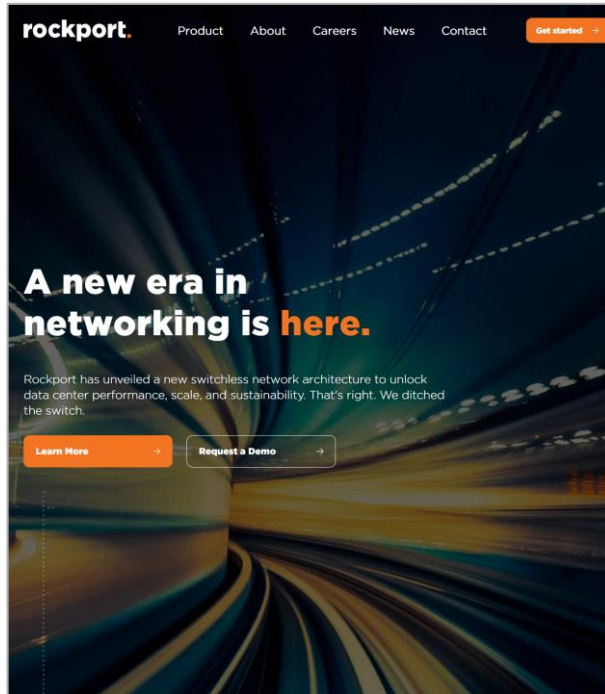
Directly Connected Nodes with no external switches

Distributed Switching

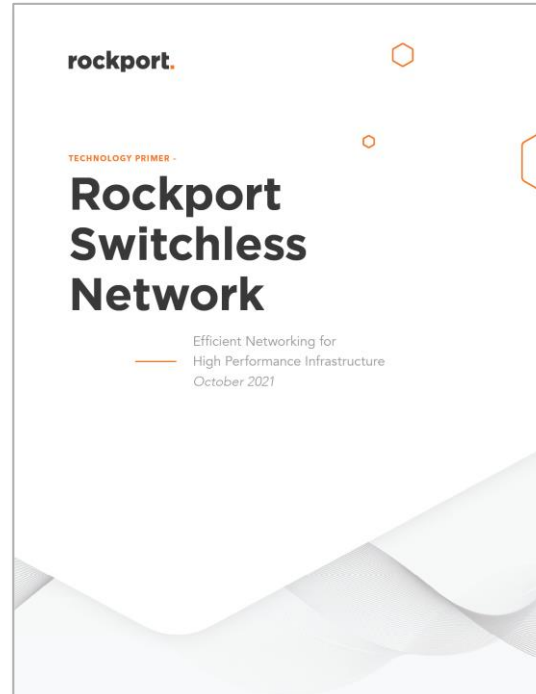
Very High Path Diversity



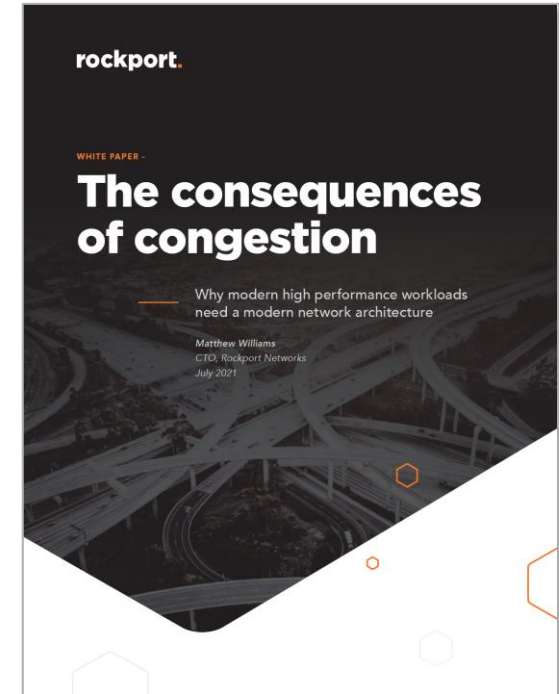
# Learn more about Rockport:



Visit  
[rockportnetworks.com](https://rockportnetworks.com)



Download Rockport  
Technology Primer



Download Congestion  
Whitepaper

Contact us at [info@rockportnetworks.com](mailto:info@rockportnetworks.com)

**rockport.**



# Q&A

# Thank You