MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

Follow us on

https://twitter.com/mvapich

High-Performance and Smart Networking Technologies for HPC and AI

A Tutorial Presented at HotI 32 (HotI 2025)

Latest Slides Can Be Found at: https://go.osu.edu/hoti2025-hpn

Dhabaleswar K. (DK) Panda

The Ohio State University

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

https://cse.osu.edu/people/panda.2

Benjamin Michalowicz

The Ohio State University

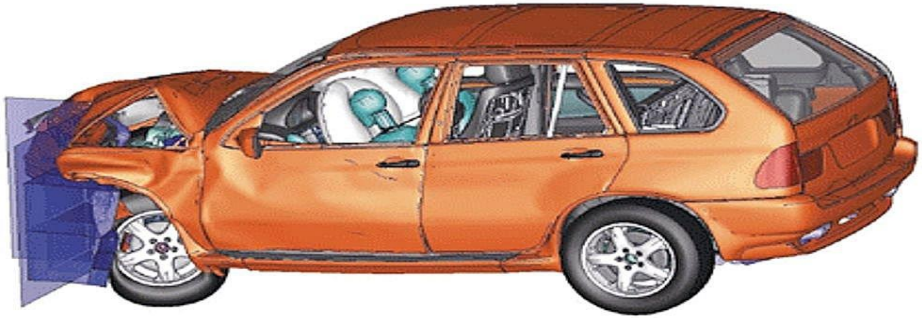
E-mail: michalowicz.2@osu.edu

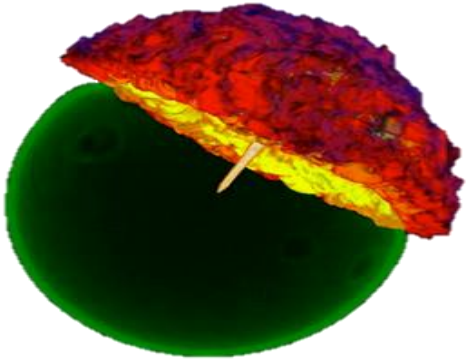
https://engineering.osu.edu/people/michalowicz.2

Presentation Overview

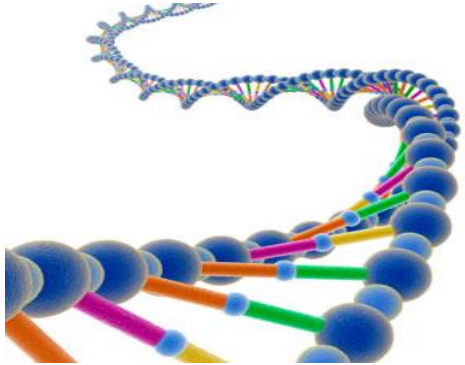
- **Introduction**
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
  - IB, HSE, their Convergence and Features
  - GPU-aware support in modern HPC networks:
    - NVLink and NVSwitch Interconnect Architecture
    - AMD Infinity Fabric Interconnect Architecture, UALink, & UltraEthernet
  - Amazon EFA Interconnect Architecture
  - Cray Slingshot Interconnect Architecture
- Overview of Emerging Smart Network Interfaces
  - Collectives w/ NVIDIA SHARP, NVIDIA BlueField DPUs, AMD Pensando Smart NICs, and Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

Current and Next Generation Applications and Computing Systems

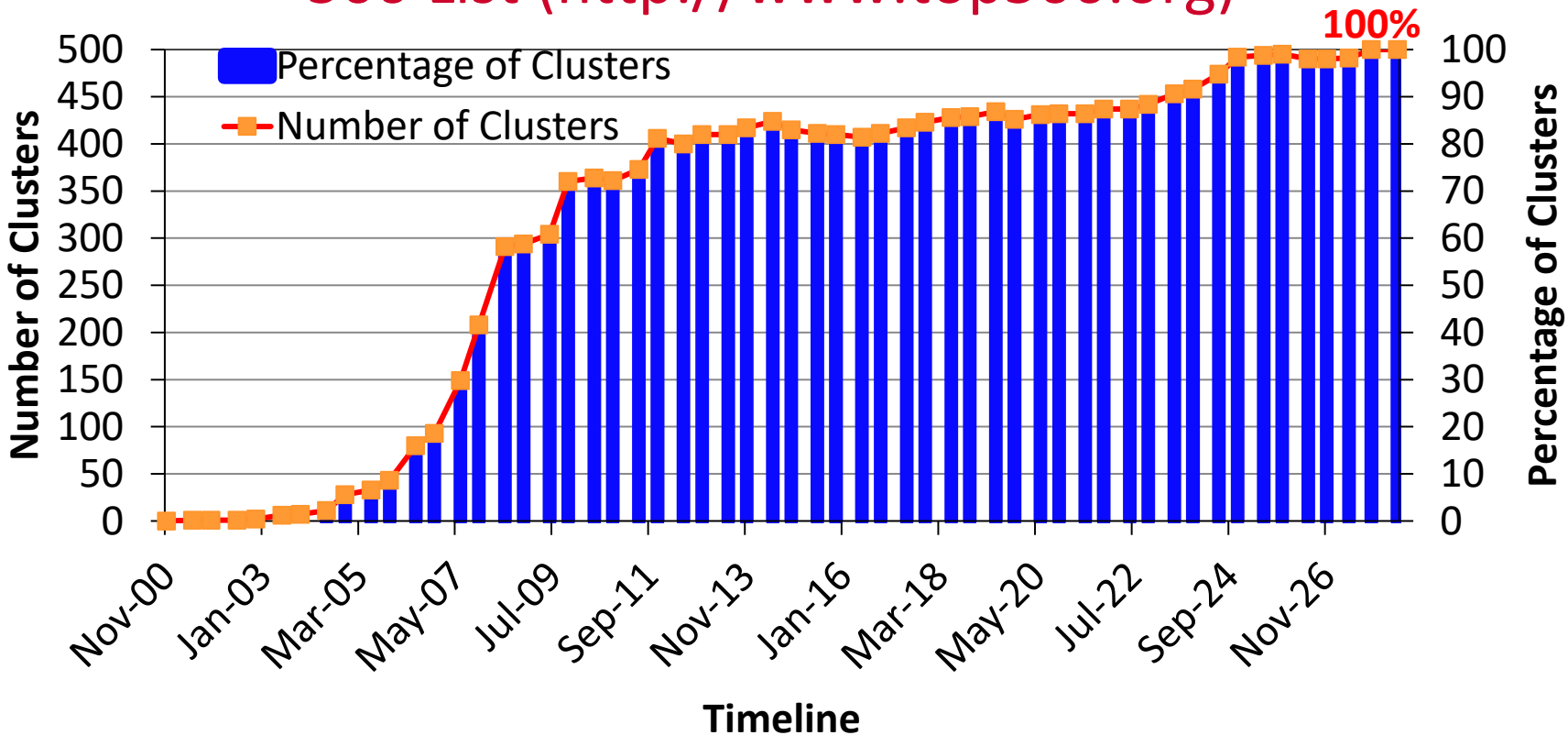




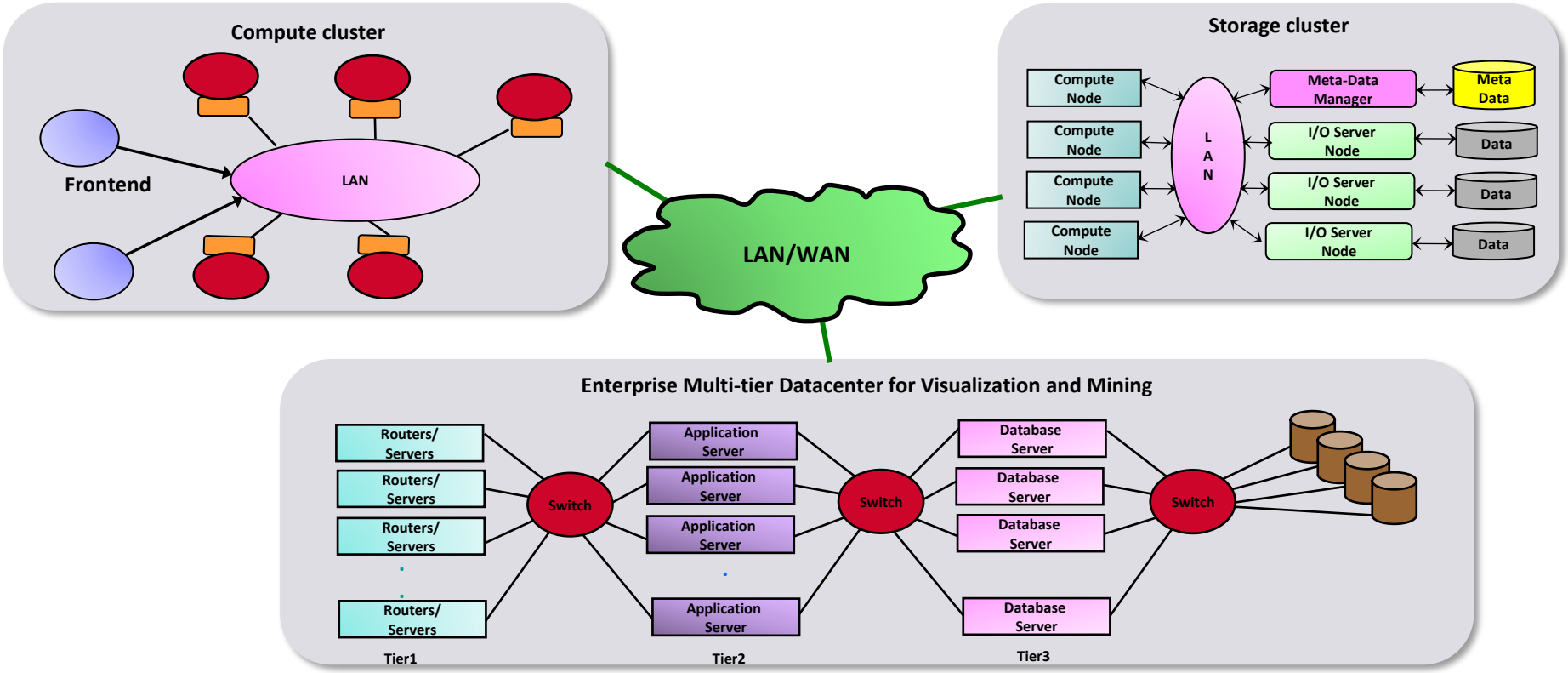
- Growth of High-Performance Computing
  - Growth in processor performance
    - Chip density doubles every 18 months
  - Growth in commodity networking
    - Increase in speed/features + reducing cost
- Clusters: popular choice for HPC
  - Scalability, Modularity and Upgradeability



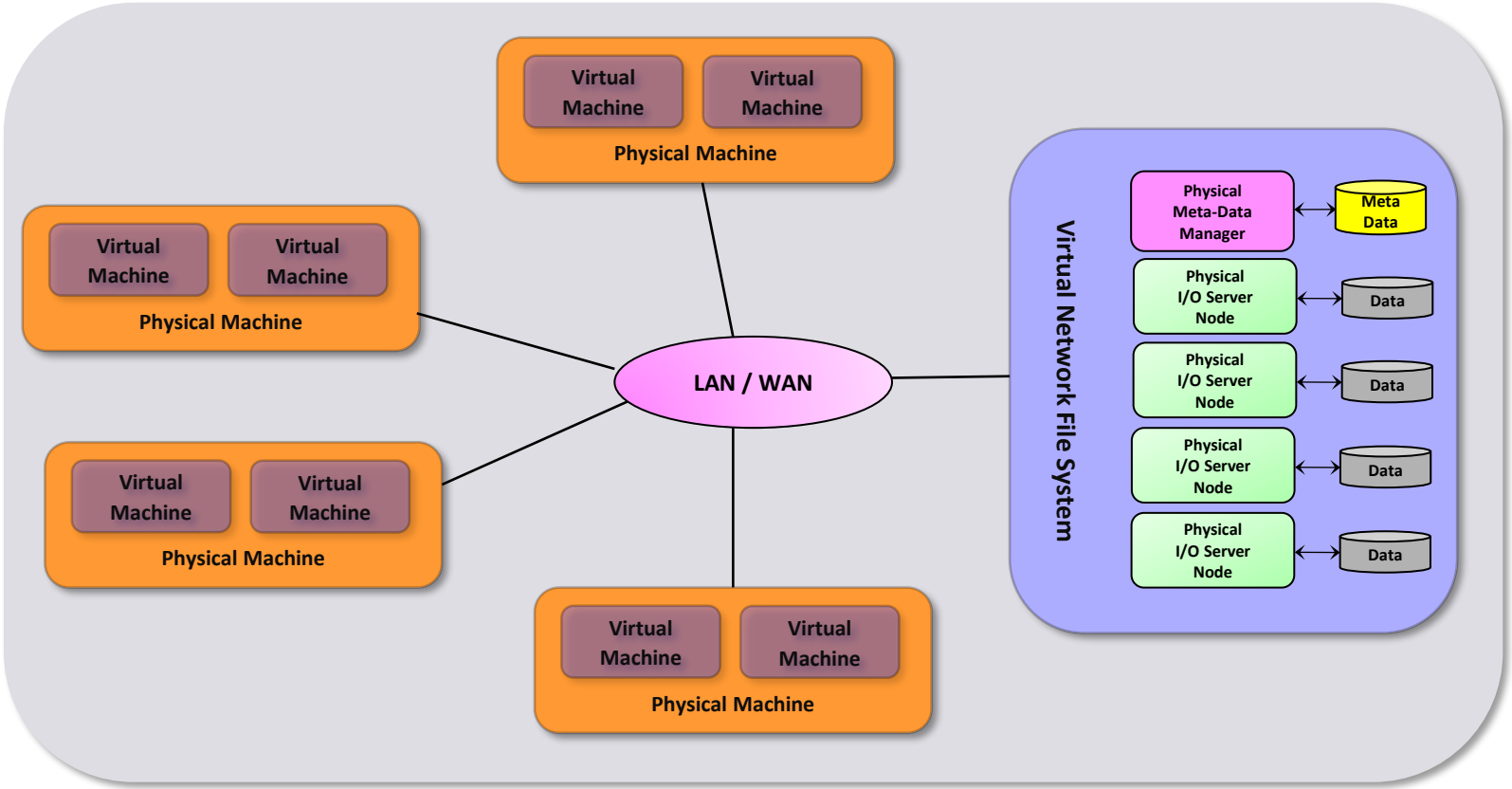
Trends for Commodity Computing Clusters in the Top 500 List (http://www.top500.org)



Integrated High-End Computing Environments

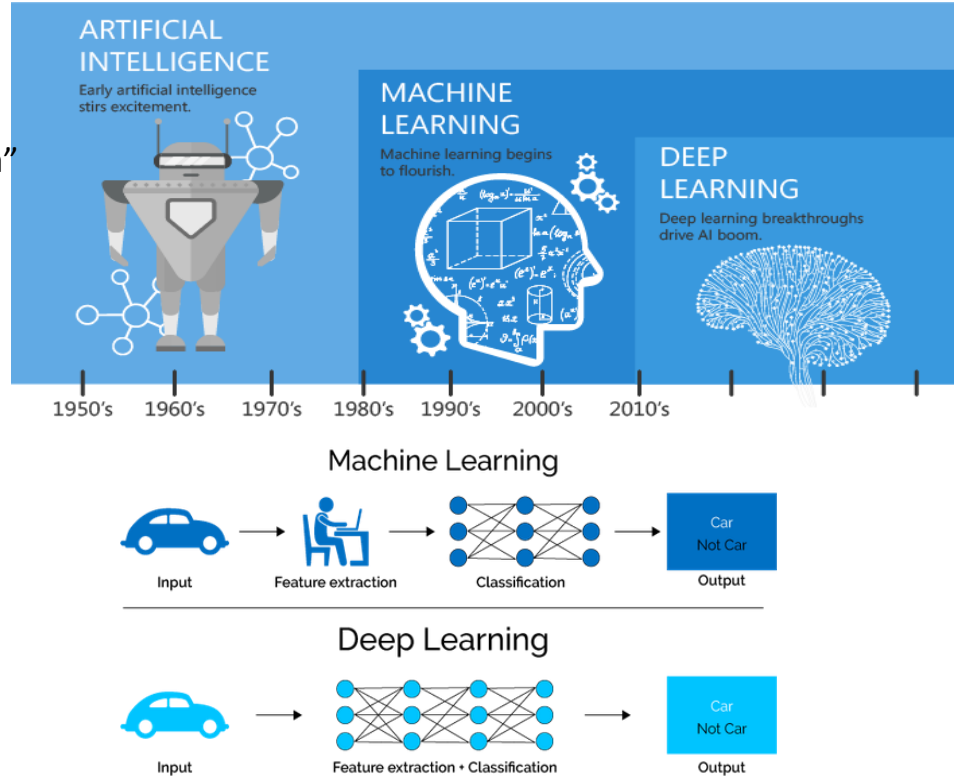


Cloud Computing Environments



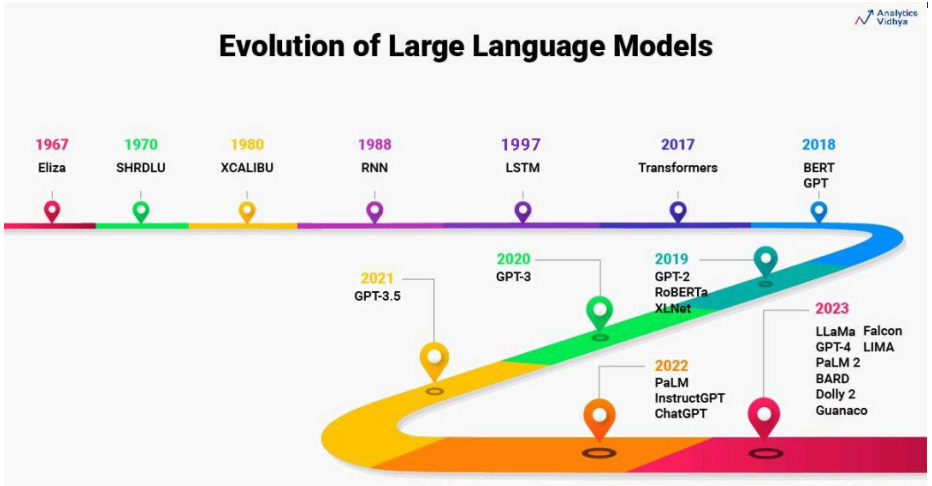
Deep/Machine Learning

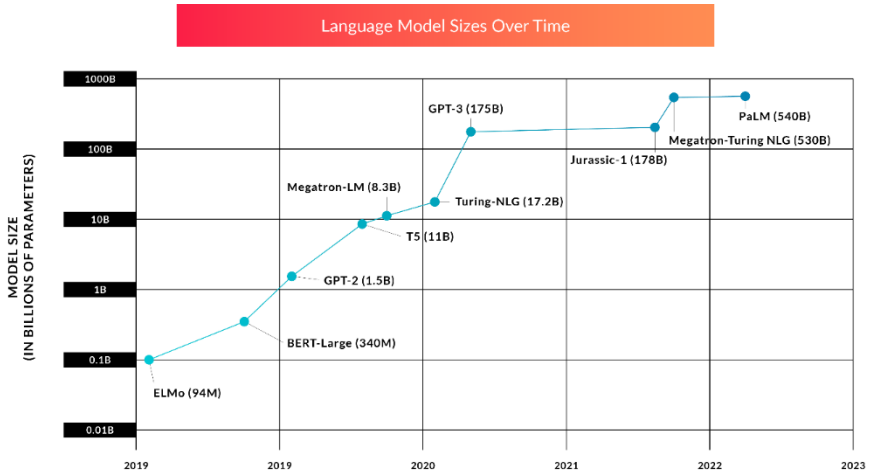
- Machine Learning (ML)
  - "the study of computer algorithms to improve automatically through experience and use of data"
- Deep Learning (DL) – a subset of ML
  - Uses Deep Neural Networks (DNNs)
  - **Perhaps, the most revolutionary subset!**
- Based on learning data representation
- DNN Examples: Convolutional Neural Networks, Recurrent Neural Networks, Hybrid Networks
- Data Scientist or Developer Perspective for using DNNs
  1. Identify DL as solution to a problem
  2. Determine Data Set
  3. Select Deep Learning Algorithm to Use
  4. Use a large data set to train an algorithm



Courtesy: https://hackernoon.com/difference-between-artificial-intelligence-machine-learning-and-deep-learning-1pcv3zeg, https://blog.dataiku.com/ai-vs.-machine-learning-vs.-deep-learning, https://en.wikipedia.org/wiki/Machine_learning

Evolution of Language Models



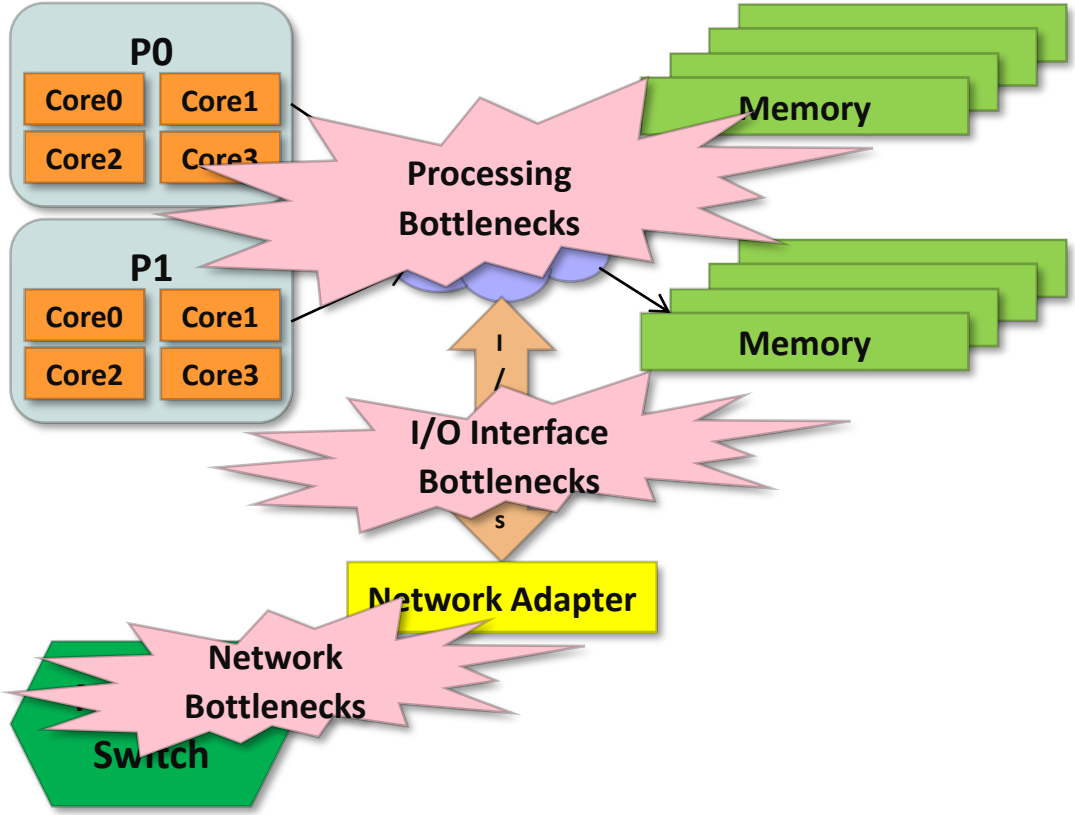


Courtesy: https://www.analyticsvidhya.com/blog/2023/07/build-your-own-large-language-models/
https://www.vinayiyengar.com/2022/08/04/the-promise-and-perils-of-large-language-models/

Networking and I/O Requirements

- Good System Area Networks with excellent performance (low latency, high bandwidth and low CPU utilization) for inter-processor communication (IPC) and I/O
- Good Storage Area Networks high performance I/O
- Good WAN connectivity in addition to intra-cluster SAN/LAN connectivity
- Quality of Service (QoS) for interactive applications
- RAS (Reliability, Availability, and Serviceability)
- With low cost

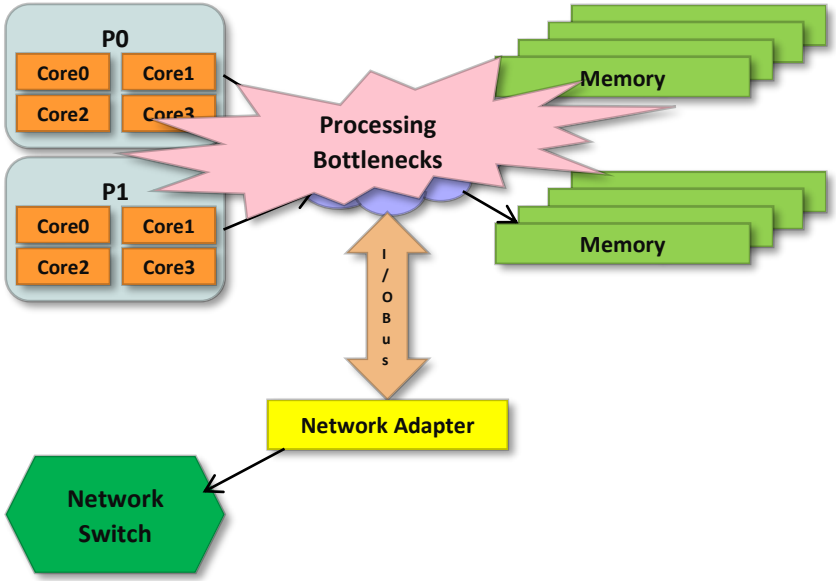
Major Components in Computing Systems



- Hardware components
  - Processing cores and memory subsystem
  - I/O bus or links
  - Network adapters/switches
- Software components
  - Communication stack
- *Bottlenecks can artificially limit the network performance the user perceives*

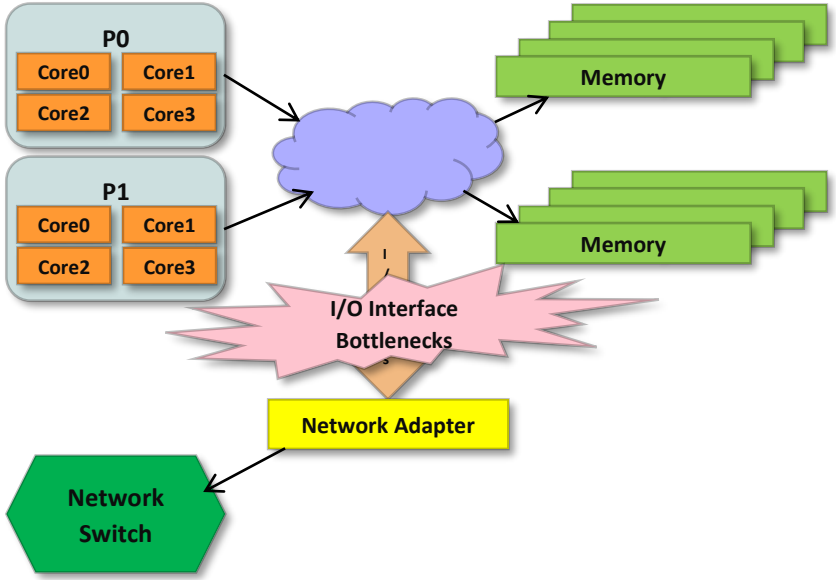
Processing Bottlenecks in Traditional Protocols

- Ex: TCP/IP, UDP/IP
- Generic architecture for all networks
- Host processor handles almost all aspects of communication
  - Data buffering (copies on sender and receiver)
  - Data integrity (checksum)
  - Routing aspects (IP routing)
- Signaling between different layers
  - Hardware interrupt on packet arrival or transmission
  - Software signals between different layers to handle protocol processing in different priority levels



Bottlenecks in Traditional I/O Interfaces and Networks

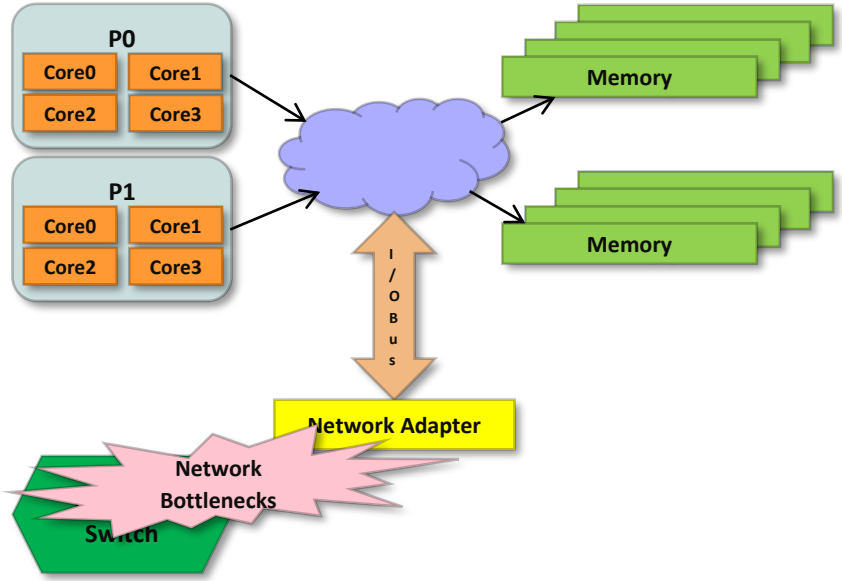
- Traditionally relied on bus-based technologies (last mile bottleneck)
  - E.g., PCI, PCI-X
  - One bit per wire
  - Performance increase through:
    - Increasing clock speed
    - Increasing bus width
  - Not scalable:
    - Cross talk between bits
    - Skew between wires
    - Signal integrity makes it difficult to increase bus width significantly, especially for high clock speeds



PCI	1990	33MHz/32bit: 1.05Gbps (shared bidirectional)
PCI-X	1998 (v1.0) 2003 (v2.0)	133MHz/64bit: 8.5Gbps (shared bidirectional) 266-533MHz/64bit: 17Gbps (shared bidirectional)

Bottlenecks on Traditional Networks

- Network speeds saturated at around 1Gbps
  - Features provided were limited
  - Commodity networks were not considered scalable enough for very large-scale systems



Ethernet (1979 -)	10 Mbit/sec
Fast Ethernet (1993 -)	100 Mbit/sec
Gigabit Ethernet (1995 -)	1000 Mbit /sec
ATM (1995 -)	155/622/1024 Mbit/sec
Myrinet (1993 -)	1 Gbit/sec
Fibre Channel (1994 -)	1 Gbit/sec

Motivation for High-Performance Networks

- Industry Networking Standards
- InfiniBand and High-speed Ethernet were introduced into the market to address these bottlenecks around 2000
- InfiniBand aimed at all three bottlenecks (protocol processing, I/O bus, and network speed)
- Ethernet aimed at directly handling the network speed bottleneck and relying on complementary technologies to alleviate the protocol processing and I/O bus bottlenecks

Presentation Overview

- Introduction
- **Why High-Performance Networking for HPC and AI?**
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
  - IB, HSE, their Convergence and Features
  - GPU-aware support in modern HPC networks:
    - NVLink and NVSwitch Interconnect Architecture
    - AMD Infinity Fabric Interconnect Architecture, UALink, & UltraEthernet
  - Amazon EFA Interconnect Architecture
  - Cray Slingshot Interconnect Architecture
- Overview of Emerging Smart Network Interfaces
  - Collectives w/ NVIDIA SHARP, NVIDIA BlueField DPUs, AMD Pensando Smart NICs, and Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

IB Trade Association

- IB Trade Association was formed with seven industry leaders (Compaq, Dell, HP, IBM, Intel, Microsoft, and Sun)
- Goal: To design a scalable and high performance communication and I/O architecture by taking an integrated view of computing, networking, and storage technologies
- Many other industry participated in the effort to define the IB architecture specification
- IB Architecture (Volume 1, Version 1.0) was released to public on Oct 24, 2000
  - Several annexes released after that (RDMA_CM - Sep'06, iSER – Sep'06, XRC – Mar'09, RoCE – Apr'10, RoCEv2 – Sep'14, Virtualization – Nov'16)
  - Latest version 1.8, released September 2024
- http://www.infinibandta.org

High-speed Ethernet Consortium

- 10GE Alliance formed by several industry leaders to take the Ethernet family to the next speed step
- Goal: To achieve a scalable and high performance communication architecture while maintaining backward compatibility with Ethernet
- There are products and standards for 10GE, 25GE, 40GE, 50GE, 100GE, 200GE, and 400 GE
- http://www.ethernetalliance.org
- 40-Gbps (Servers) and 100-Gbps Ethernet (Backbones, Switches, Routers): IEEE 802.3 WG
- 25-Gbps Ethernet Consortium targeting 25/50Gbps (July 2014)
  - http://25gethernet.org
- Energy-efficient and power-conscious protocols
  - On-the-fly link speed reduction for under-utilized links
- Ethernet Alliance Technology Forum looking forward to 2026
  - http://insidehpc.com/2016/08/at-ethernet-alliance-technology-forum/

Tackling Communication Bottlenecks with IB and HSE

- **Network speed bottlenecks**
- Protocol processing bottlenecks
- I/O interface bottlenecks

Network Bottleneck Alleviation: InfiniBand (“Infinite Bandwidth”) and High-speed Ethernet

- Bit serial differential signaling
  - Independent pairs of wires to transmit independent data (called a lane)
  - Scalable to any number of lanes
  - Easy to increase clock speed of lanes (since each lane consists only of a pair of wires)
- Theoretically, no perceived limit on the bandwidth

Network Speed Acceleration over the years

Ethernet (1979 -)	10 Mbit/sec
Fast Ethernet (1993 -)	100 Mbit/sec
Gigabit Ethernet (1995 -)	1000 Mbit /sec
ATM (1995 -)	155/622/1024 Mbit/sec
Myrinet (1993 -)	1 Gbit/sec
Fibre Channel (1994 -)	1 Gbit/sec
InfiniBand (2001 -)	2 Gbit/sec (1X SDR)
10-Gigabit Ethernet (2001 -)	10 Gbit/sec
InfiniBand (2003 -)	8 Gbit/sec (4X SDR)
InfiniBand (2005 -)	16 Gbit/sec (4X DDR)
	24 Gbit/sec (12X SDR)
InfiniBand (2007 -)	32 Gbit/sec (4X QDR)
40-Gigabit Ethernet (2010 -)	40 Gbit/sec
InfiniBand (2011 -)	54.6 Gbit/sec (4X FDR)
InfiniBand (2012 -)	2 x 54.6 Gbit/sec (4X Dual-FDR)
25-/50-Gigabit Ethernet (2014 -)	25/50 Gbit/sec
100-Gigabit Ethernet (2015 -)	100 Gbit/sec
Omni-Path (2015 -)	100 Gbit/sec
InfiniBand (2015 -)	100 Gbit/sec (4X EDR)
InfiniBand (2017 -)	200 Gbit/sec (4X HDR)
Slingshot10/11 (2021 -)	200 Gbit/sec
Omni-Path-Express (2021 -)	100 Gbit/sec
Google Aquila (2021 -)	100 Gbit/sec
InfiniBand (2022 -)	400 Gbit/sec (4X NDR)
Omni-Path-Express (2024 -)	400 Gbit/sec (CN5000)

200 times in the last 24 years!!

Tackling Communication Bottlenecks with IB and HSE

- Network speed bottlenecks
- **Protocol processing bottlenecks**
- I/O interface bottlenecks

Capabilities of High-Performance Networks

- Intelligent Network Interface Cards
- Support entire protocol processing completely in hardware (hardware protocol offload engines)
- Provide a rich communication interface to applications
  - ***User-level communication capability***
  - Gets rid of intermediate data buffering requirements
- No software signaling between communication layers
  - All layers are implemented on a ***dedicated*** hardware unit, and not on a ***shared*** host CPU

Tackling Communication Bottlenecks with IB and HSE

- Network speed bottlenecks
- Protocol processing bottlenecks
- **I/O interface bottlenecks**

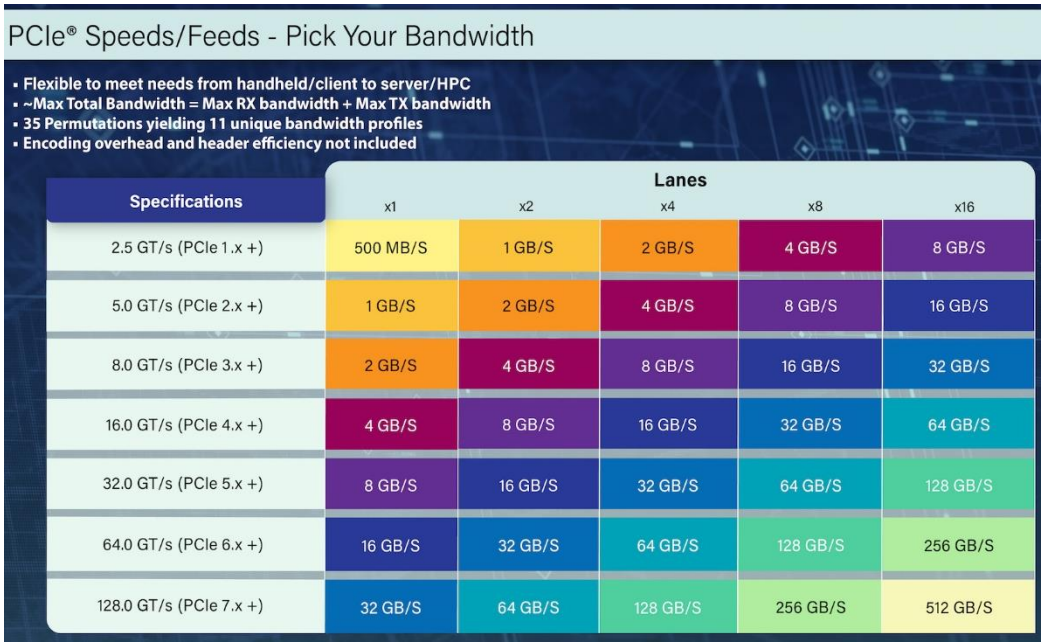
Interplay with I/O Technologies

- InfiniBand initially intended to replace I/O bus technologies with networking-like technology
  - That is, bit serial differential signaling
  - With enhancements in I/O technologies that use a similar architecture (HyperTransport, PCI Express), this has become mostly irrelevant now
- Both IB and HSE today come as network adapters that plug into existing I/O technologies

Trends in I/O Interfaces with Servers

- Recent trends in I/O interfaces show that they are nearly matching head-to-head with network speeds (though they still lag a little bit)

PCI	1990	33MHz/32bit: 1.05Gbps (shared bidirectional)
PCI-X	1998 (v1.0) 2003 (v2.0)	133MHz/64bit: 8.5Gbps (shared bidirectional) 266-533MHz/64bit: 17Gbps (shared bidirectional)
AMD HyperTransport (HT)	2001 (v1.0), 2004 (v2.0) 2006 (v3.0), 2008 (v3.1)	102.4Gbps (v1.0), 179.2Gbps (v2.0) 332.8Gbps (v3.0), 409.6Gbps (v3.1) (32 lanes)
Intel QuickPath Interconnect (QPI)	2009	153.6-204.8Gbps (20 lanes)



* https://insidehpc.com/2018/06/implementing-pcie-gen-4-expansion/
+ https://insidehpc.com/2019/08/video-pci-express-6-0-specification-to-reach-64-gigatransfers-sec/
https://arstechnica.com/gadgets/2022/06/months-after-finalizing-pcie-6-0-pci-sig-looks-to-double-speeds-again-with-pcie-7-0/

Upcoming I/O Interface Architectures

- Cache Coherence Interconnect for Accelerators (CCIX)
  - https://www.ccixconsortium.com/
- NVLink
  - http://www.nvidia.com/object/nvlink.html
- CAPI/OpenCAPI
  - http://opencapi.org/
- GenZ
  - http://genzconsortium.org/

Compute eXpress Link (CXL)

- Open industry standard
- Provides a cache coherent interconnect between
  - CPUs
  - Accelerators, like GPUs
  - Smart I/O devices, like DPUs, and
  - Various flavors of DDR4/DDR5 and persistent memories
- Allows the CPU to work on the same memory regions as the connected devices
- Improving performance and power efficiency while reducing software complexity and data movement

Courtesy: Toms Hardware & CXL Consortium

CXL 3.1 Spec Feature Comparison

Courtesy: Toms Hardware & CXL Consortium

CXL Specification Feature Summary

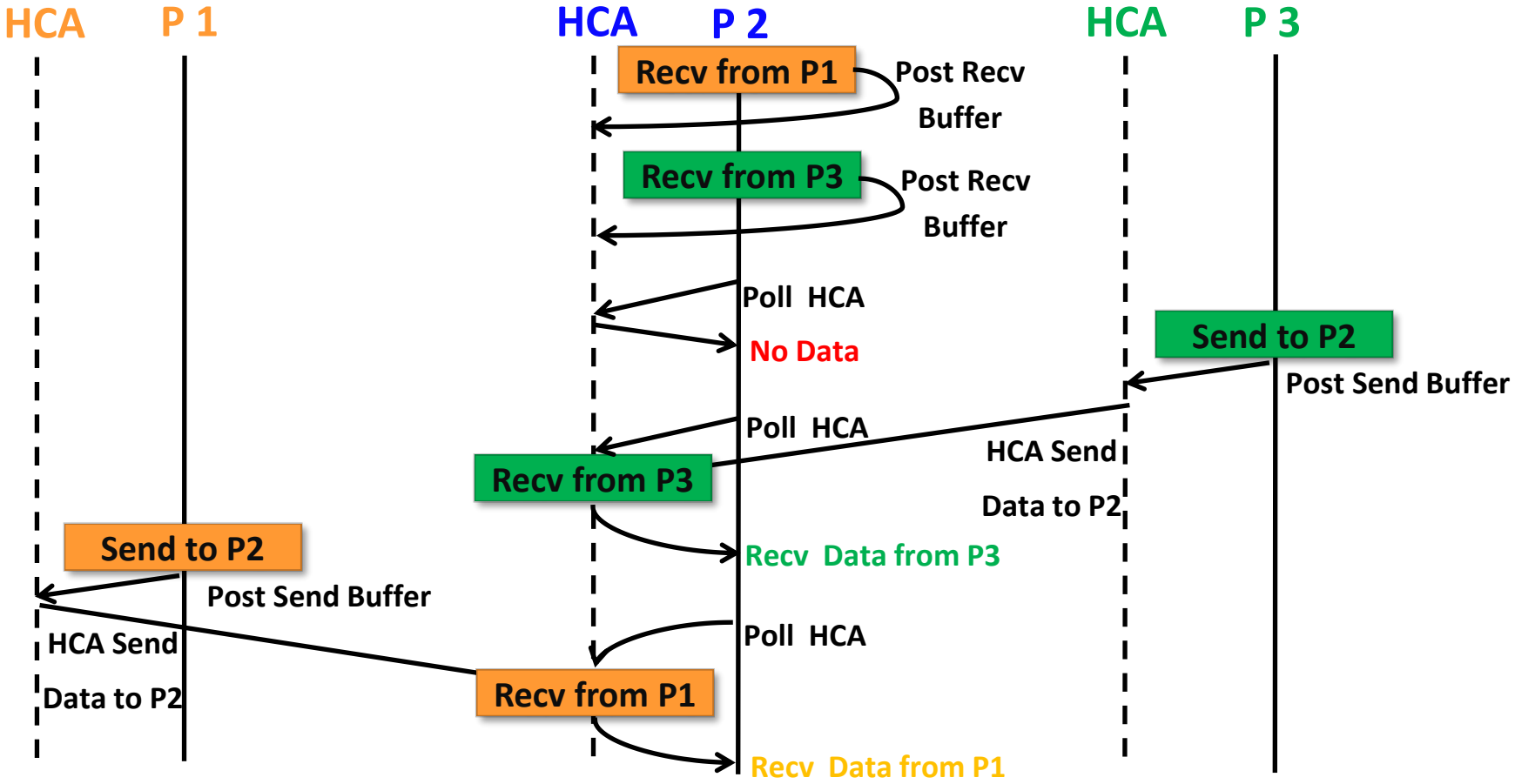
Features	CXL 1.0 / 1.1	CXL 2.0	CXL 3.0	CXL 3.1
Release date	2019	2020	August 2022	November 2023
Max link rate	32GTs	32GTs	64GTs	64GTs
Flit 68 byte (up to 32 GTs)	✓	✓	✓	✓
Flit 256 byte (up to 64 GTs)			✓	✓
Type 1, Type 2 and Type 3 Devices	✓	✓	✓	✓
Memory Pooling w/ MLDs		✓	✓	✓
Global Persistent Flush		✓	✓	✓
CXL IDE		✓	✓	✓
Switching (Single-level)		✓	✓	✓
Switching (Multi-level)			✓	✓
Direct memory access for peer-to-peer			✓	✓
Enhanced coherency (256 byte flit)			✓	✓
Memory sharing (256 byte flit)			✓	✓
Multiple Type 1/Type 2 devices per root port			✓	✓
Fabric capabilities (256 byte flit)			✓	✓
Fabric Manager API definition for PBR Switch				✓
Host-to-Host communication with Global Integrated Memory (GIM) concept				✓
Trusted-Execution-Environment (TEE) Security Protocol				✓
Memory expander enhancements (up to 34-bit of meta data, RAS capability enhancements)				✓

Not Supported ✓ Supported

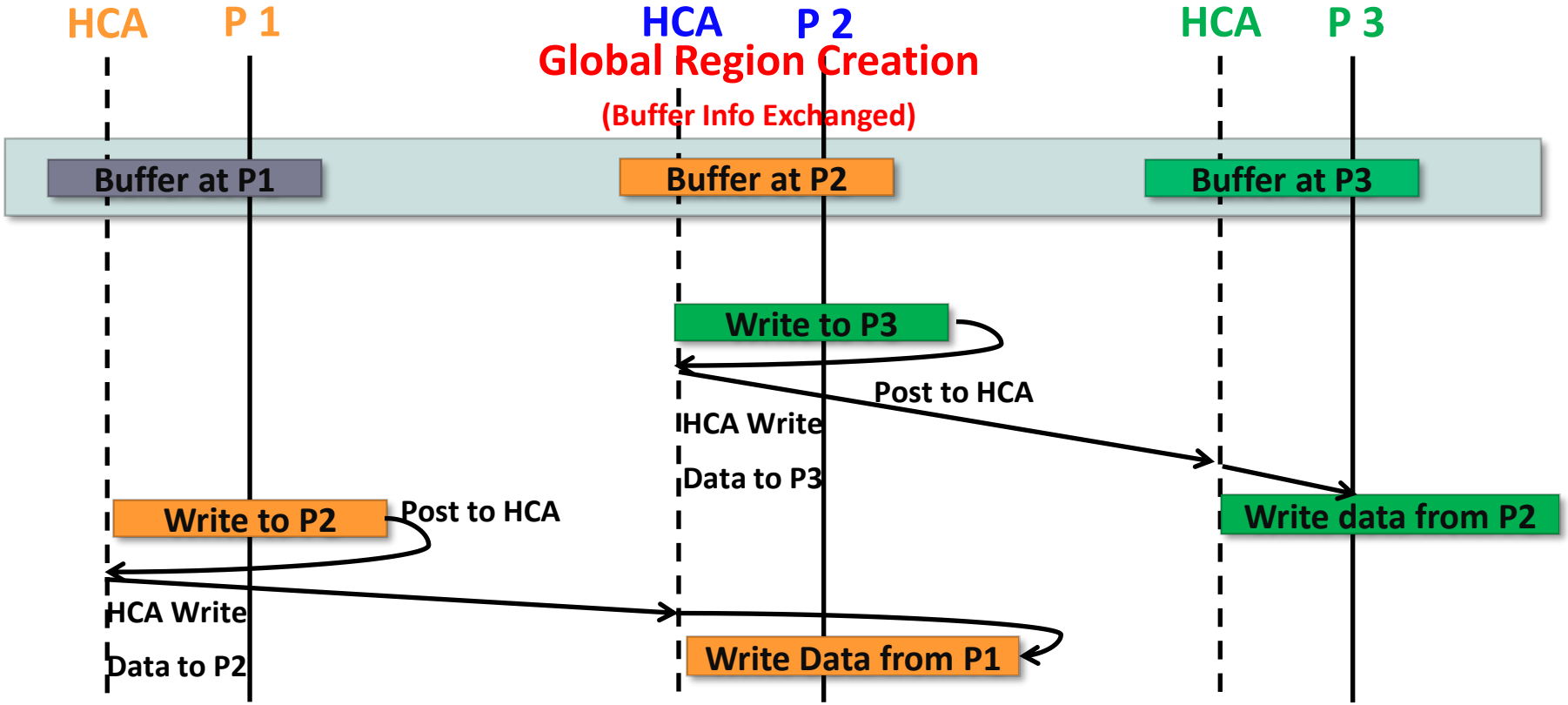
Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- **Communication Model and Semantics of High-Performance Networks**
- Architectural Overview of High-Performance Networks
  - IB, HSE, their Convergence and Features
  - GPU-aware support in modern HPC networks:
    - NVLink and NVSwitch Interconnect Architecture
    - AMD Infinity Fabric Interconnect Architecture, UALink, & UltraEthernet
  - Amazon EFA Interconnect Architecture
  - Cray Slingshot Interconnect Architecture
- Overview of Emerging Smart Network Interfaces
  - NVIDIA BlueField DPUs, AMD Pensando Smart NICs, and Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

Two-sided Communication Model

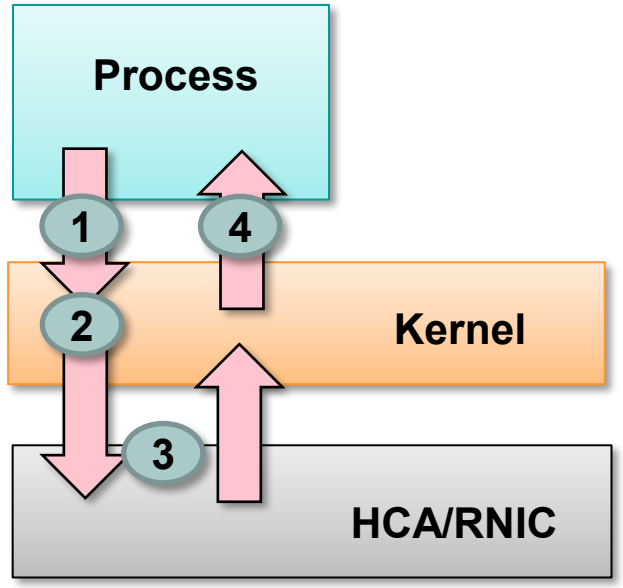


One-sided Communication Model



Memory Registration

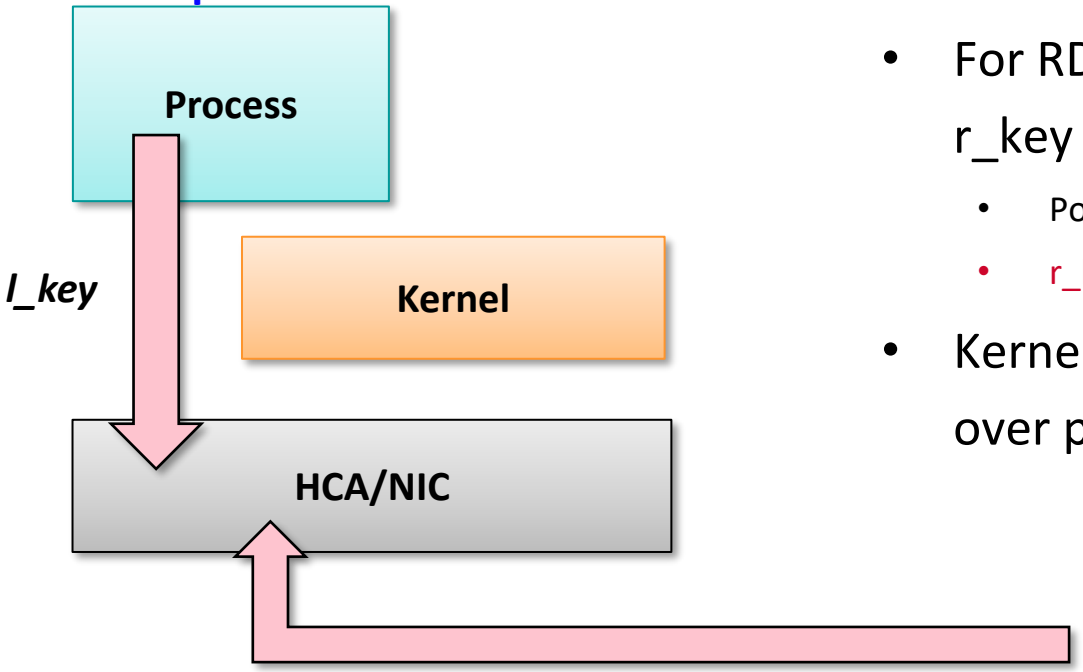
Before we do any communication:
All memory used for communication must be registered



1. Registration Request
   - Send virtual address and length
2. Kernel handles virtual->physical mapping and pins region into physical memory
   - Process cannot map memory that it does not own (security !)
3. HCA caches the virtual to physical mapping and issues a handle
   - Includes an *l_key* and *r_key*
4. Handle is returned to application

Memory Protection

For security, keys are required for all operations that touch buffers

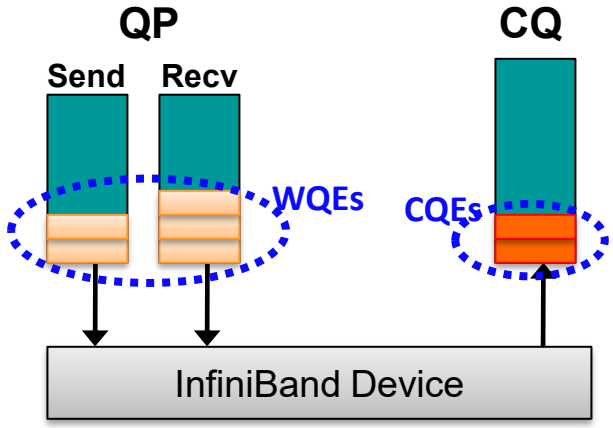


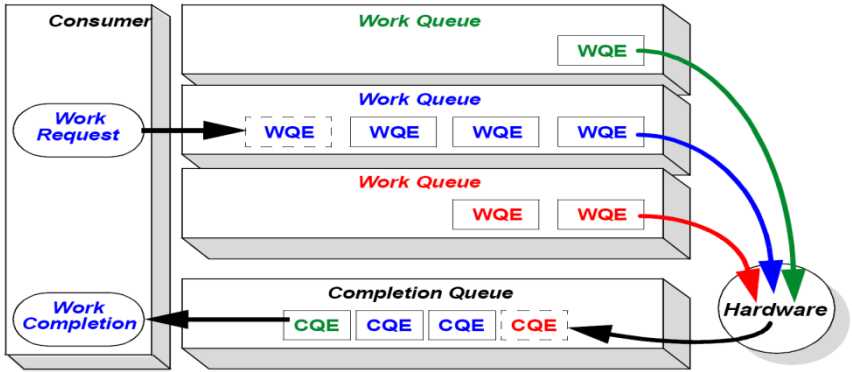
- To send or receive data the *l_key* must be provided to the HCA
  - HCA verifies access to local memory
- For RDMA, initiator must have the r_key for the remote virtual address
  - Possibly exchanged with a send/recv
  - r_key is not encrypted in IB
- Kernel bypass grants improved latency over prior transfer mechanisms

Critical to Latency Reduction

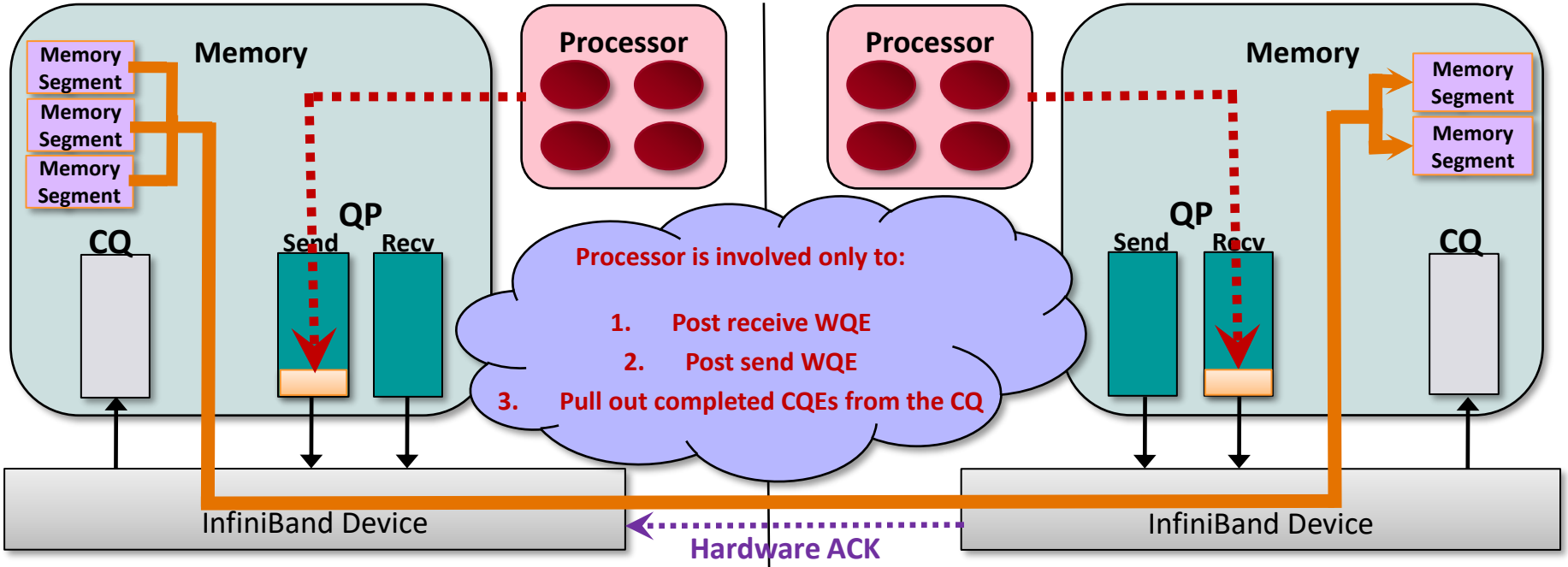
Queue Pair Model

- Each QP has two queues
  - Send Queue (SQ)
  - Receive Queue (RQ)
  - Work requests are queued to the QP (WQEs: "Wookies")
- QP to be linked to a Complete Queue (CQ)
  - Gives notification of operation completion from QPs
  - Completed WQEs are placed in the CQ with additional information (CQEs: "Cookies")





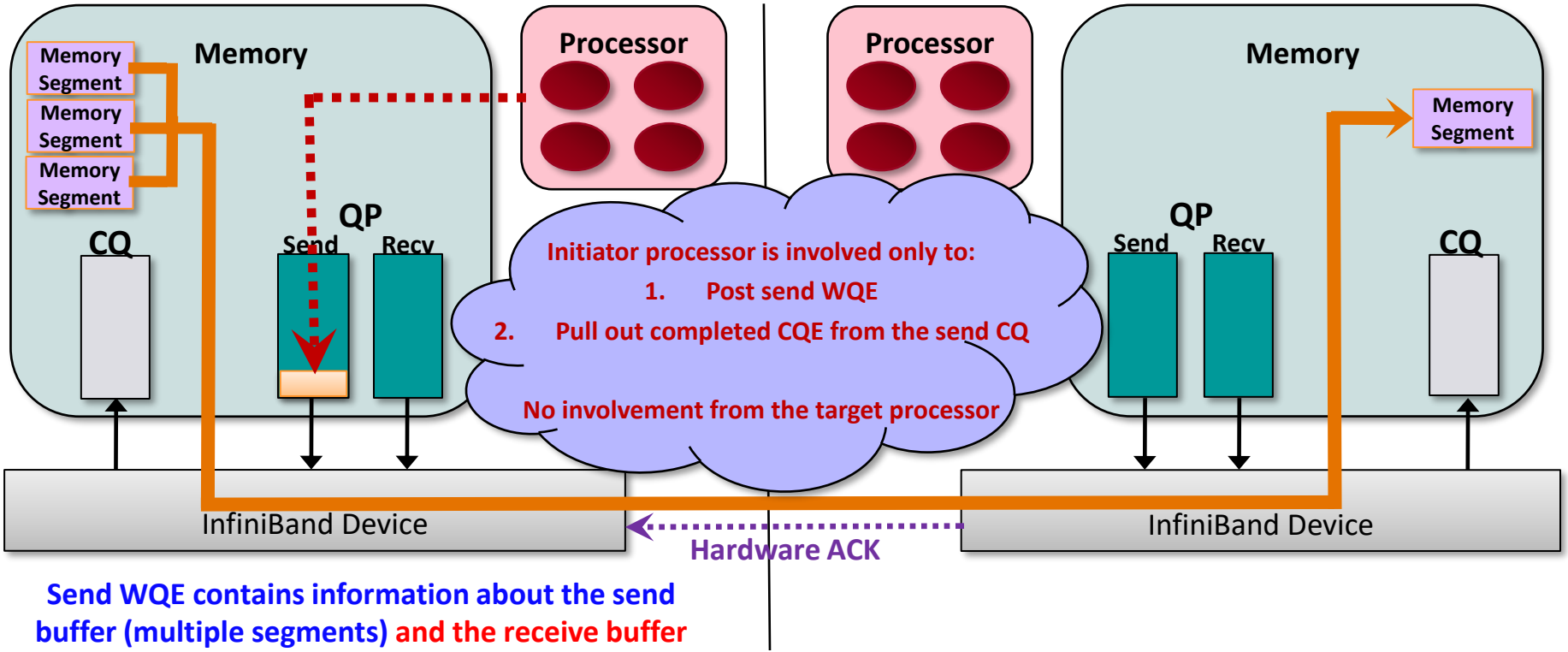
Communication in the Channel Semantics (Send/Receive Model)



Send WQE contains information about the send buffer (multiple non-contiguous segments)

Receive WQE contains information on the receive buffer (multiple non-contiguous segments); Incoming messages have to be matched to a receive WQE to know where to place

Communication in the Memory Semantics (RDMA Model)



Send WQE contains information about the send buffer (multiple segments) and the receive buffer (single segment)

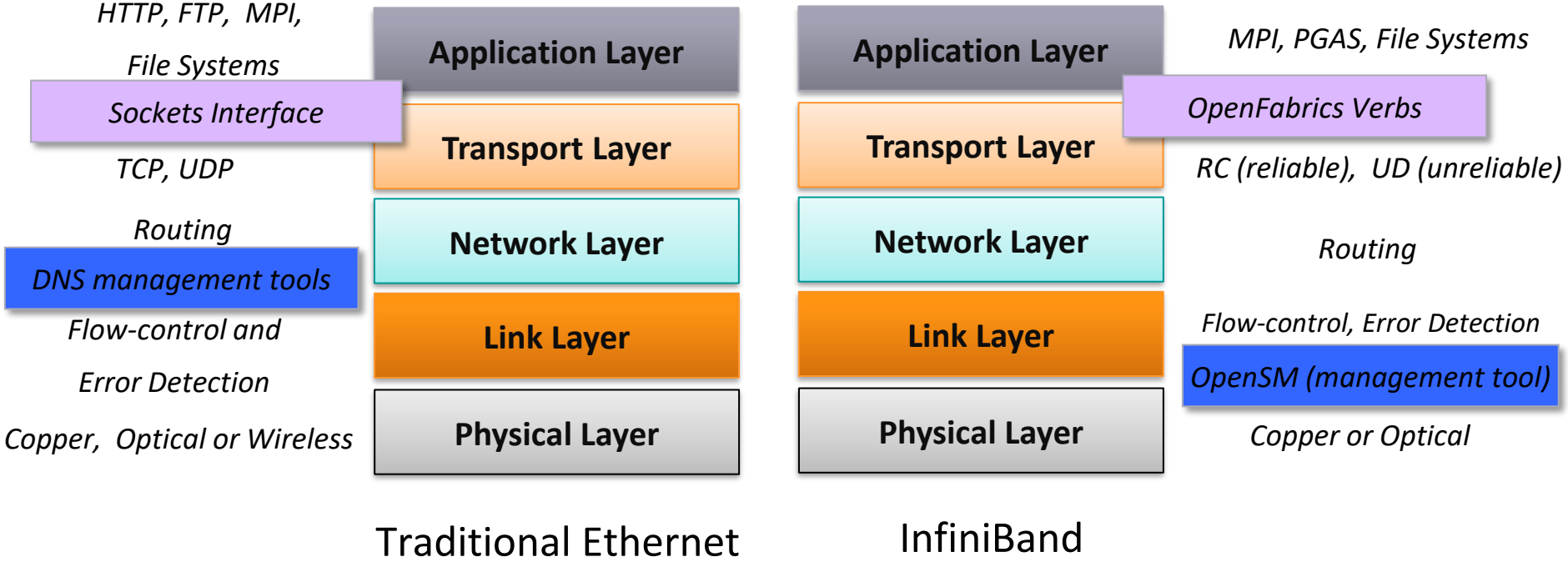
Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- **Architectural Overview of High-Performance Networks**
  - **IB, HSE, their Convergence and Features**
  - GPU-aware support in modern HPC networks:
    - NVLink and NVSwitch Interconnect Architecture
    - AMD Infinity Fabric Interconnect Architecture, UALink, & UltraEthernet
  - Amazon EFA Interconnect Architecture
  - Cray Slingshot Interconnect Architecture
- Overview of Emerging Smart Network Interfaces
  - NVIDIA BlueField DPUs, AMD Pensando Smart NICs, and Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

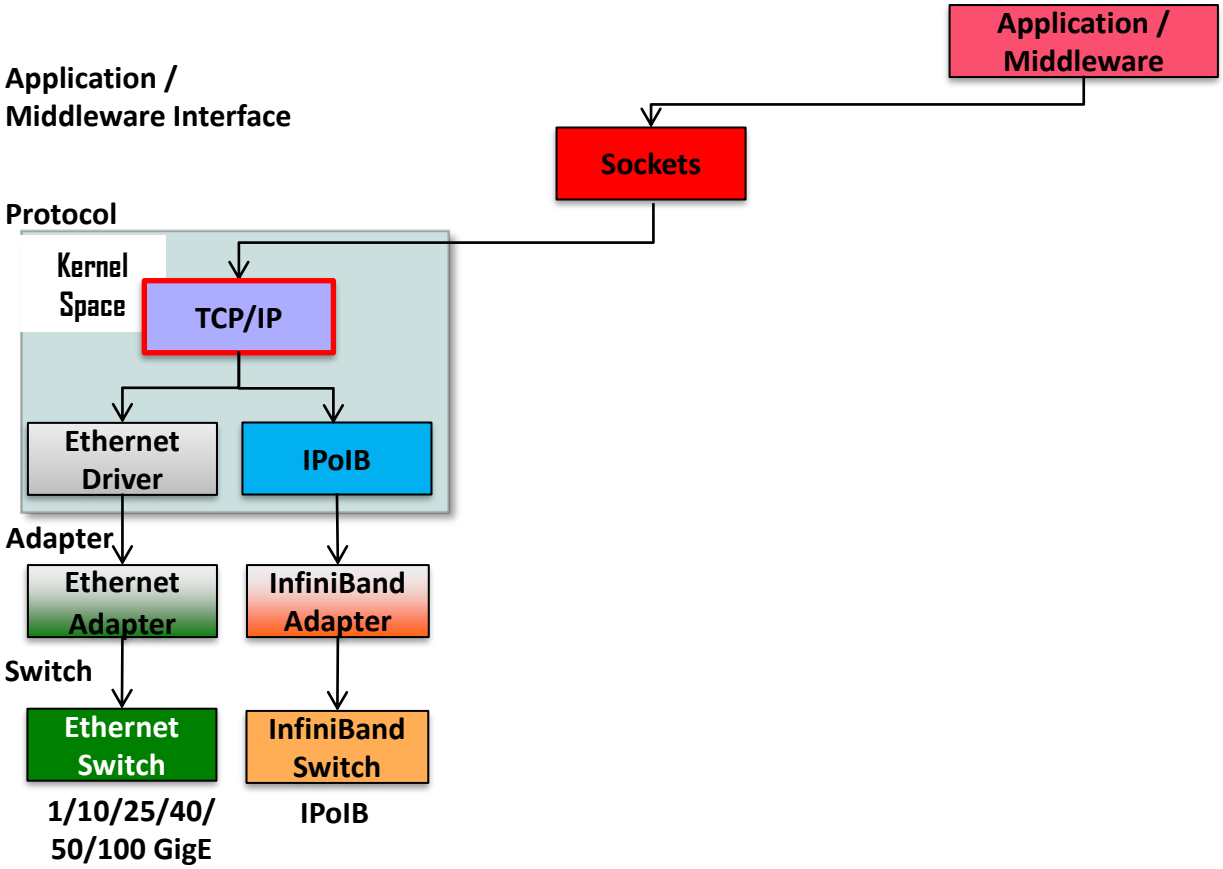
IB, HSE and their Convergence

- **InfiniBand**
  - **Architecture and Basic Hardware Components**
  - Hardware Protocol Offload
- High-speed Ethernet Family
  - Internet Wide Area RDMA Protocol (iWARP)
- InfiniBand/Ethernet Convergence Technologies
  - (InfiniBand) RDMA over Converged (Enhanced) Ethernet (RoCE)

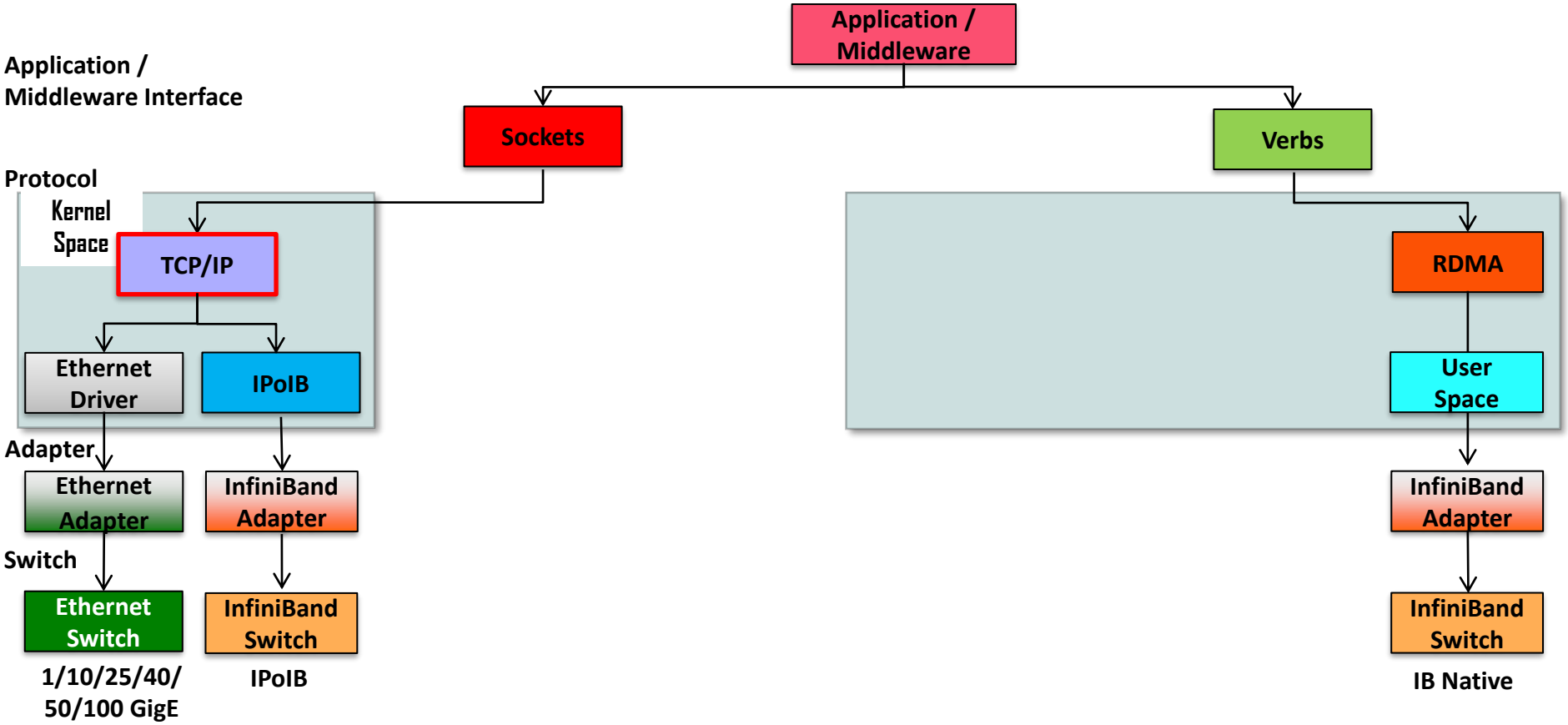
Comparing InfiniBand with Traditional Networking Stack



TCP/IP Stack and IPoIB



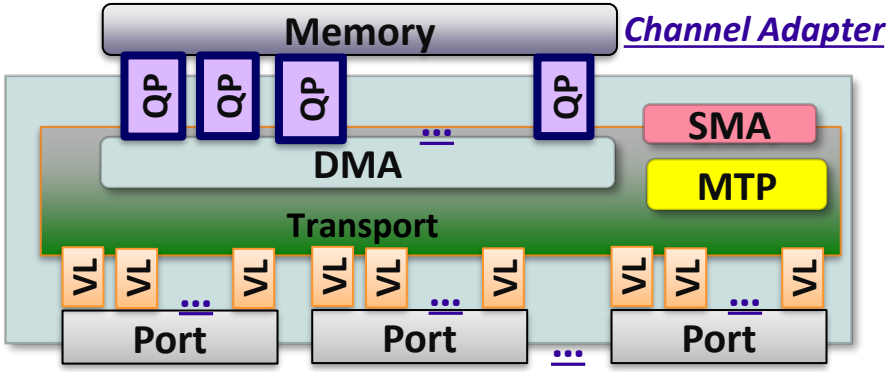
TCP/IP, IPoIB and Native IB Verbs

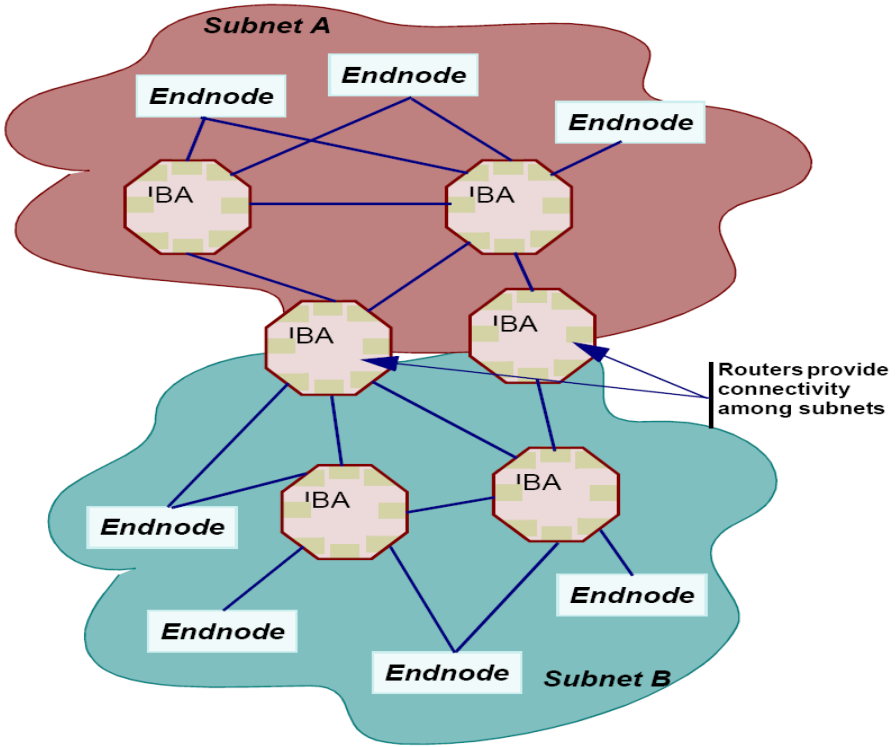


IB, HSE and their Convergence

- **InfiniBand**
  - **Architecture and Basic Hardware Components**
  - Hardware Protocol Offload
- High-speed Ethernet Family
  - Internet Wide Area RDMA Protocol (iWARP)
- InfiniBand/Ethernet Convergence Technologies
  - (InfiniBand) RDMA over Converged (Enhanced) Ethernet (RoCE)

Components: Channel Adapters, Switches/Routers, and Links



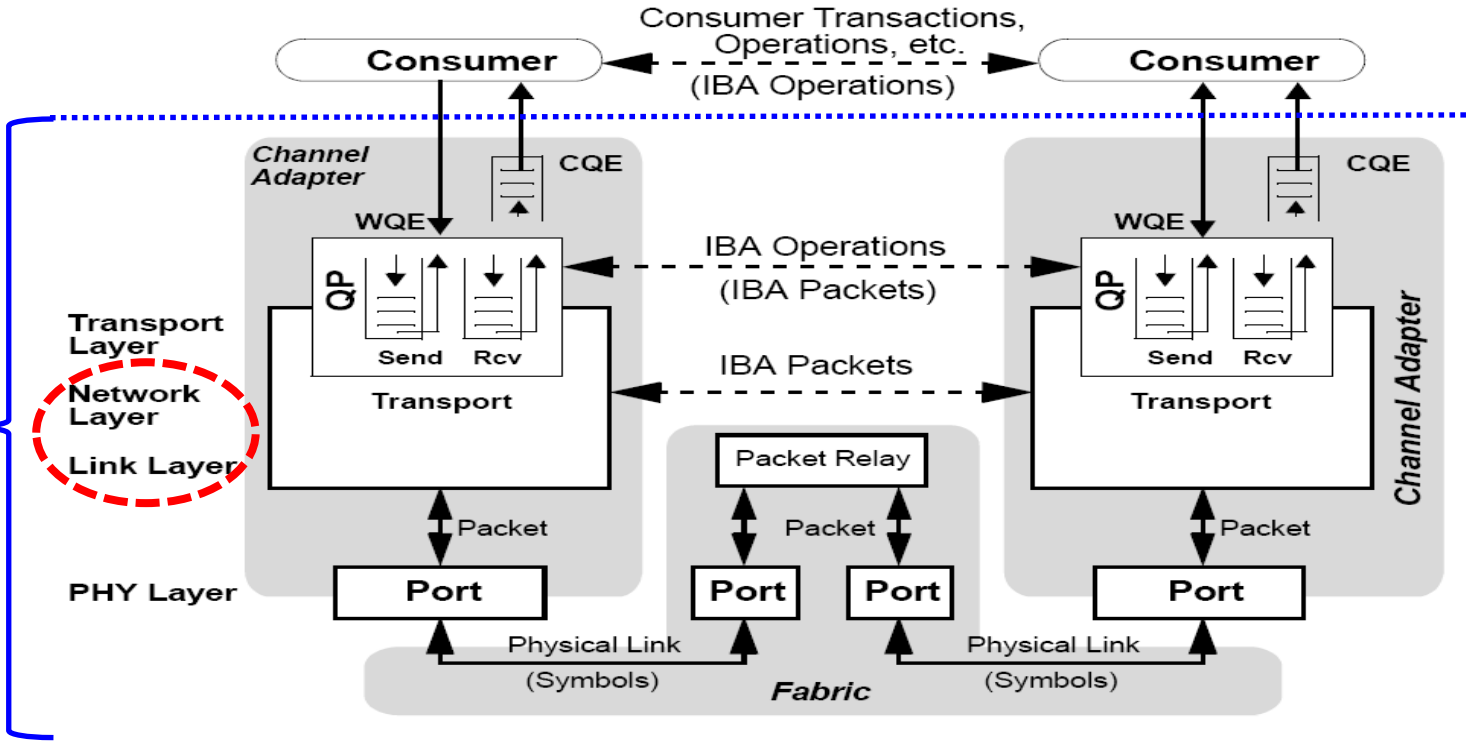


IB, HSE and their Convergence

- **InfiniBand**
  - Architecture and Basic Hardware Components
  - **Hardware Protocol Offload**
- High-speed Ethernet Family
  - Internet Wide Area RDMA Protocol (iWARP)
- InfiniBand/Ethernet Convergence Technologies
  - (InfiniBand) RDMA over Converged (Enhanced) Ethernet (RoCE)

Hardware Protocol Offload

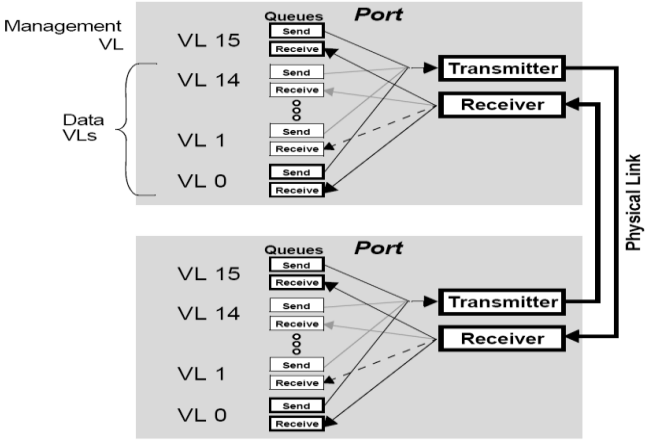
Complete Hardware Implementations Exist

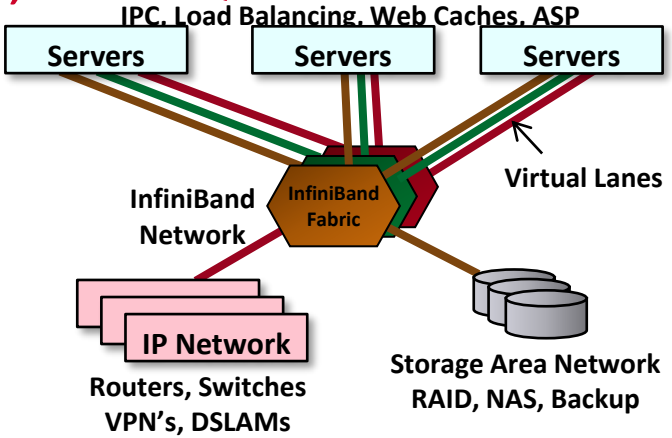


Link/Network Layer Capabilities

- Buffering and Flow Control
- Virtual Lanes, Service Levels, and QoS
- Switching and Multicast

Virtual Lanes, Service Levels, and QoS





(Courtesy: Mellanox Technologies)

Traffic Segregation

- Virtual Lanes (VL)
  - Multiple (between 2 and 16) virtual links within same physical link
    - 0 – default data VL; 15 – VL for management traffic
  - Separate buffers and flow control
  - Avoids Head-of-Line Blocking
- Service Level (SL):
  - Packets may operate at one of 16, user defined SLs
- SL to VL mapping:
  - SL determines which VL on the next link is to be used
  - Each port (switches, routers, end nodes) has a SL to VL mapping table configured by the subnet management
- Partitions:
  - Fabric administration (through Subnet Manager) may assign specific SLs to different partitions to isolate traffic flows

Switching (Layer-2 Routing) and Multicast

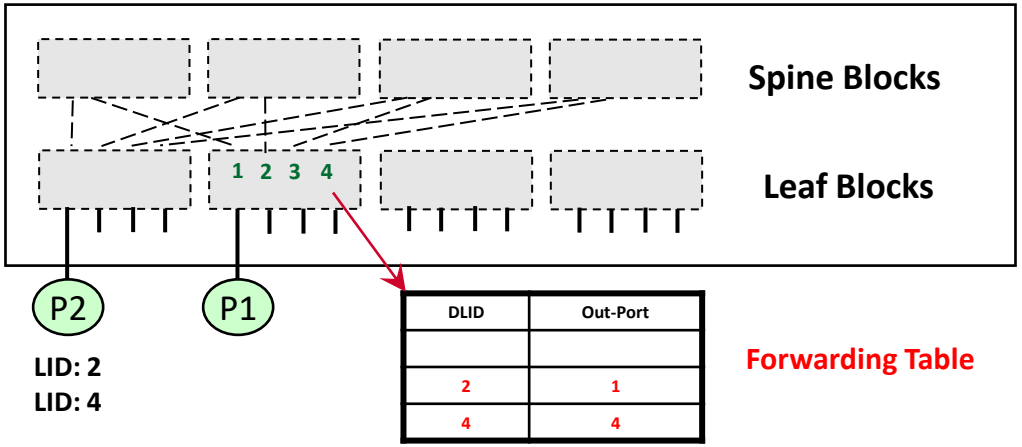
- Each port has one or more associated LIDs (Local Identifiers)
  - Switches look up which port to forward a packet to based on its destination LID (DLID)
  - This information is maintained at the switch
- For multicast packets, the switch needs to maintain multiple output ports to forward the packet to
  - Packet is replicated to each appropriate output port
  - Ensures at-most once delivery & loop-free forwarding
  - There is an interface for a group management protocol
    - Create, join/leave, prune, delete group

Switch Complex

- Basic unit of switching is a crossbar
  - Current InfiniBand products use either 24-port (DDR), 36-port (QDR and FDR), and 48-port (EDR) crossbars
- Switches available in the market are typically collections of crossbars within a single cabinet
- Do not confuse “non-blocking switches” with “crossbars”
  - Crossbars provide all-to-all connectivity to all connected nodes
    - *For any random node pair selection, all communication is non-blocking*
  - Non-blocking switches provide a fat-tree of many crossbars
    - *For any random node pair selection, there exists a switch configuration such that communication is non-blocking*
    - *If the communication pattern changes, the same switch configuration might no longer provide fully non-blocking communication*

IB Switching/Routing: An Example

An Example IB Switch Block Diagram (Mellanox 144-Port)



- Someone has to setup the forwarding tables and give every port an LID
  - "Subnet Manager" does this work
- Different routing algorithms give different paths

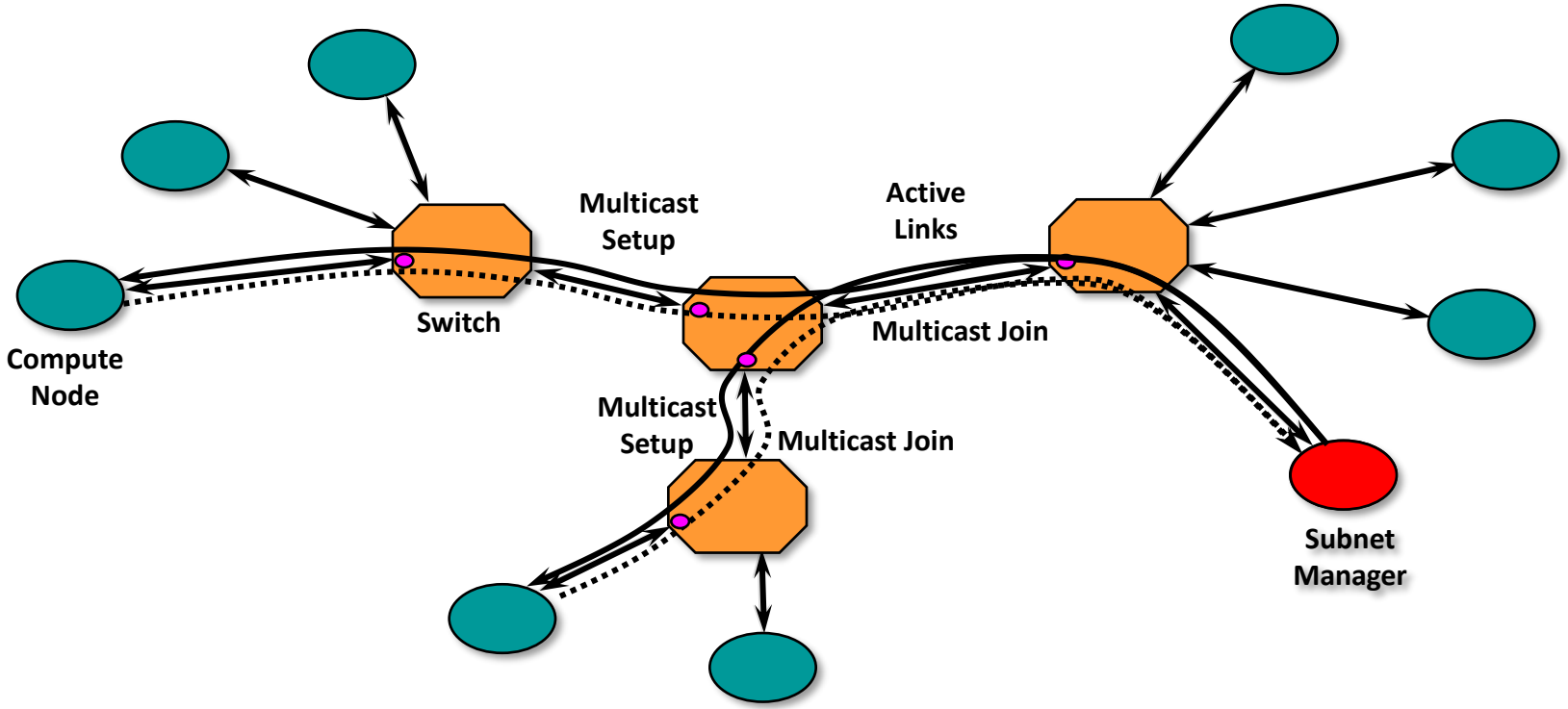
Switching: IB supports Virtual Cut Through (VCT)

Routing: Unspecified by IB SPEC

Up*/Down*, Shift are popular routing engines supported by OFED

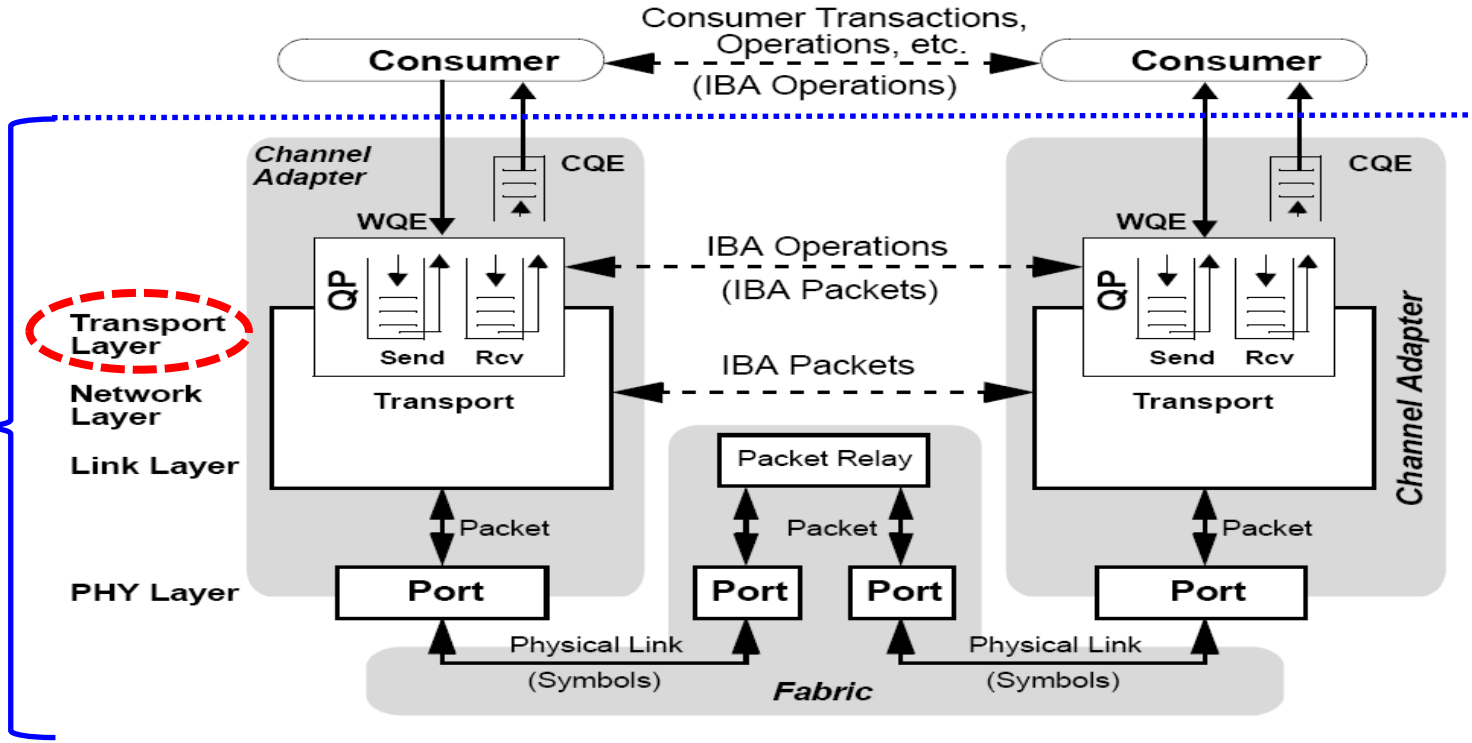
- Fat-Tree is a popular topology for IB Cluster
  - Different over-subscription ratio may be used
- Other topologies
  - 3D Torus (Sandia Red Sky, SDSC Gordon) and SGI Altix (Hypercube)
  - 10D Hypercube (NASA Pleiades)

IB Multicast Example



Hardware Protocol Offload

Complete Hardware Implementations Exist



IB Transport Types and Associated Trade-offs

Attribute		Reliable Connection	Reliable Datagram	Dynamic Connected	eXtended Reliable Connection	Unreliable Connection	Unreliable Datagram	Raw Datagram
Scalability (M processes, N nodes)		**M^2N** QPs per HCA	**M** QPs per HCA	**M** QPs per HCA	**MN** QPs per HCA	**M^2N** QPs per HCA	**M** QPs per HCA	**1** QP per HCA
Reliability	**Corrupt data detected**	**Yes**						
	Data Delivery Guarantee	**Data delivered exactly once**				**No guarantees**		
	Data Order Guarantees	**Per connection**	**One source to multiple destinations**	**Per connection**	**Per connection**	**Unordered, duplicate data detected**	**No**	**No**
	Data Loss Detected	**Yes**					**No**	**No**
	Error Recovery	**Errors (retransmissions, alternate path, etc.) handled by transport layer. Client only involved in handling fatal errors (links broken, protection violation, etc.)**				**Packets with errors and sequence errors are reported to responder**	**None**	**None**

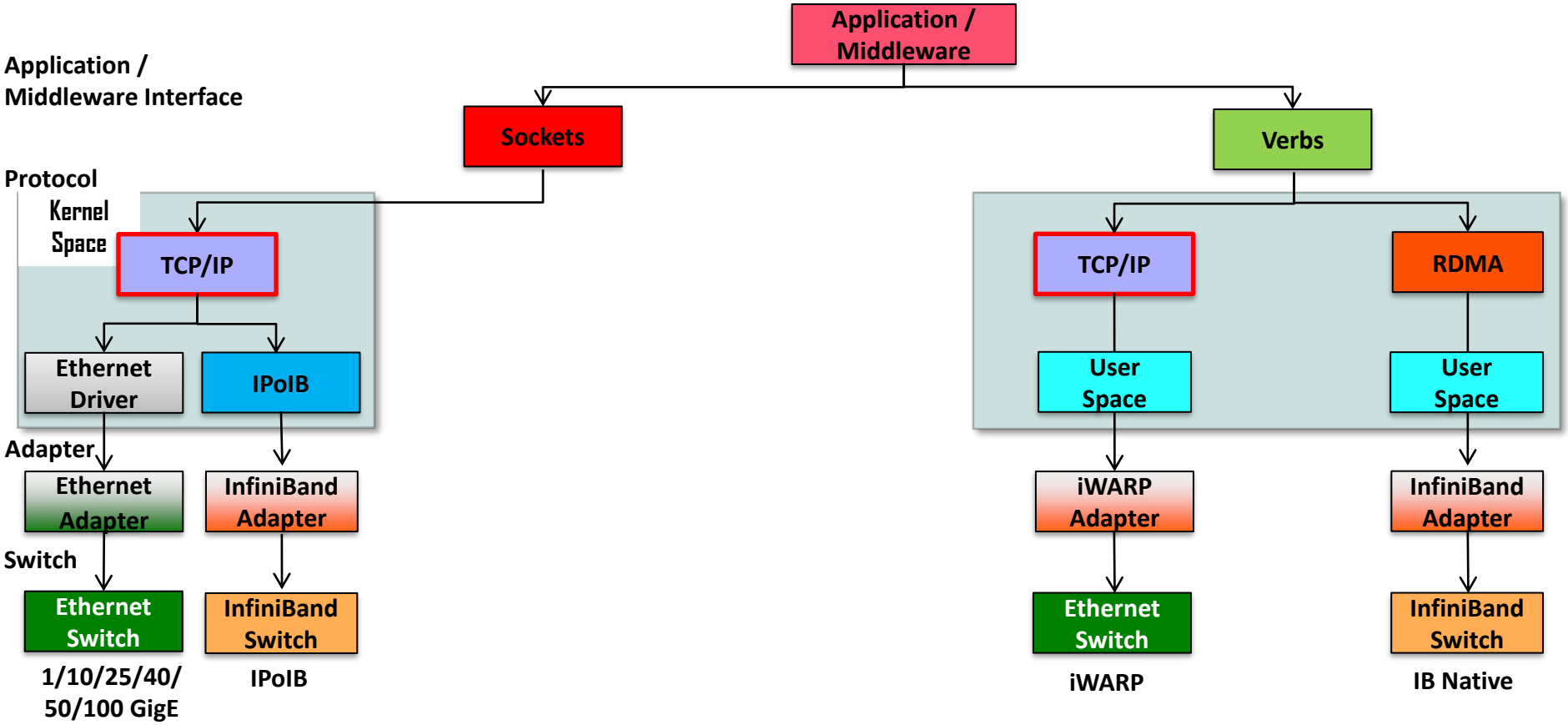
IB, HSE and their Convergence

- **InfiniBand**
  - Architecture and Basic Hardware Components
  - Hardware Protocol Offload
- **High-speed Ethernet Family**
  - **Internet Wide Area RDMA Protocol (iWARP)**
- InfiniBand/Ethernet Convergence Technologies
  - (InfiniBand) RDMA over Converged (Enhanced) Ethernet (RoCE)

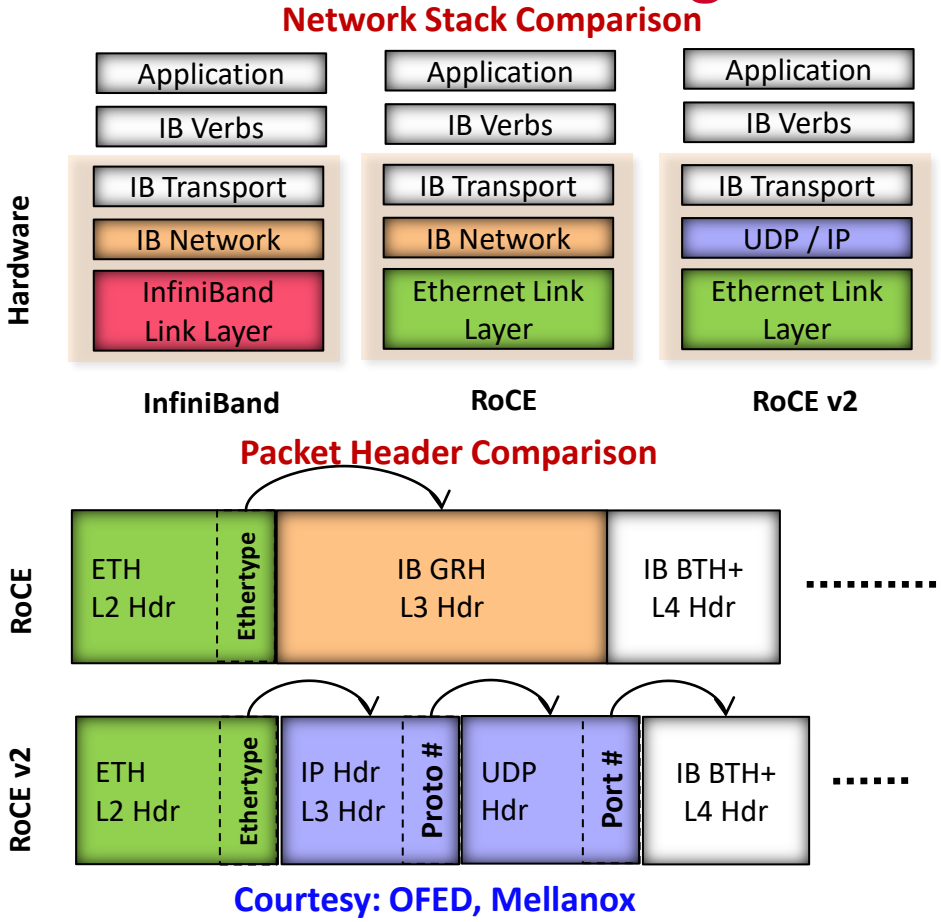
IB and 10/40GE RDMA Models: Commonalities and Differences

Features	IB	iWARP/HSE
Hardware Acceleration	Supported	Supported
RDMA	Supported	Supported
Atomic Operations	Supported	Not supported
Multicast	Supported	Supported
Congestion Control	Supported	Supported
Data Placement	Ordered	Out-of-order
Data Rate-control	Static and Coarse-grained	Dynamic and Fine-grained
QoS	Prioritization	Prioritization and Fixed Bandwidth QoS
Multipathing	Using DLIDs	Using VLANs

iWARP and TOE

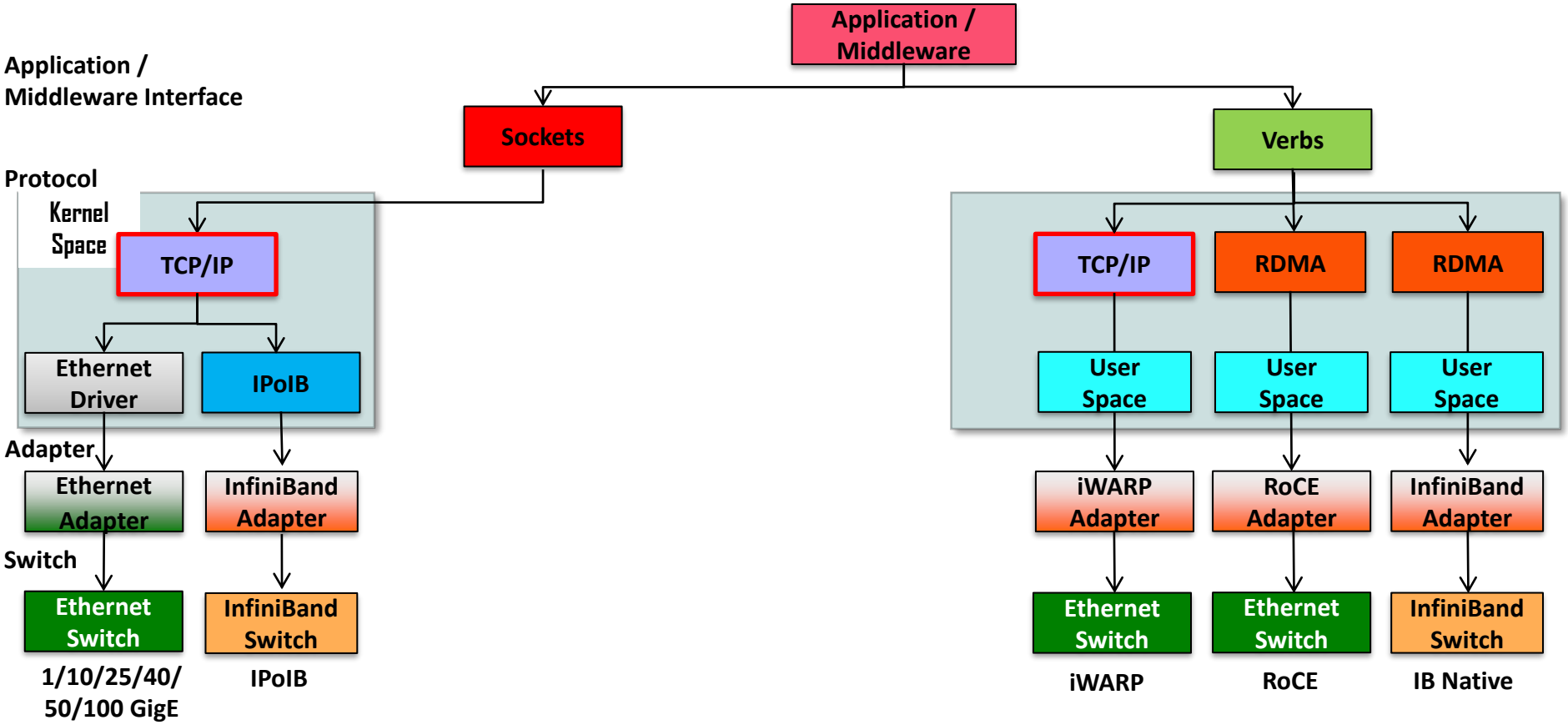


RDMA over Converged Enhanced Ethernet (RoCE)



- Takes advantage of IB and Ethernet
  - Software written with IB-Verbs
  - Link layer is Converged (Enhanced) Ethernet (CE)
  - 100Gb/s support from latest EDR and ConnectX-3 Pro adapters
- Pros: IB Vs RoCE
  - Works natively in Ethernet environments
    - Entire Ethernet management ecosystem is available
  - Has all the benefits of IB verbs
  - Link layer is very similar to the link layer of native IB, so there are no missing features
- RoCE v2: Additional Benefits over RoCE
  - Traditional Network Management Tools Apply
  - ACLs (Metering, Accounting, Firewalling)
  - GMP Snooping for Optimized Multicast
  - Network Monitoring Tools

RDMA over Converged Ethernet (RoCE)

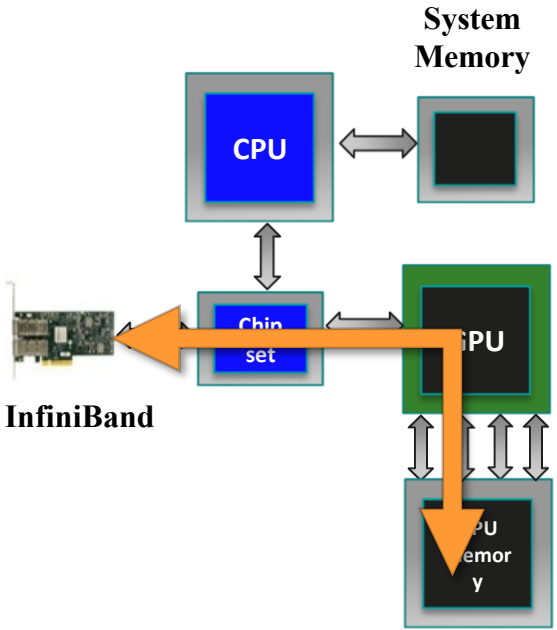


Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- **Architectural Overview of High-Performance Networks**
  - IB, HSE, their Convergence and Features
  - **GPU-aware support in modern HPC networks:**
    - **NVLink and NVSwitch Interconnect Architecture**
    - **AMD Infinity Fabric Interconnect Architecture, UALink, & UltraEthernet**
  - Amazon EFA Interconnect Architecture
  - Cray Slingshot Interconnect Architecture
- Overview of Emerging Smart Network Interfaces
  - NVIDIA BlueField DPUs, AMD Pensando Smart NICs, and Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

GPU-Direct RDMA

- Fastest possible communication between GPU and other PCI-E devices
- Network adapter can directly read/write data from/to GPU device memory
- Avoids copies through the host
- Allows for better asynchronous communication
- Project done jointly between OSU, Mellanox, and NVIDIA during 2011-15. (ISC '11 paper on CUDA-Aware MPI)
- Very widely used in current days HPC and AI middleware currently for all GPU-based systems (NVIDIA, AMD, and Intel) with different interconnects (InfiniBand, Slingshot, Omni-Path, ROCE, etc.)



GPU-Aware (CUDA-Aware) MPI Library: MVAPICH2-GPU

- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

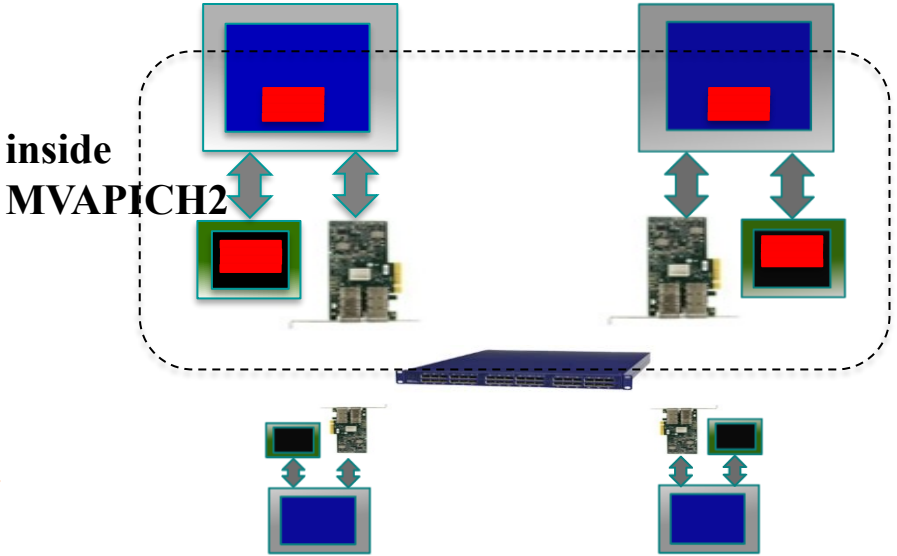
At Sender:

```
MPI_Send(s_devbuf, size, …);
```

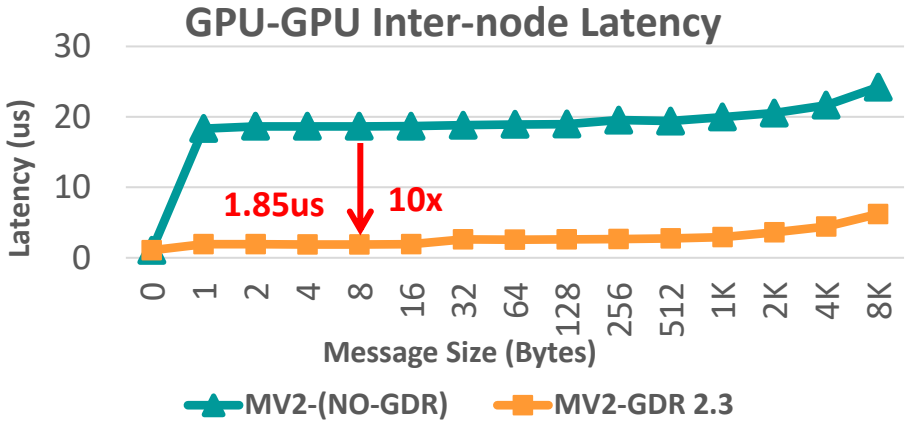
At Receiver:

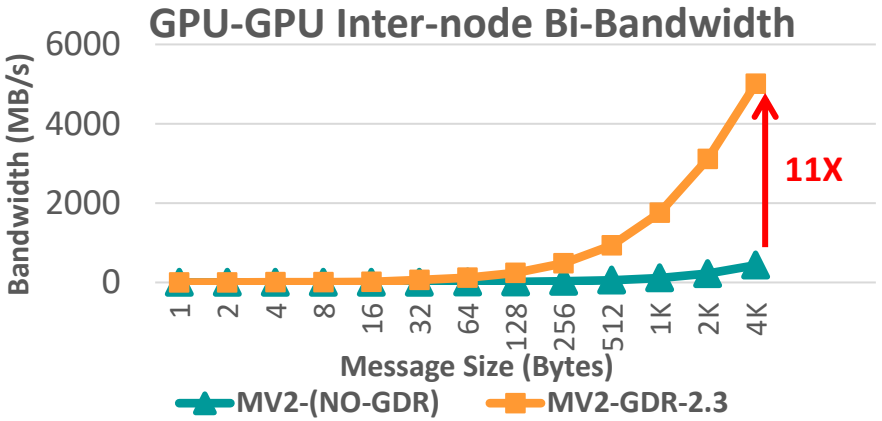
```
MPI_Recv(r_devbuf, size, …);
```

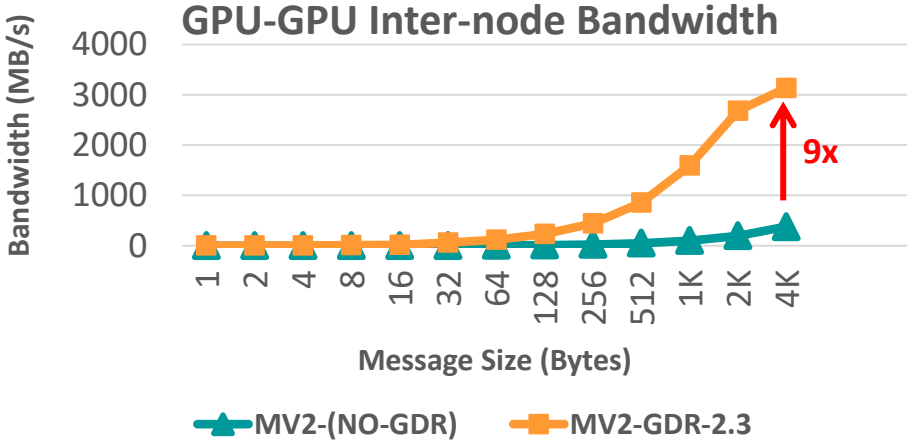
High Performance and High Productivity



Optimized MVAPICH2-GDR Design



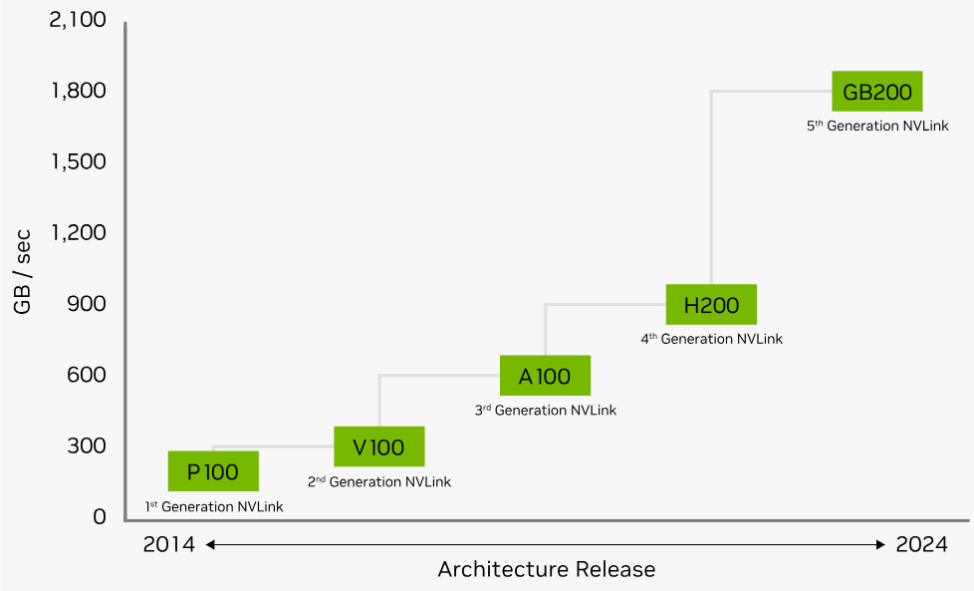




MVAPICH2-GDR-2.3.1
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

NVLink and NVLink2

- High-performance interconnect for emerging dense GPU systems
  - Allows Load-Store operations between all GPUs

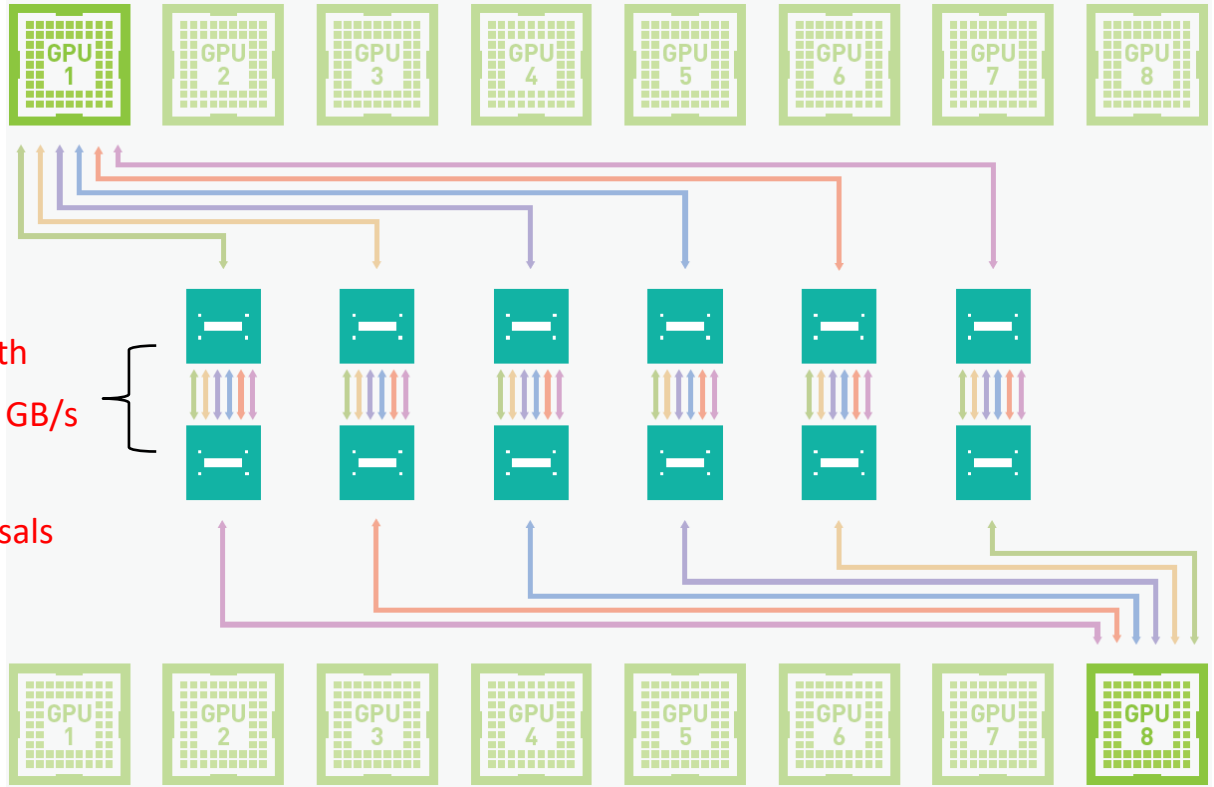


	Second Generation	Third Generation	Fourth Generation	Fifth Generation
NVLink bandwidth per GPU	300GB/s	600GB/s	900GB/s	1,800GB/s
Maximum Number of Links per GPU	6	12	18	18
Supported NVIDIA Architectures	NVIDIA Volta™ architecture	NVIDIA Ampere architecture	NVIDIA Hopper™ architecture	NVIDIA Blackwell architecture

NVLink Performance Trends

Courtesy: NVIDIA

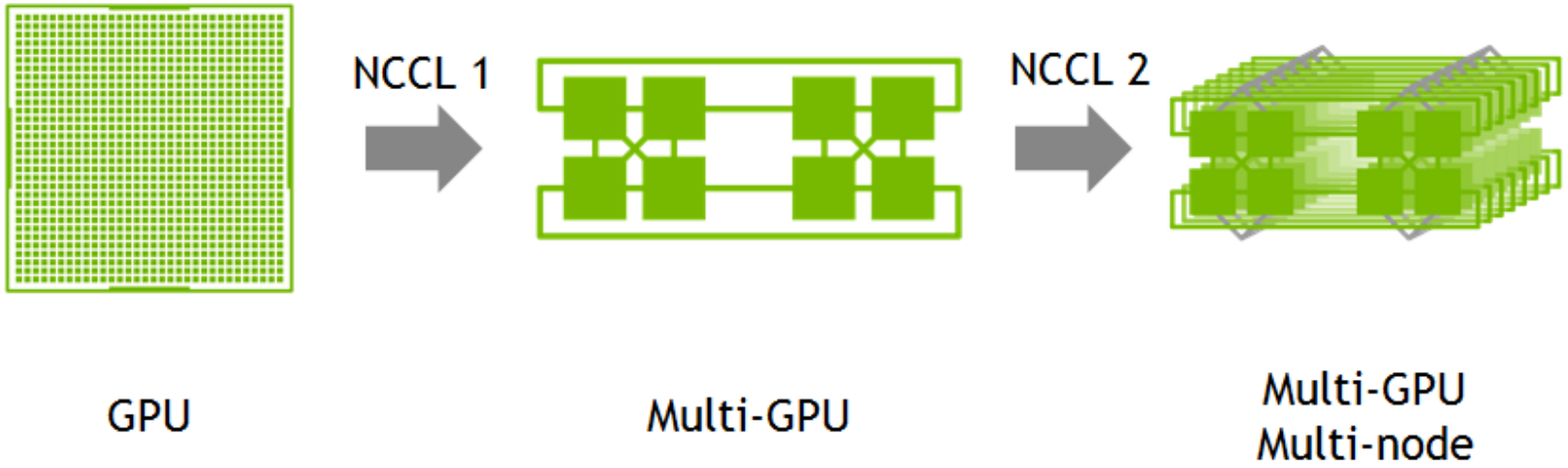
NVSwitch Topology



Courtesy: http://images.nvidia.com/content/pdf/nvswitch-technical-overview.pdf

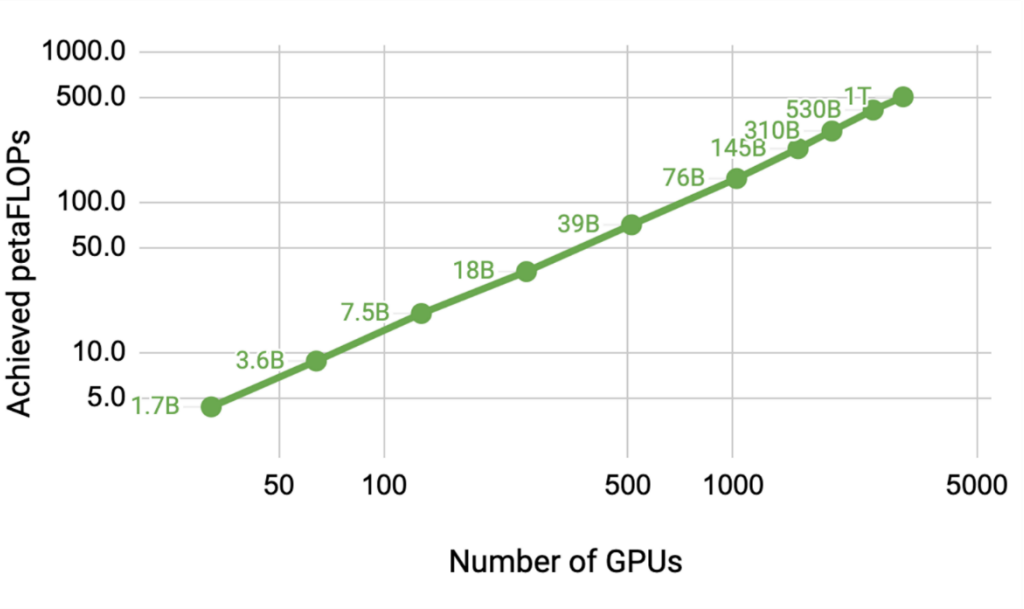
NCCL Communication Library

- NVIDIA Collective Communication Library (NCCL)
- Main Motivation: Deep Learning workloads
- NCCL1– efficient dense-GPU communication within the node
- NCCL2– multiple DGX systems connected to each other with InfiniBand systems



Courtesy: https://developer.nvidia.com/nccl

Scaling Large Language Models



Weak scaling performance for GPT models ranging from 1 billion to 1 trillion parameters

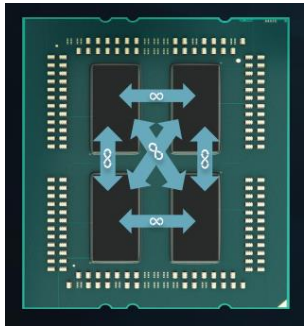
Setup:

- 3D Parallelism with Megatron-LM
- DGX system with 8 NVIDIA 80-GB A100 per node connected via NVLink.
- 3072 A100 GPUs (384 DGX nodes)
- 200Gbps HDR InfiniBand interconnect between nodes.

Courtesy: https://developer.nvidia.com/blog/scaling-language-model-training-to-a-trillion-parameters-using-megatron/

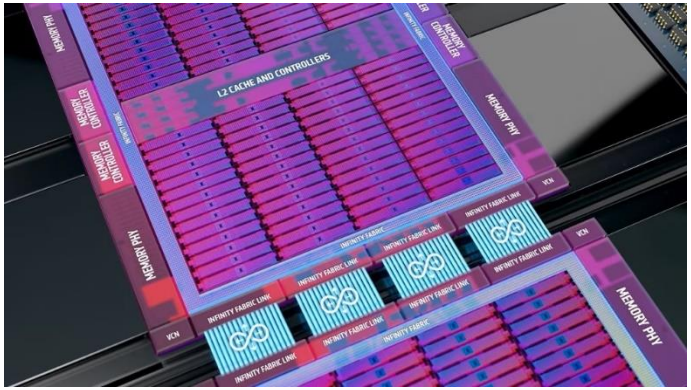
AMD Infinity Fabric

- A cache coherent interconnect data fabric
- Used within/across chiplets on CPUs as well as to provide a high-speed fabric between GPUs
- Up to 145GB/s DRAM bandwidth/socket
- Up to 400GB/s bandwidth between Graphical computing dies (GCDs) on MI250/MI250X
- Maintains cache coherence between CPUs and GPUs when using 3rd generation AMD EPYC processors



Die to Die interconnect on CPUs

Courtesy: AMD

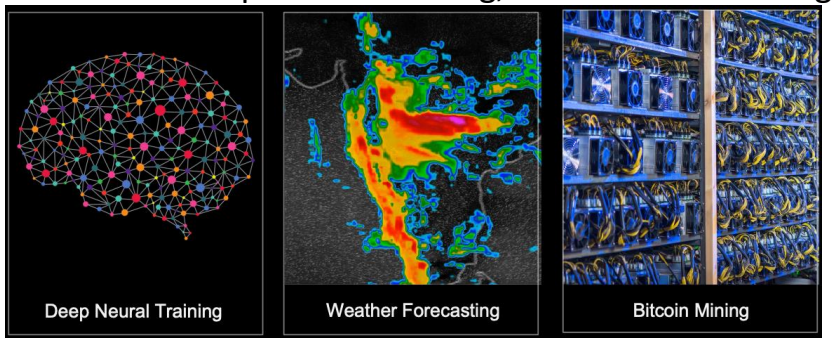


Infinity fabric

Courtesy: https://www.amd.com/en/technologies/cdna2

RCCL Communication Library

- ROCm Collective Communication Library (RCCL) – (pronounced “Rickel”)
- Uses the same C API as NCCL
- Intra-node communication support
  - PCIe and xGMI high-speed interconnects
  - InfiniBand, RoCE, and TCP/IP for inter-node communication.
- Inter-node communication support
  - InfiniBand, RoCE, and TCP/IP
- Useful for multi-GPU computing of workloads
  - Deep Neural training, Weather Forecasting, Bitcoin Mining



Multi-GPU computing use cases

Courtesy :

https://www.amd.com/system/files/documents/multi-gpu-6.pdf

https://rocm.docs.amd.com/projects/rccl/en/develop/

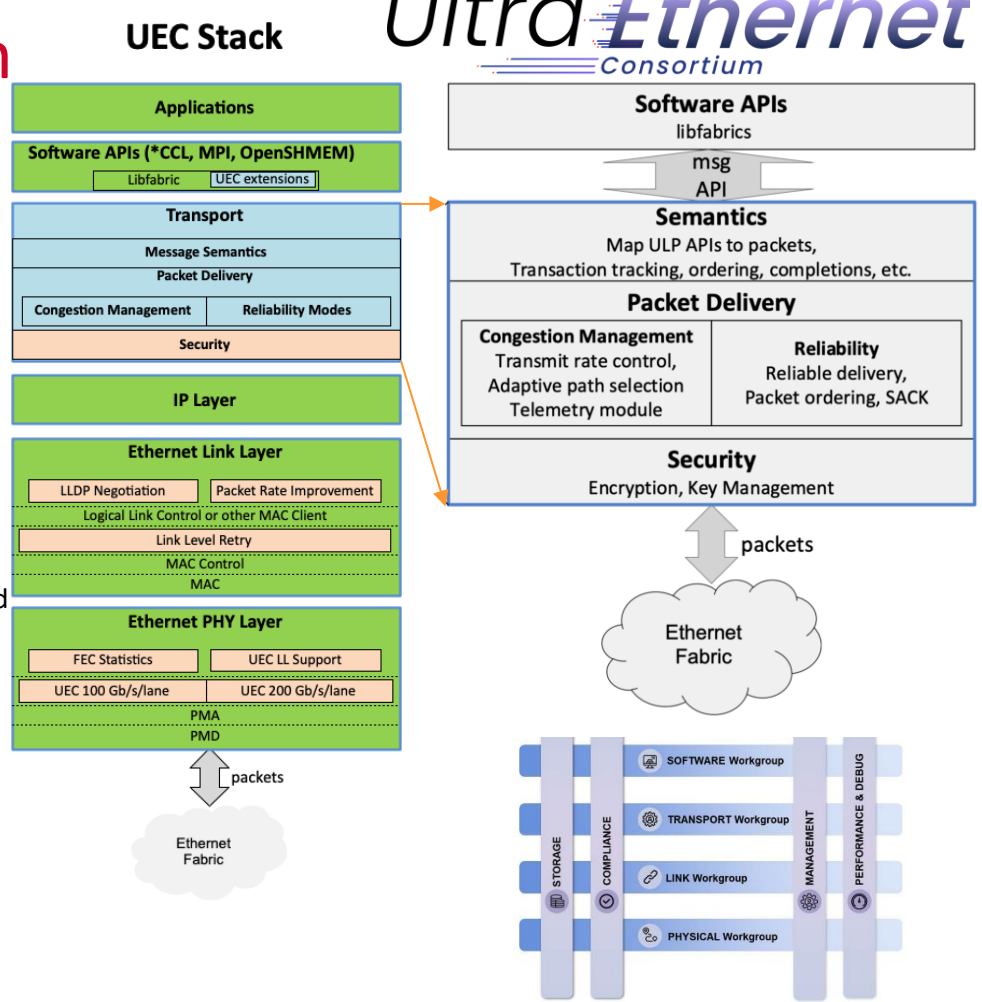
Ultra-Accelerator Link Consortium (UALink)

Courtesy: https://ualinkconsortium.org

- Open Standard for AI Accelerator-to-Accelerator Communication
  - Abstract the notion of Accelerator to allow for easy plug-and-play alongside interconnects such as CXL, PCIe, XGMI, AMD InfinityFabric, etc.
- Focus on direct load, store, atomic ops between accelerators, connected via an "UltraLink Switch"
  - Low-latency/high-bandwidth fabric
  - 100s of accelerators supported within a pod
- Version 1.0 of the standard is available now! https://ualinkconsortium.org/

Ultra Ethernet Consortium

- Aimed at addressing the challenges posed by modern AI and HPC Jobs
- Aiming to build on top of advantage ethernet has in terms of adoption
- Goal -> “Tail latency” should be minimized
  - Multi-pathing and packet spraying
  - Flexible delivery order
  - Modern congestion control mechanisms
  - End-to-end telemetry
  - Larger scale, stability, and reliability
- Aims to address issues with current transport protocol services used by RoCE and IB [1]
  - Issues with DCQCN congestion control mechanism
  - Recovering from lost or out of order packet
  - Use a more scalable transport protocols with compared to RC which has N^2 connection overhead
  - Improved load balancing capabilities to handle larger messages/flows used by AI workloads



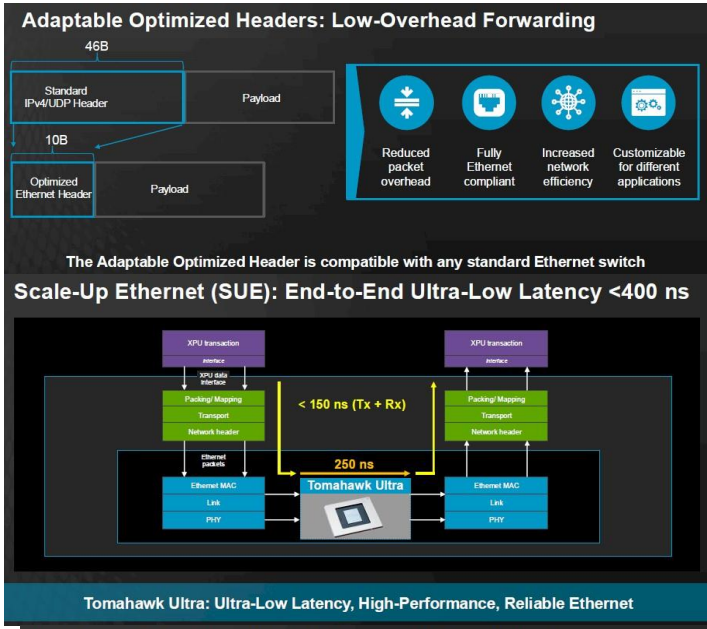
[1] Data Center Ethernet and Remote Direct Memory Access: Issues at Hyperscale, Hoefler et al., in Computer, July 2023

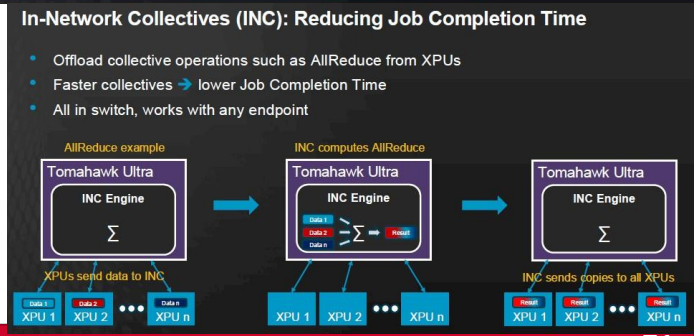
Courtesy: Ultra Ethernet Consortium

UEC-Complaint Hardware: Broadcom Tomahawk Ultra

- Early adoption of UEC Standard Effort
  - Reduced-size Ethernet header for improved routing latency
- Scale-Up Ethernet (SUE): Alternative to NVLink AND UALink
  - x < 400ns for XPU-XPU transfer time, < 150ns for Tx/Rx at Transport Layer
- In-Network Collectives/Computing (INC)
  - Analogous to NVIDIA SHARP
  - Similar set of support (Blocking/Nonblocking Barrier, Bcast, All/reduce)

- Courtesy: https://www.nextplatform.com/2025/07/17/broadcom-tries-to-kill-infiniband-and-nvswitch-with-one-ethernet-stone/

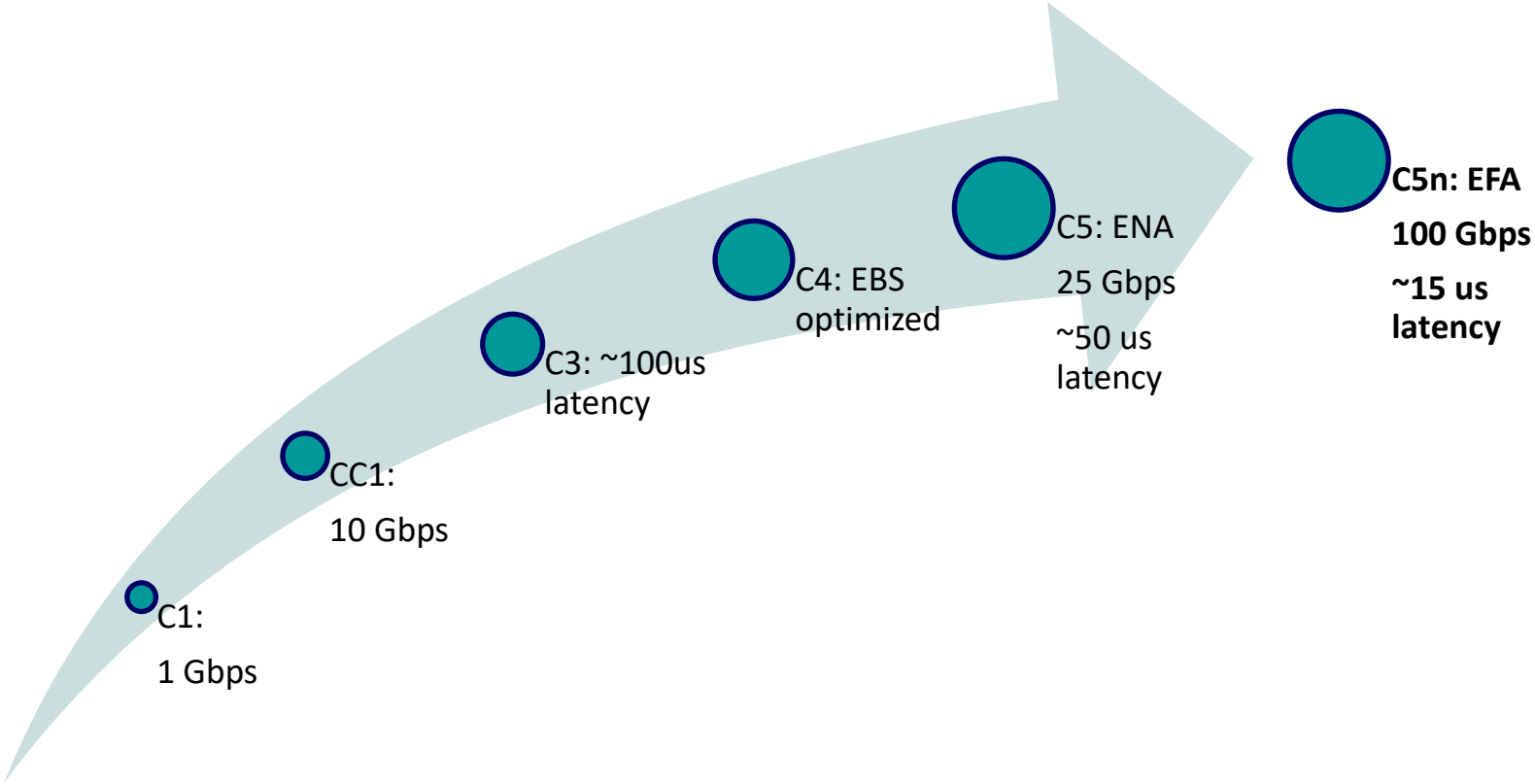




Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- **Architectural Overview of High-Performance Networks**
  - IB, HSE, their Convergence and Features
  - GPU-aware support in modern HPC networks:
    - NVLink and NVSwitch Interconnect Architecture
    - AMD Infinity Fabric Interconnect Architecture
  - **Amazon EFA Interconnect Architecture**
  - Cray Slingshot Interconnect Architecture
- Overview of Emerging Smart Network Interfaces
  - NVIDIA BlueField DPUs, AMD Pensando Smart NICs, and Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

Evolution of networking on AWS



Deep Dive on OpenMPI and Elastic Fabric Adapter (EFA) - AWS Online Tech Talks, Linda Hedges

Amazon Elastic Fabric Adapter (EFA)

- Enhanced version of Elastic Network Adapter (ENA)
- Network aware multi-path routing
- Exposed through libibverbs and libfabric interfaces
- Introduces new Queue-Pair (QP) type
  - Scalable Reliable Datagram (SRD)
  - Also supports Unreliable Datagram (UD)
  - No support for Reliable Connected (RC)
- Low latency, OS bypass
- Libfabric-based, works with Intel MPI, Open MPI, MPICH, MVAPICH2, Nvidia NCCL, etc.
- Scalable Reliable Datagram (SRD)
  - Unordered, reliable, connectionless
  - Highly multipathed
  - Latency-based congestion control

Generation	Latency	Bandwidth	New Features
1st (2018)	14 µs	100 Gbps	Send/recv semantics
2nd (2020)	9.5 µs	170 Gbps	RDMA semantics
3rd (2022)	6.5 µs	200 Gbps	ML hardware optimizations

IB Transport Types and Associated Trade-offs

Attribute		Reliable Connection	Reliable Datagram	Dynamic Connected	Scalable Reliable Datagram	Unreliable Connection	Unreliable Datagram	Raw Datagram
Scalability (M processes, N nodes)		M^2N QPs per HCA	M QPs per HCA	M QPs per HCA	M QPs per HCA	M^2N QPs per HCA	M QPs per HCA	1 QP per HCA
Reliability	**Corrupt data detected**	Yes						
	Data Delivery Guarantee	Data delivered exactly once				No guarantees		
	Data Order Guarantees	Per connection	One source to multiple destinations	Per connection	No	Unordered, duplicate data detected	No	No
	Data Loss Detected	Yes					No	No
	Error Recovery	Errors (retransmissions, alternate path, etc.) handled by transport layer. Client only involved in handling fatal errors (links broken, protection violation, etc.)				Errors are reported to responder	None	None

Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- **Architectural Overview of High-Performance Networks**
  - IB, HSE, their Convergence and Features
  - GPU-aware support in modern HPC networks:
    - NVLink and NVSwitch Interconnect Architecture
    - AMD Infinity Fabric Interconnect Architecture
  - Amazon EFA Interconnect Architecture
  - **Cray Slingshot Interconnect Architecture**
- Overview of Emerging Smart Network Interfaces
  - NVIDIA BlueField DPUs, AMD Pensando Smart NICs, and Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

Slingshot: HPE/Cray's 8th Generation Scalable Interconnect

64 ports x 200 Gbps

Over 250K endpoints with a diameter of just three hops

Ethernet Compatible

Easy connectivity to datacenters and third-party storage.

"HPC inside"

World class Adaptive Routing and QoS

High utilization at scale. Strong support for hybrid workloads.

Efficient Congestion Control

Performance isolation between workloads.

Low, Uniform Latency

Focus on tail latency, because real apps synchronize.

Courtesy: HPE/Cray Inc. (ExaComm '19 Keynote Talk by Steve Scott)

Slingshot Quality of Service Classes

- Highly tunable QoS classes
  - Priority, ordering routing protocol, minimum bandwidth guarantees, maximum bandwidth constraints, etc.
- Supports multiple, overlaid virtual networks…
  - High priority compute
  - Standard compute
  - Low-latency control & synchronization
  - Bulk I/O
  - Scavenger class background
- Jobs can use multiple traffic classes
- Provides performance isolation for different types of traffic
  - Small message reductions do not get stuck behind large messages
  - Less interference between compute and I/O

Courtesy: HPE/Cray Inc. (ExaComm '19 Keynote Talk by Steve Scott)

Slingshot Congestion Management

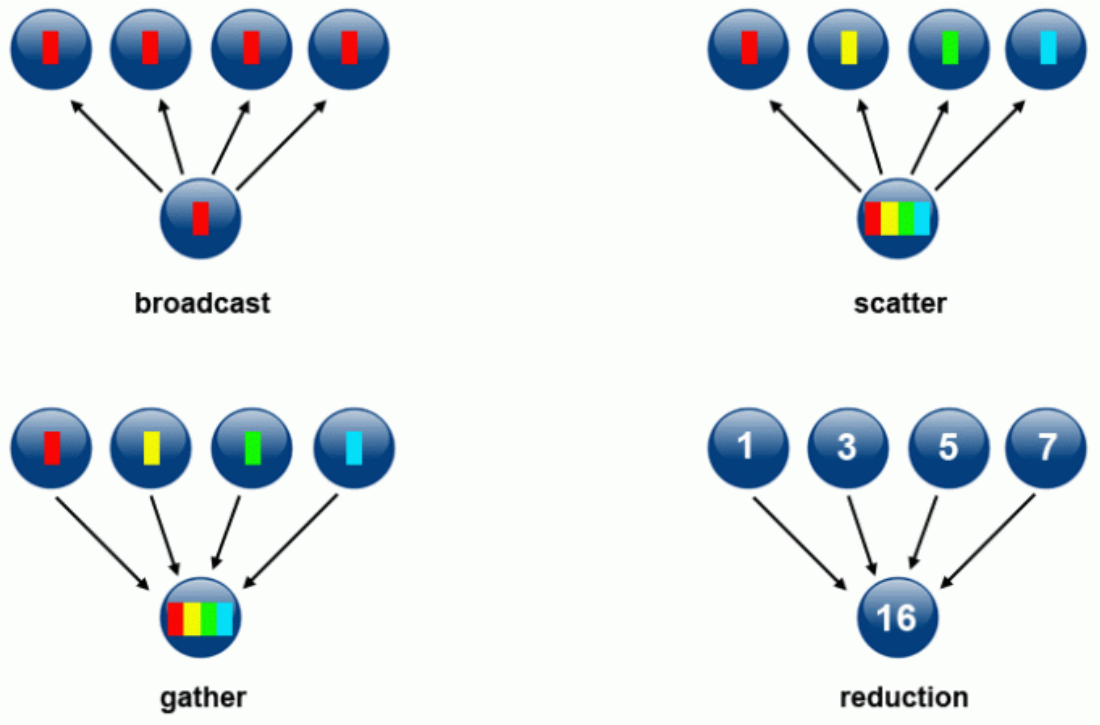
- Hardware automatically tracks all outstanding packets
  - Knows what is flowing between every pair of endpoints
- Quickly identifies and controls causes of congestion
  - Pushes back on sources… just enough
  - Frees up buffer space for everyone else
  - Other traffic not affected
  - Avoids HOL blocking end to end
- Fast and stable across wide variety of traffic patterns
  - Suitable for dynamic HPC traffic
- Performance isolation between apps on same QoS class
  - Applications much less vulnerable to other traffic on the network
  - Predictable runtimes
  - Lower mean and tail latency – a big benefit in apps with global synchronization

Courtesy: HPE/Cray Inc. (ExaComm '19 Keynote Talk by Steve Scott)

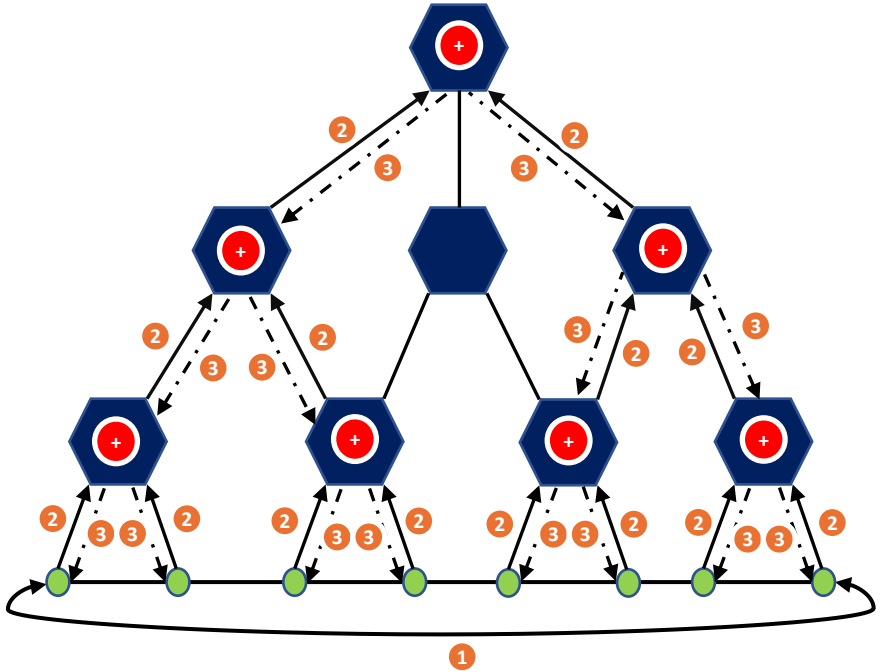
Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
- **Overview of Emerging Smart Networking Technologies**
  - **Collectives (NVIDIA SHARP)**
  - Overview of SmartNIC Architecture
  - NVIDIA BlueField DPUs
  - AMD Pensando Smart NICs
  - Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

Collective Communication (across CPUs or GPUs)



In-Network Computing - Collective Support (SHARP)



CPU/GPU

Reduce data in switch

Switch

Broadcast reduced data from switch

Send data to switch

1. Initialize in-network offload once per job
2. Send data to switch
3. Receive data from switch

No involvement of CPU/GPU once data reaches switches

Presentation Overview

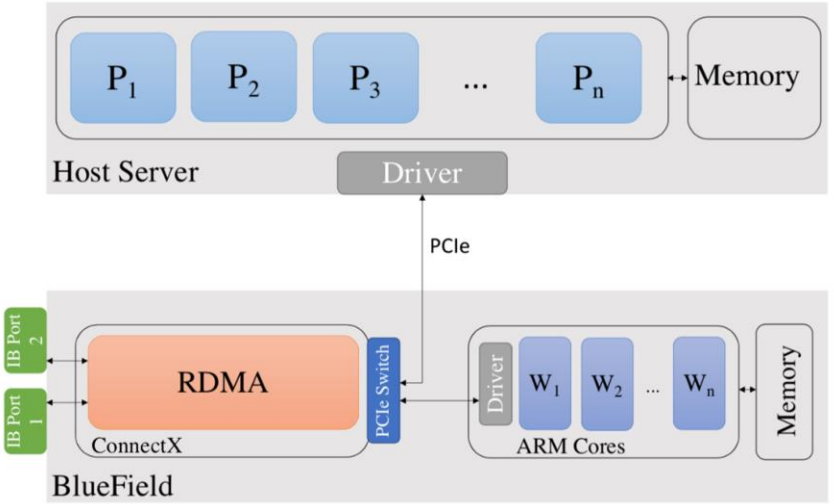
- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
- **Overview of Emerging Smart Networking Technologies**
  - Collectives (NVIDIA SHARP)
  - **Overview of SmartNIC Architecture**
  - **NVIDIA BlueField DPUs**
  - AMD Pensando Smart NICs
  - Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

SmartNICs

- Broadly two types of SmartNICs
  - CPU based : NIC + Programmable CPU cores + ASICs
    - NVIDIA Bluefield Data Processing Units (DPUs)
    - Marvell Octeon
    - AMD Pensando DSC
  - Field Programmable Gate Arrays (FPGAs) based : NIC + FPGAs
    - AMD Alveo
    - Intels FPGA SmartNICs

Accelerating Applications with BlueField-3 DPU

- InfiniBand network adapter with up to 400Gbps speed
- System-on-chip containing 16 64-bit ARMv8.2 A78 cores with 2.75 GHz each
- Up to 32 GB of memory for the ARM cores



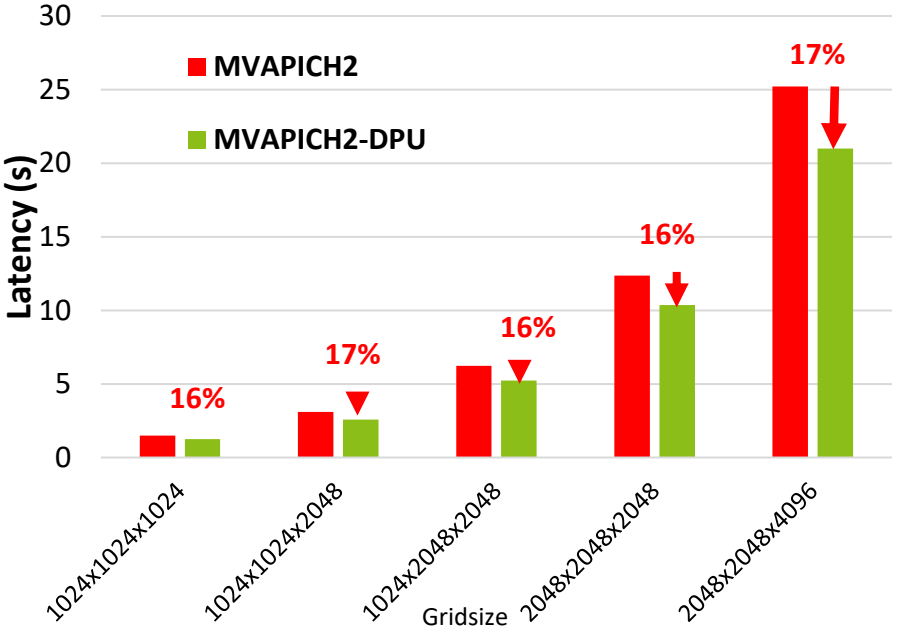
MVAPICH2-DPU Library Release

X-ScaleSolutions

- Supports all features available with the MVAPICH2 release (http://mvapich.cse.ohio-state.edu)
- Novel framework to offload non-blocking collectives to DPU
- Offloads non-blocking Alltoall/v (MPI_Ialltoall/v) to DPU
- Offloads non/blocking point-to-point to the DPU
- Offloads non-blocking Broadcast (MPI_Ibcast) to DPU

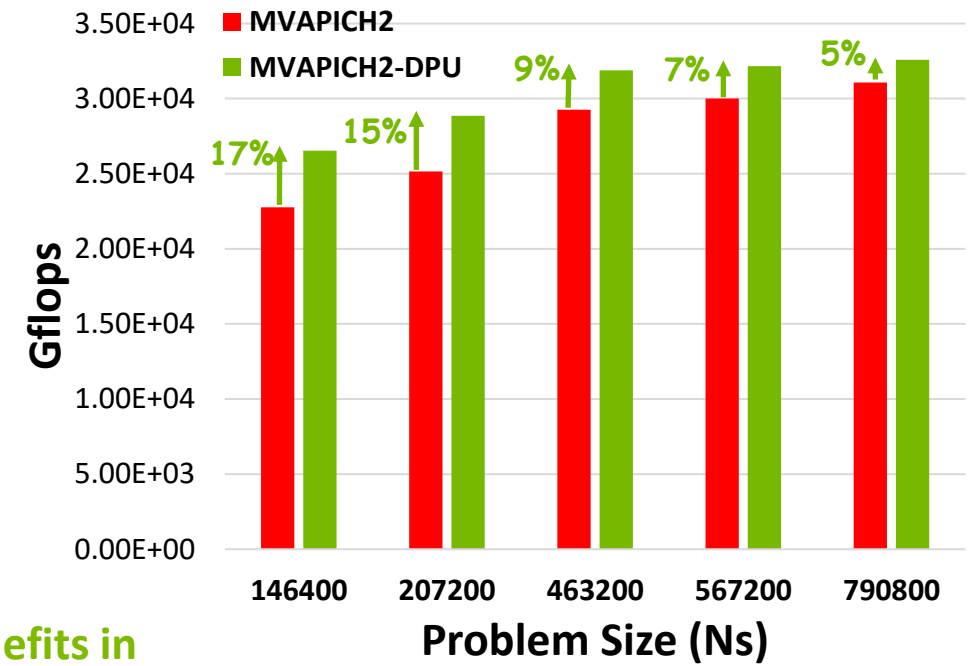
Available from X-ScaleSolutions, please send a note to contactus@x-scalesolutions.com to get a trial license.

Application-level Benefits (P3DFFT, HPL With DPU Co-Design)



P3DFFT,16 Nodes, 32 PPN

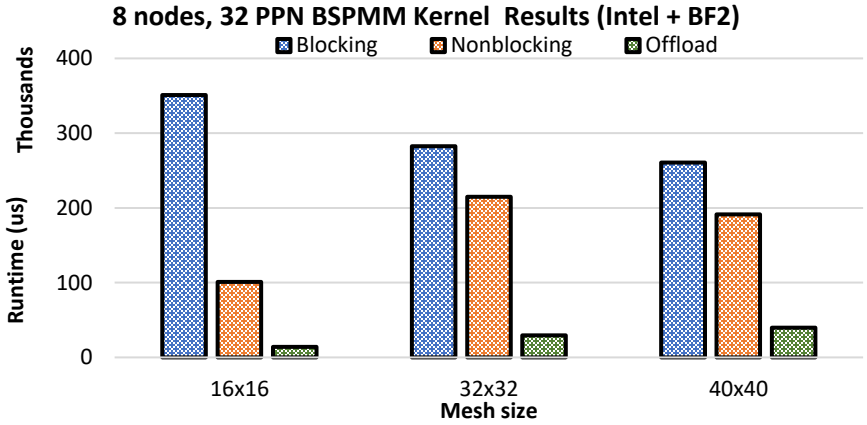
Benefits in application-level execution time

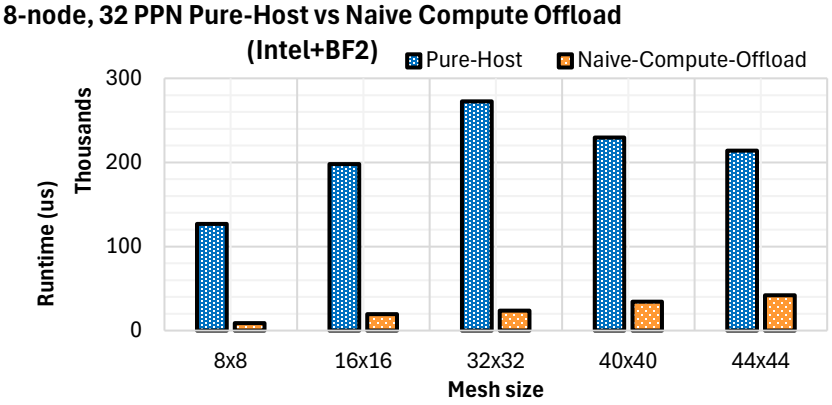


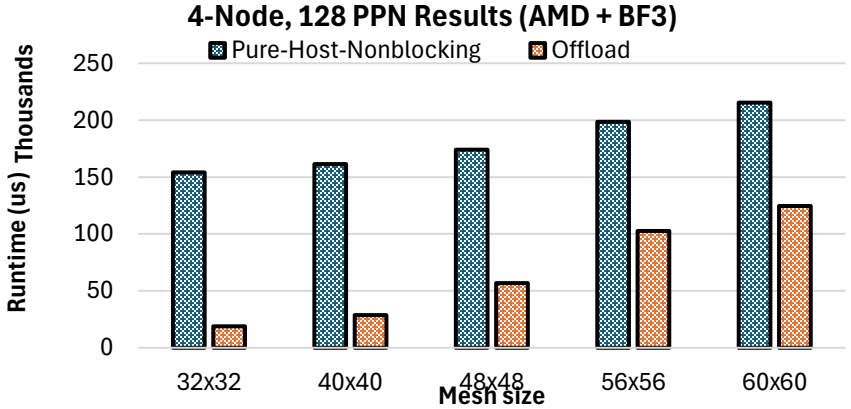
HPL via XScaleHPL-DPU, 31x32 process grid

Use of BlueField-2/3 DPUs to Offload One-Sided Communication

- Use of GVMI and IB Primitives to create APIs for offloading one-sided MPI Put/Get and OpenSHMEM Nonblocking put/get (RMA)
- Use of Block Sparse Matrix Multiplication (BSPMM) kernel with get/compute/update pattern (Comparison against blocking RMA and nonblocking RMA versions of the kernel)



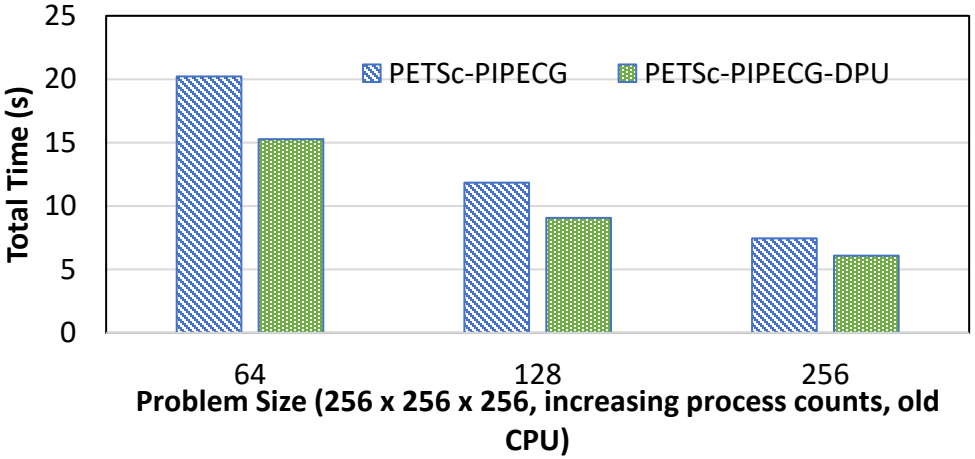


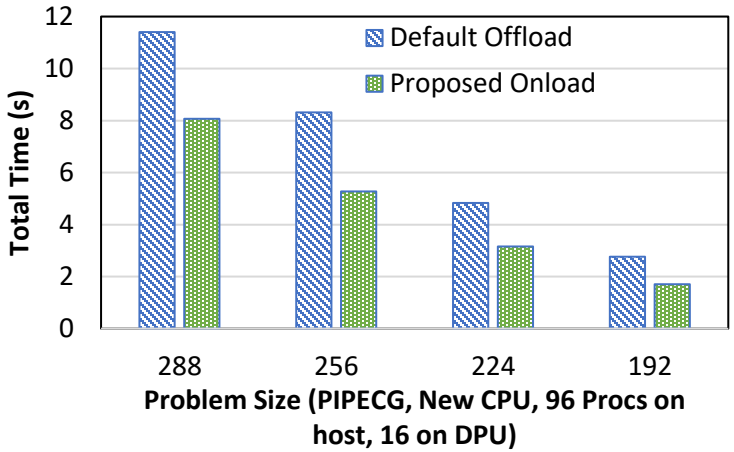


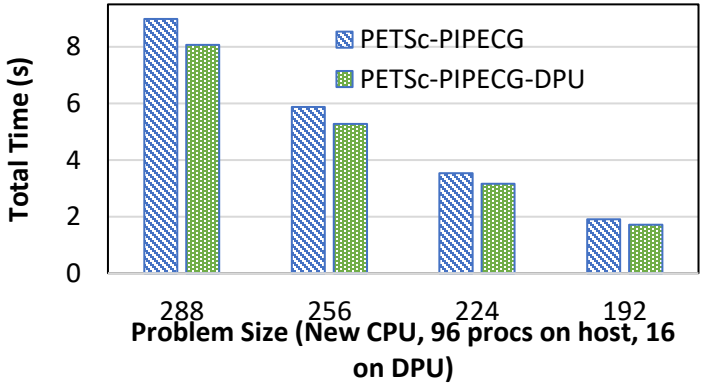
B. Michalowicz, K. Suresh, H. Subramoni, M. Abduljabbar, DK Panda, and S. Poole, Efficient Offloading Designs for One-Sided Communication to SmartNICs, HiPC '24, Dec 2024.

Smart Compute Offload (Hybrid CPU + BlueField-3 SmartNICs)

- Targeted towards libraries like PETSc and HYPRE
  - Creating a set of APIs for Vector-Multiply Add (VMA), Distributed Dot (DDOT), and Matrix-Vector (MATVEC) operations
  - Onloading Scheme for reducing cost of data movement
- Older CPUs (Intel Broadwell) + BF3: Up to 24% performance improvement
- Newer CPUs (Intel SPR) + BF3: Up to 10-15% Improvement







K. Suresh, B. Michalowicz, N. Contini, B. Ramesh, M. Abduljabbar, A. Shafi, H. Subramoni, and DK Panda, Using Bluefield-3 SmartNICs to Offload Vector Operations in the Kryolov Subspace Method, HiPC '24, Dec 2024.

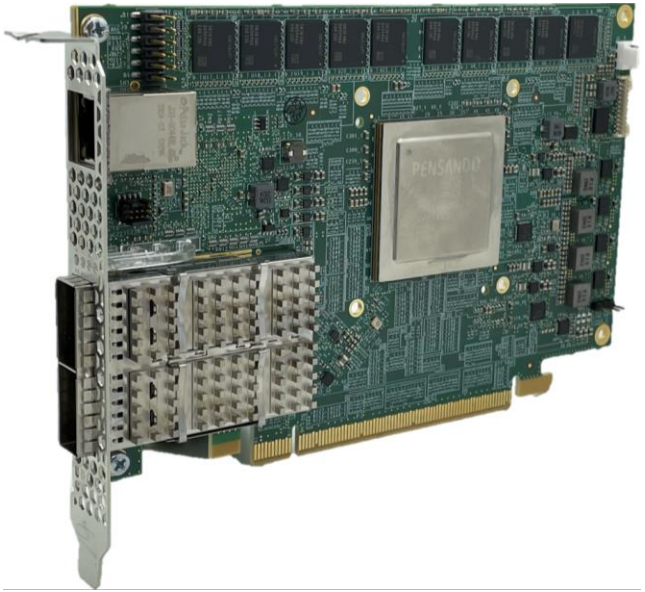
Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
- **Overview of Emerging Smart Networking Technologies**
  - Collectives (NVIDIA SHARP)
  - Overview of SmartNIC Architecture
  - NVIDIA BlueField DPUs
  - **AMD Pensando Smart NICs**
  - **Intel Columbiaville IPUs**
- High-Performance Network Deployments for AI Workloads
  - Cerebras and Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

AMD SmartNICs

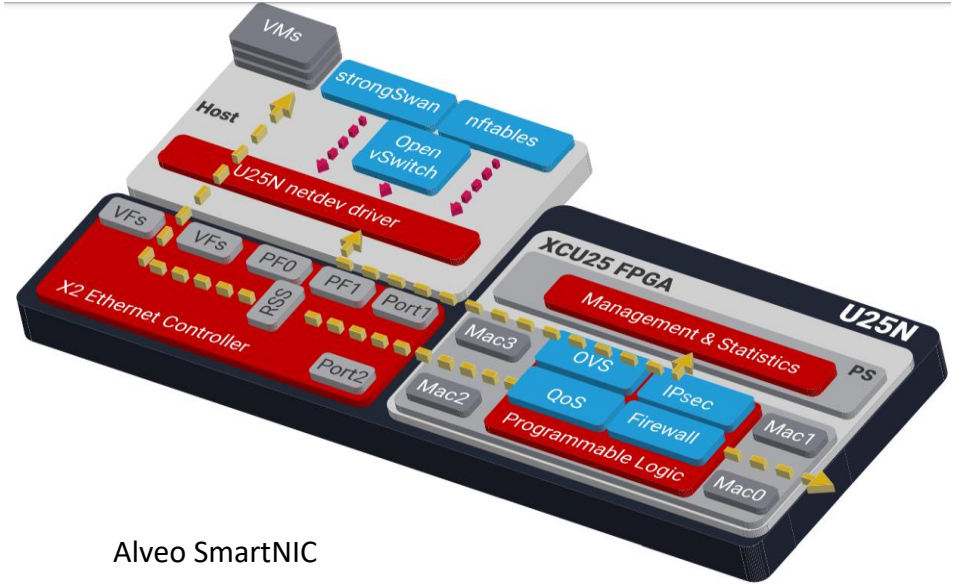
Two types of SmartNICs

- CPU based AMD Pensando DSC
- FPGA based Alveo



AMD Pensando DSC

- Powered by Pensando DPU
- P4- programmable custom match processing units (MPUs)
- combined with a 16x A72 ARM® core complex
- dedicated data encryption and storage offload engines

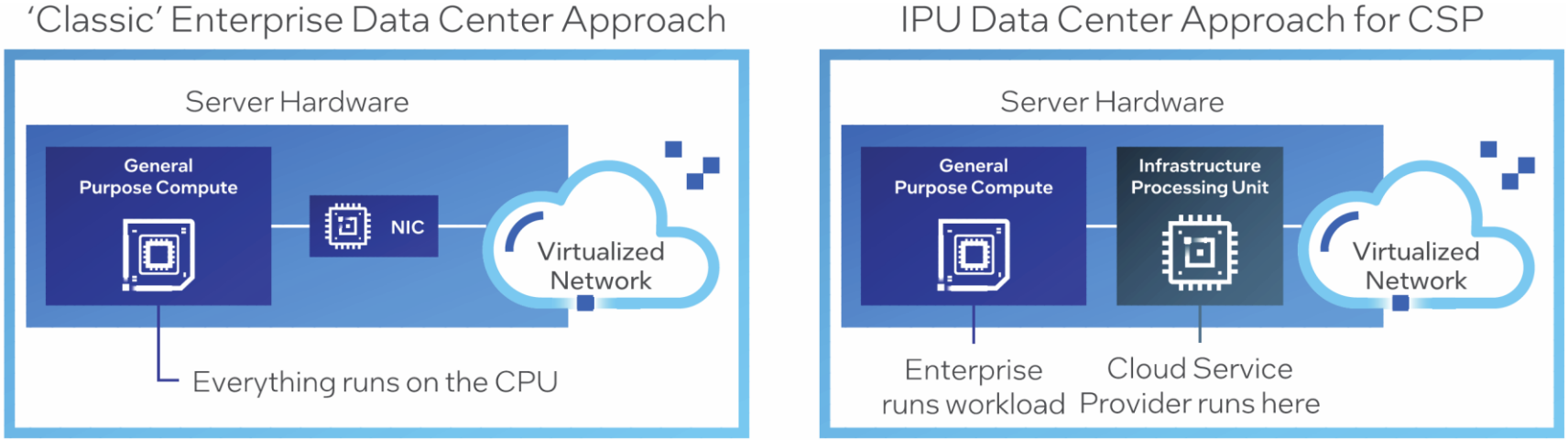


Alveo SmartNIC

- Eg: U25N SmartNIC
- XtremeScale™ X2 Ethernet Controller
- AMD UltraScale+™ FPGA
- Multi-core Arm processor
- FPGA has programmable dataplane piplenes like QoS, IPsec, Match Engine

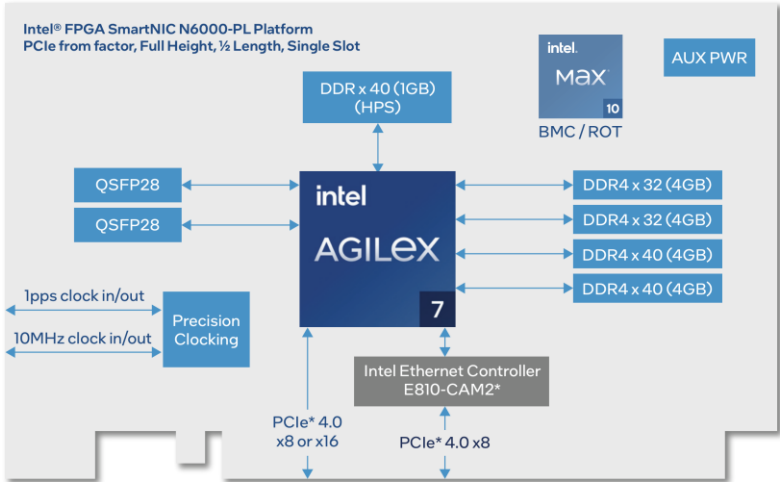
Intel Smart Networking Devices

- Two types of Smart Networking devices
  - Intel Infrastructure Processing Units (IPUs)
    - Primarily used to provide Cloud Services by Offloading Network, Storage, Security
  - Intel FPGA based SmartNICs
    - Programmable network device to accelerate infrastructure applications
    - Unlike IPUs, cannot offload entire infrastructure stack with storage and security



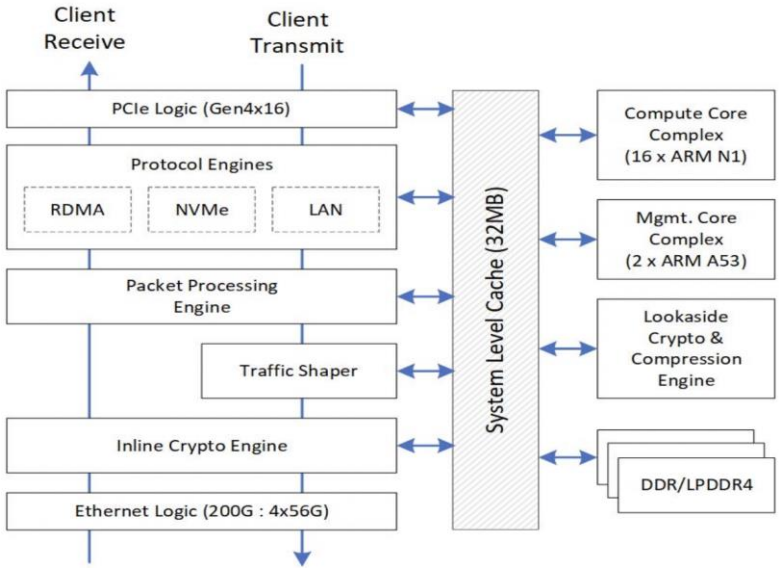
Intel IPUs and SmartNIC Examples

Intel FPGA SmartNIC NL6000



- 2x100 Gbps Ethernet
- Onboard Ethernet Controller
- Intel Agilex® 7 FPGA

Intel IPU E2000



- packet processing engine
- RDMA and storage capability including NVMe offload
- ARM Neoverse based compute complex

Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
- Overview of Emerging Smart Networking Technologies
  - Collectives (NVIDIA SHARP)
  - Overview of SmartNIC Architecture
  - NVIDIA BlueField DPUs
  - AMD Pensando Smart NICs
  - Intel Columbiaville IPUs
- **High-Performance Network Deployments for AI Workloads**
  - **Cerebras**
  - Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

Cerebras WSE-3 architecture

- Cores:
  - 4 trillion transistors (5nm TSMC process)
  - 900,000 AI cores
  - 125 petaflops of peak AI performance
- Memory:
  - 44GB on-chip SRAM; 21 PB/s
  - External memory: 1.5TB, 12TB, or 1.2PB
- Fabric Interconnection:
  - All cores connected in a 2D-mesh ("Swarm" – on chip interconnect)
  - 214 Pb/s

Courtesy: Cerebras Inc

Cerebras WSE-3
4 Trillion Transistors
46,225 mm² Silicon

Largest GPU
80 BillionTransistors
814 mm² Silicon

	WSE-3	Nvidia H100	Difference
Chip Size	46,225 mm²	826 mm²	57 X
Cores	900,000	16,896 FP32 + 528 Tensor	52X
On-chip memory	44 GB	0.05 GB	880 X
Memory bandwidth	21 PB/sec	0.003 PB/sec	7,000 X
Fabric bandwidth	214 Pb/sec	0.0576 Pb/sec	3,715 X

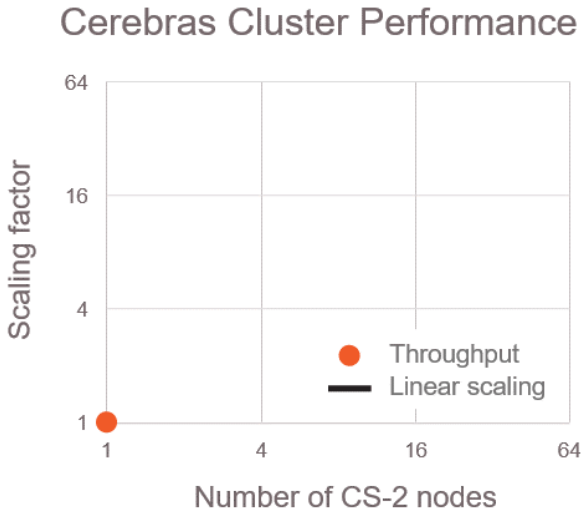
LLM Training Scalability with Cerebras

Ease-of-use

```
python run.py
--params params.yaml      ← Where's your dataset?
--num_csx = 1             ← How many nodes?
--model_dir = model_dir   ← Where to store weights?
--num_steps = 1000        ← How many training steps?
--mode=train              ← Train, evaluate or infer?
```

cerebras

Linear scaling



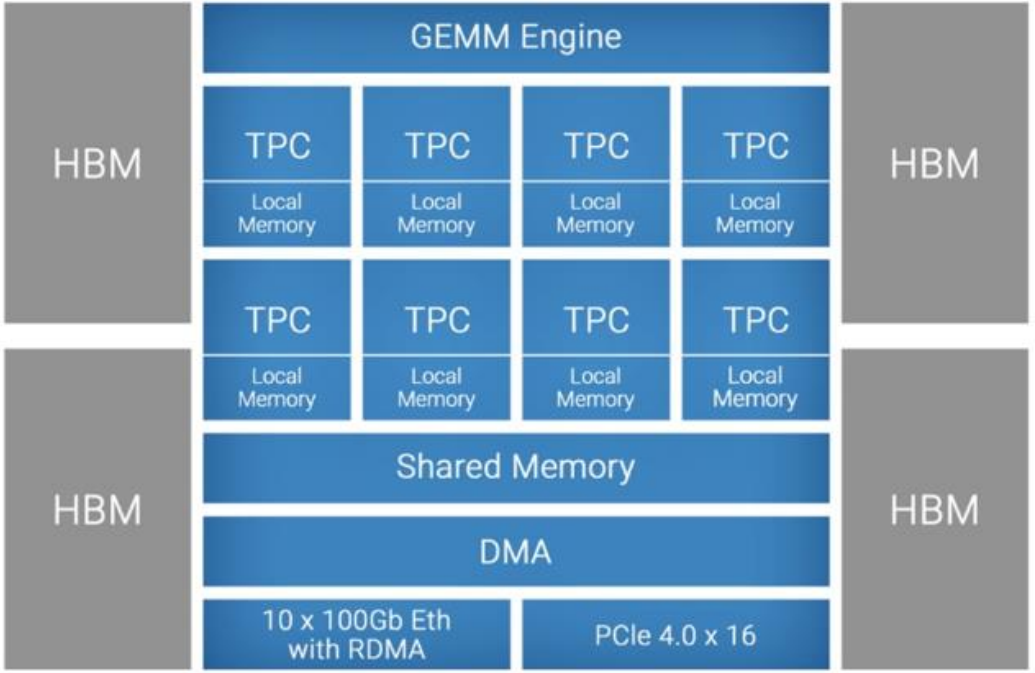
On GPT-3 XL model, Cerebras shows perfect linear scaling up to 16 CS-2s — that's perfect scaling up to 13.6 million cores.

Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
- Overview of Emerging Smart Networking Technologies
  - Collectives (NVIDIA SHARP)
  - Overview of SmartNIC Architecture
  - NVIDIA BlueField DPUs
  - AMD Pensando Smart NICs
  - Intel Columbiaville IPUs
- **High-Performance Network Deployments for AI Workloads**
  - Cerebras
  - **Habana-Gaudi**
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

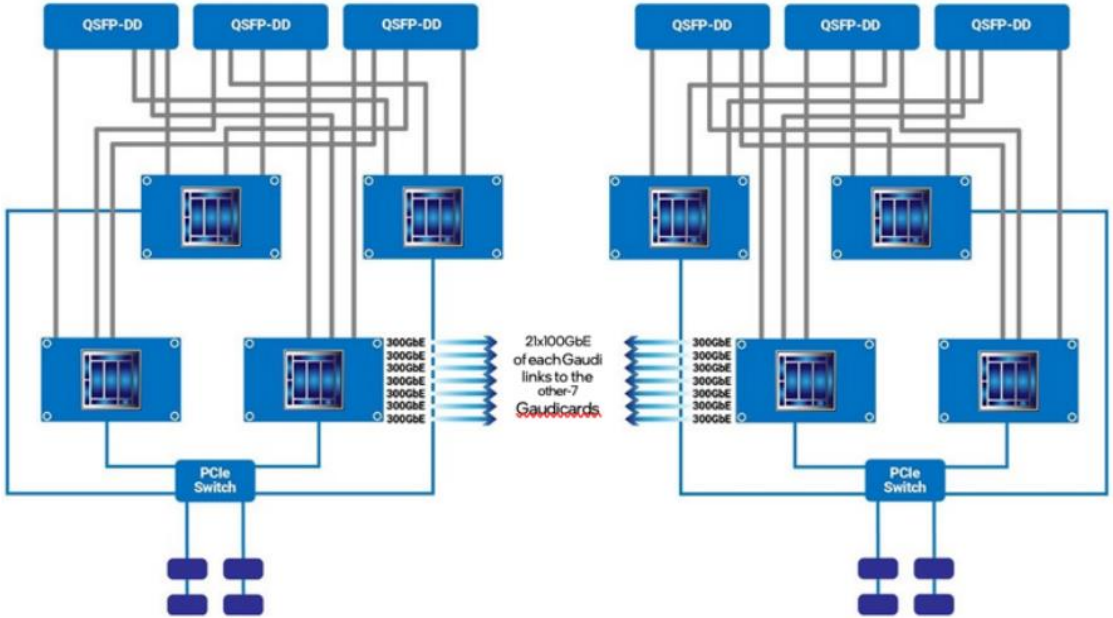
Habana Gaudi : Architecture

- AI processor is designed to maximize training throughput and efficiency
- Tensor Processing Cores (TPC) :
  - VLIW SIMD processor
  - GEMM operation acceleration
  - Supports: FP32, BF16, INT32, INT16, INT8, UINT32, UINT16, and UINT8
- Memory :
  - Per core on-die SRAM, local memories
  - four HBM devices, 32 GB Capacity, 1 TB/s bandwidth
- Network: RDMA over Converged Ethernet (**RoCE v2**) engines on-chip

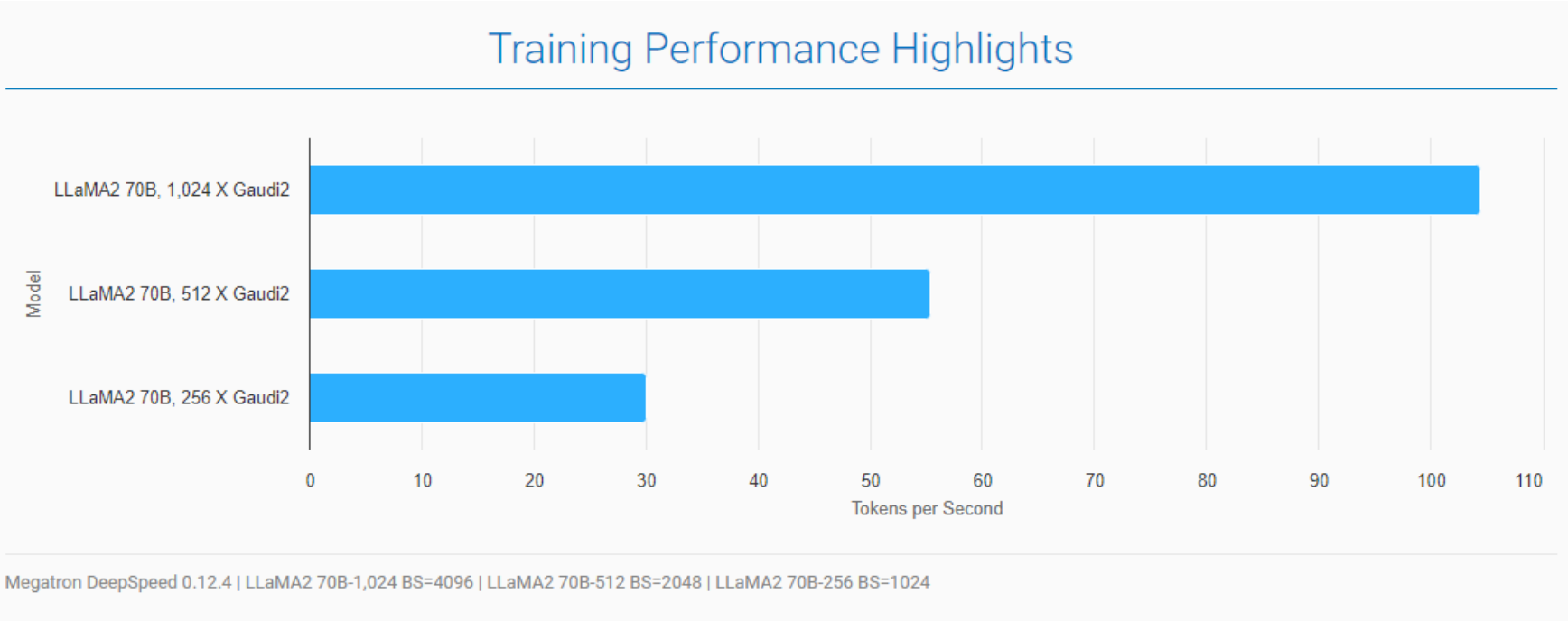


Scaling Out on Habana Gaudi

- HLS-Gaudi®2 Server:
  - All-to-all connectivity across eight Intel Gaudi2 processors
- Each server has 24x100GbE, three ports per Intel Gaudi2 accelerator
- The Habana Communication Library provides communication support



LLM Training Scalability with Habana Gaudi



Tokens per second training on LLaMA2 70B model with Gaudi2 HL-225H Mezzanine cards and two Intel® Xeon® Platinum 8380 CPU @ 2.30GHz, and 1TB of System Memory

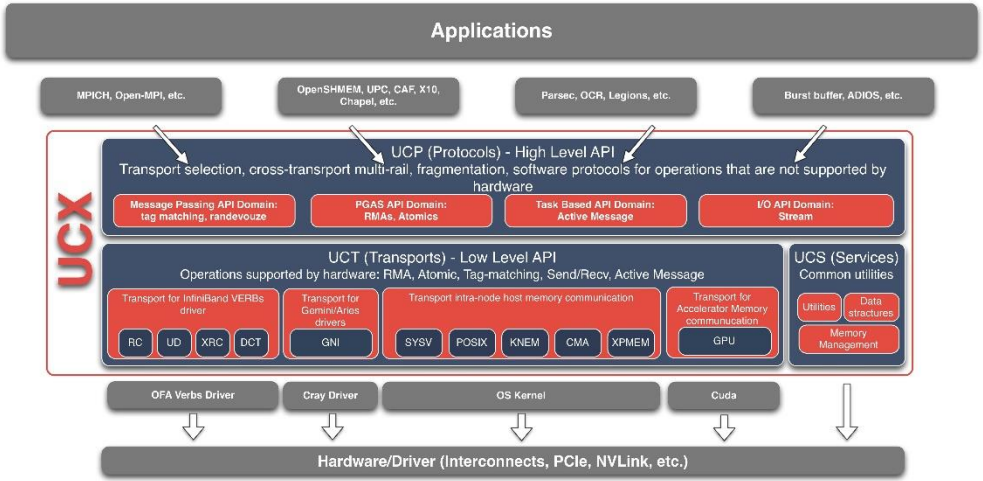
Presentation Overview

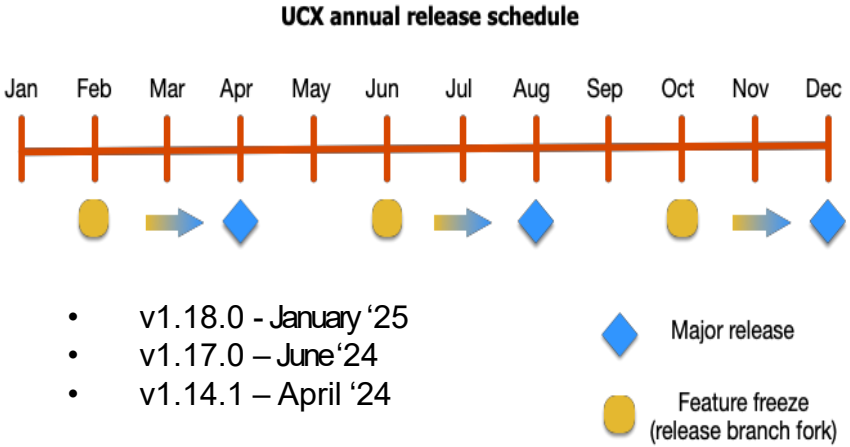
- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
- Overview of Emerging Smart Networking Technologies
  - Collectives (NVIDIA SHARP)
  - Overview of SmartNIC Architecture
  - NVIDIA BlueField DPUs
  - AMD Pensando Smart NICs
  - Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras
  - Habana-Gaudi
- **Overview of Software Stacks for Commodity High-Performance Networks**
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

Software Convergence with OpenFabrics

- Open source organization (formerly OpenIB)
  - www.openfabrics.org
- Incorporates both IB, RoCE, and iWARP in a unified manner
  - Support for Linux and Windows
- Users can download the entire stack and run
  - Latest stable release is OFED 4.17-1
    - New naming convention to get aligned with Linux Kernel Development
    - OFED 5.3 was under development

UCX Software Stack



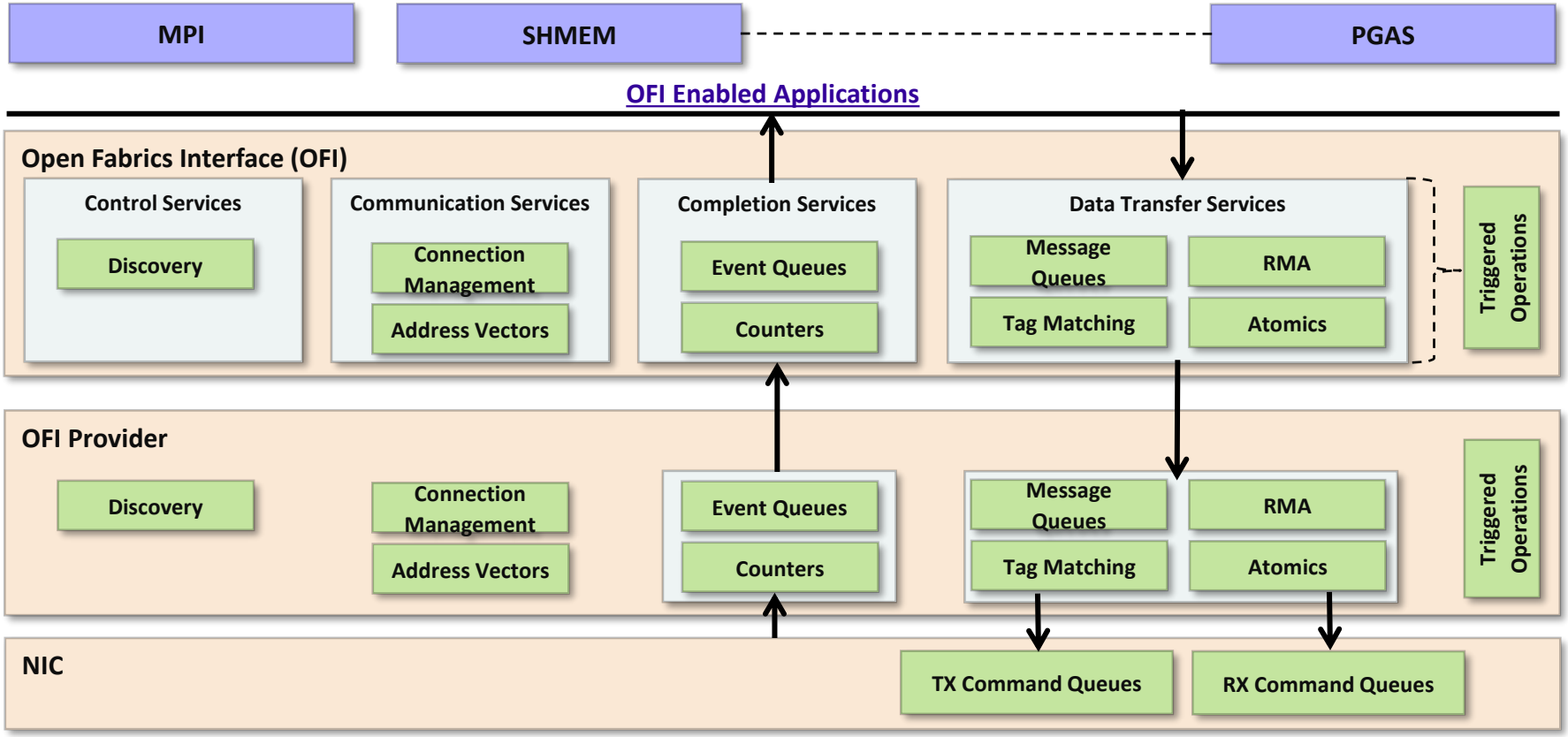


- Collaboration between industry, laboratories, and academia to create production grade communication frameworks and open standards for data centric, ML/AI, and high-performance applications
  - Tuned Support for x86_64 (Xeon/AMD), Power 8/9, Arm v8 (Cortex-A/N1/ThunderX2/Huawei)
  - Support for AMD and Nvidia GPUs
  - Runs on Servers, Raspberry PI like platforms, SmartNIC, Nvidia Jetson platforms, etc.

Courtesy: https://www.openucx.org/

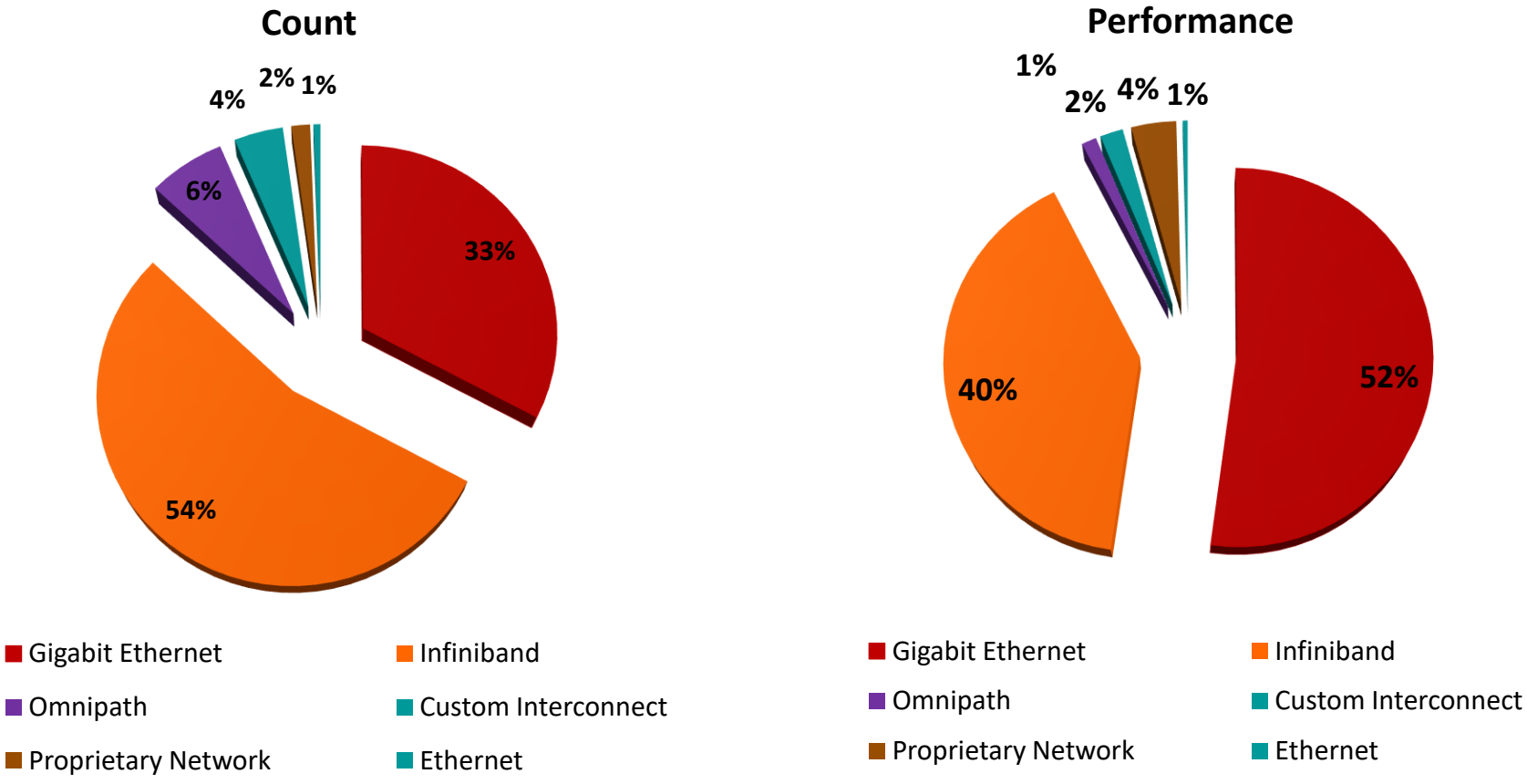
- Projects & Working Groups
  - UCX – Unified Communication X
  - UCC – Collective Library
  - OpenSNAPI – Smart network Project
  - SparkUCX – www.sparkucx.org
  - UCD – Advanced Datatype Engine
  - HPCA Benchmark – Benchmarking Effort

Libfabrics Software Stack



Courtesy: http://www.slideshare.net/seanhefty/ofi-overview?ref=http://ofiwg.github.io/libfabric/

InfiniBand in the Top500 (June 2025)



Large-scale InfiniBand Installations

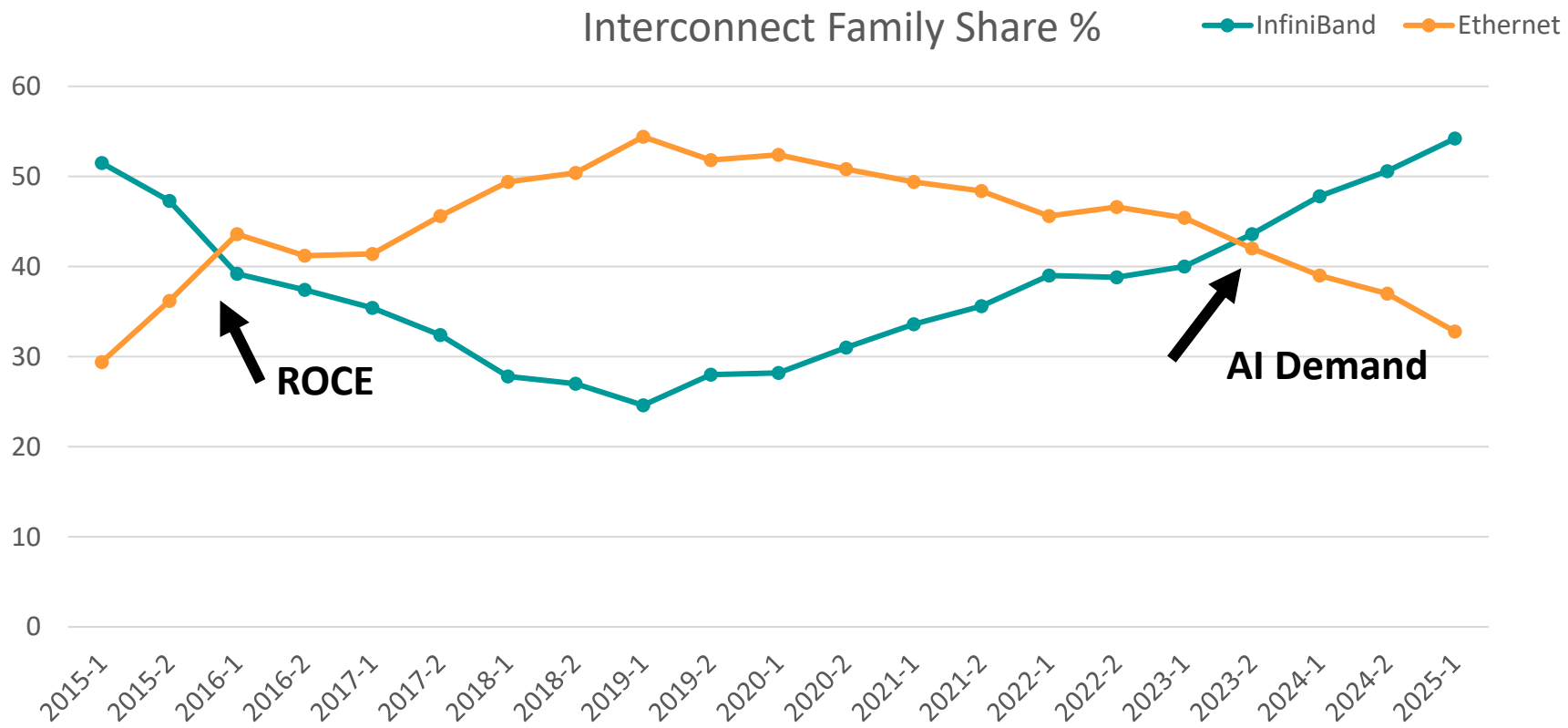
- 271 IB Clusters (54 %) in the Jun '25 Top500 list
  - (http://www.top500.org)
- Installations in the Top 50 (30 systems):

4,801,344-core (JUPITER Booster) @ Eviden/EuroHPC/FZJ in Germany (4th) – new	315-120-core (ABCI-Q) @ AIST in Japan (27th) – new
1,123,200-core (Eagle) @ Microsoft (4th)	223,088-core (Gefion) @ Danish Centre for AI Innovation in Denmark (29th)
1,824,768-core (Leonardo) @CINECA in Italy (10th)	555,520-core (Selene) @ NVIDIA (30th)
718,848-core (ISEG2)@Nebius AI (Netherlands) (13th) – new	185,712-core (SuperPOD), at NVIDIA (32nd)
663,040-core (MareNostrum 5 ACC),@BSC in Spain (11th)	445,440-core (Explorer-WUS3) @ Microsft Azure (USA) (34th) – new
479,232-core (ABCI 3.0),@AIST in Jaman (15th) – new	227,136-core (Jean Zay H100) @ CNRS/IDRIS-GENCI in France (35th) – new
485,888-core (Eos NVIDIA DGX SuperPOD),@NVIDIA (16th)	146,304-core (FPT AI Factory Japan) in Japan (36th) – new
349,440-core (SSC-24)@Samsung Electronics in Korea (18th) – new	221,952-core (Miyabi-G) @ JCAHPC in Japan (37th)
1,572,480-core (Sierra),@LLNL (20th)	142,240-core (FPT AI Factory Vietnam) in Vietnam (38th) – new
297,840-core (CHIE-3),@Softbank in Japan (23rd)	218,880-core (ISEG) @ Nebius AI in the Netherlands (39th) – new
297,840-core (CHIE-2),@Softbank in Japan (25th)	143,360-core (Ubilink) @ Ubilink in Taiwan (41st)
237,280-core (Njoerd) @ Northern Data Group in the UK (26th) – new	**and many more!**

Ethernet-based Scientific Computing Installations (Jun 2025)

- 167 Ethernet-based (1G, 10G, 25G, 50G, 100G, 200G, 400G, 800G) compute systems with ranking in the Jun '25 Top500 list
  - 11,039,616-core (El Capitan) using Slingshot-11 at LLNL (1st)
  - 8,699,904-core (Frontier) using Slingshot-11 at ORNL (2nd)
  - 4,742,808-core (Aurora) using Slingshot-11 at ANL (3rd)
  - 3,143,520-core (HPC6) using Slingshot-11 at Eni S.p.A., Italy (6th)
  - 2,121,600-core (Alps) using Slinshot-11 at CSCS, Switzerland (8th)
  - 2,752,704-core (LUMI) using Slingshot-11 at EuroHPC, Finland (9th)
  - 1,028,160-core (Isambard-AI phase 2) using Slingshot-11 at Univ. Bristol (11th) – new
  - 1,161,216-core (Tuolumne) using Slingshot-11 at LLNL (12th) – new
  - 822,528-core (Discovery 6) using Slingshot-11 at ExxonMobil (17th) – new
  - 481,440-core (Venado) using Slingshot-11 at LANL (19th)
  - 548,5332-core (CEA-HE) using BXI-v2 at CEA in France (24th)
  - 888,832-core (Perlmutter) using Slingshot-11 at NERSC (25th)
  - 383,040-core (El Dorado) using Slingshot-11 at SNL (28th)
  - 483,840-core (AI-03) using Broadcom NetXtreme-E at Core42 (UAE) (33rd) –new
  - 319,072-core (Adastra) Slingshot-11 at GENCI-CINES, France (40th)
  - 74,880-core (Israel-1) using NVIDIA Spectrum-X at NVIDIA Israel (44th) – new
  - 877,824-core (Shaheen III) Slingshot-11 at KAUST, Saudi Arabia (47th)
  - 181,440-core (Hunter) using Slingshot-11 at HLRS, Germany (54th) – new
  - 660,800-core (Crossroads) Slingshot-11 at LANL/SNL/NNSA/DOE (57th)
  - 181,248-core (Setonix) Slingshot-11 at Pawsey Supercomputing Centre, Australia (59th)
  - 232,000-core (Discovery 5) Slingshot-11 at ExxonMobil, USA (60th)
  - and many more!

Trends in Commodity Interconnects: Last 10 Years



Courtesy: TOP500.org

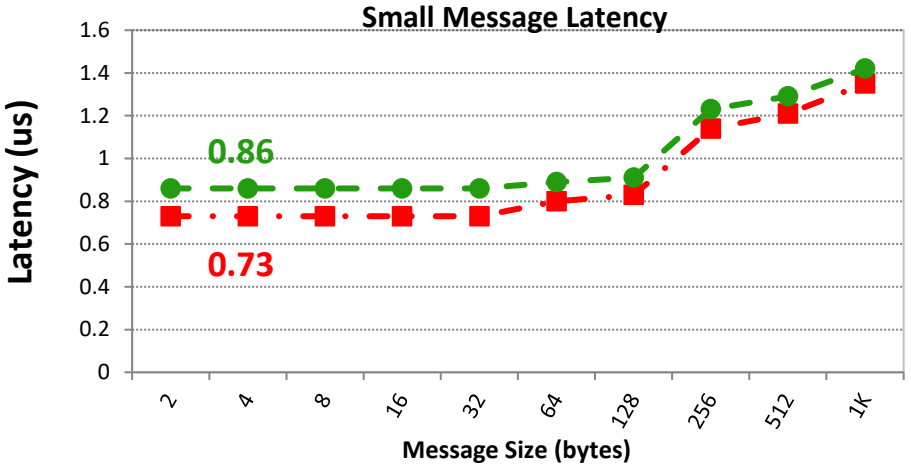
Presentation Overview

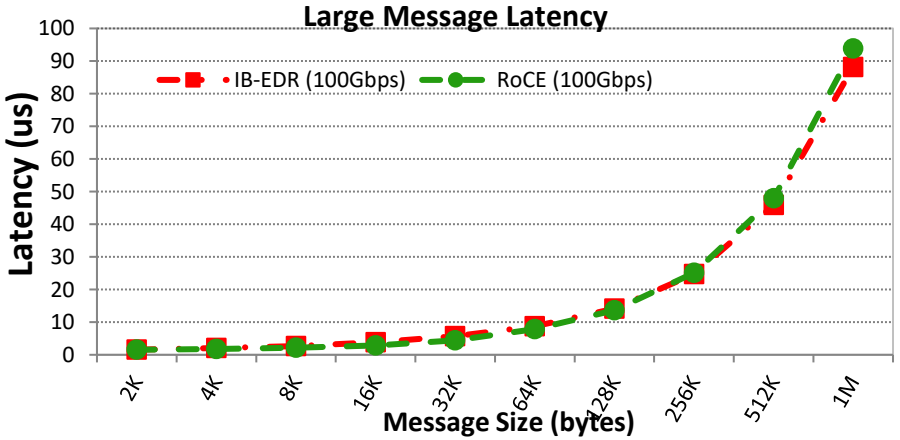
- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
- Overview of Emerging Smart Network Technologies
  - Collectives (NVIDIA SHARP)
  - Overview of SmartNIC Architecture
  - NVIDIA BlueField DPUs
  - AMD Pensando Smart NICs
  - Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras
  - Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- **Sample Case Studies and Performance Numbers**
- Hands on Exercises: IB Technologies and MPI Collectives
- Conclusions and Final Q&A

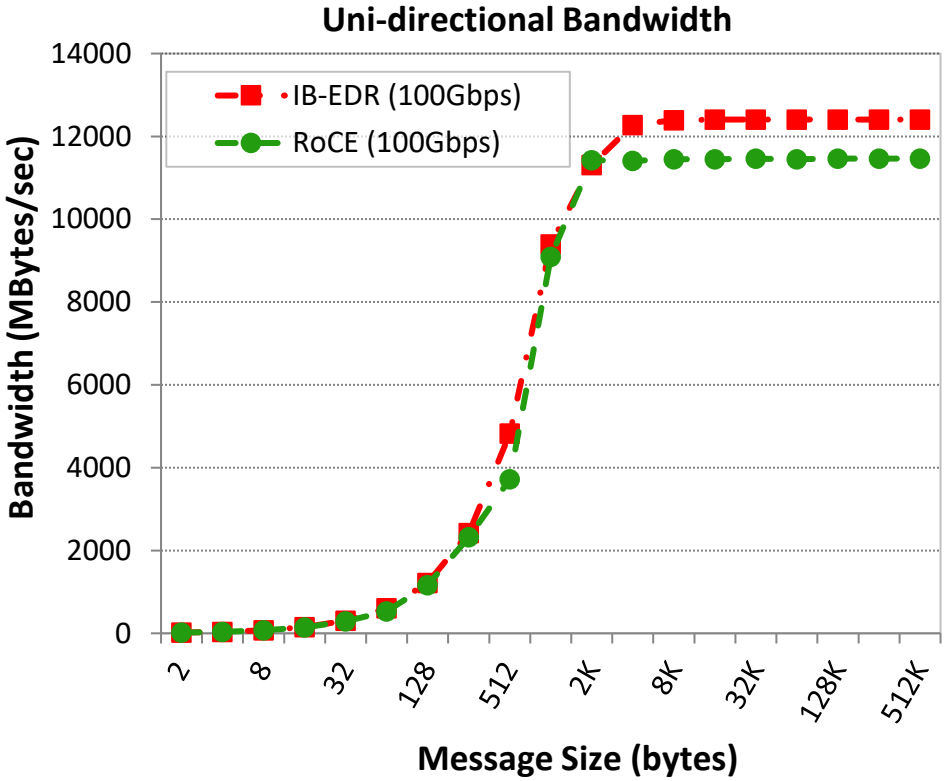
Case Studies

- **Low-level Performance**
- Message Passing Interface (MPI)

Low-level Latency and Uni-directional Bandwidth Measurements (IB-EDR v RoCE)

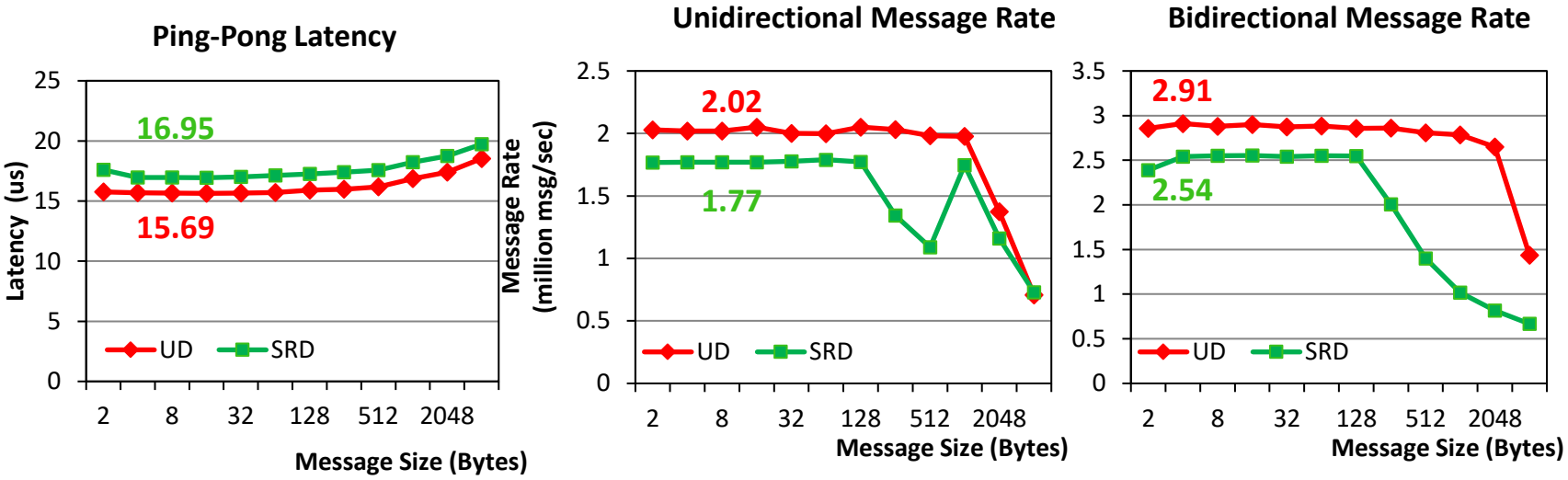






ConnectX-4 EDR (100 Gbps): 3.1 GHz Deca-core (Haswell) Intel Back-to-back
ConnectX-4 EN (100 Gbps): 3.1 GHz Deca-core (Haswell) Intel Back-to-back

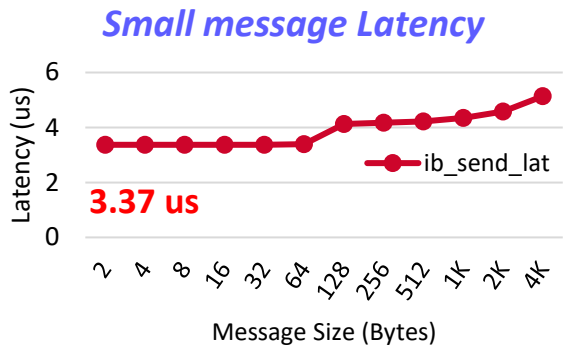
Verbs level evaluation of EFA performance

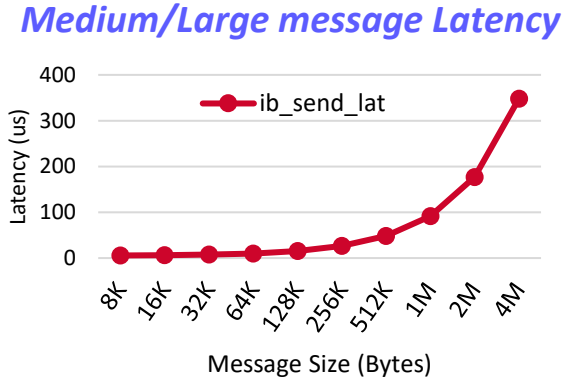


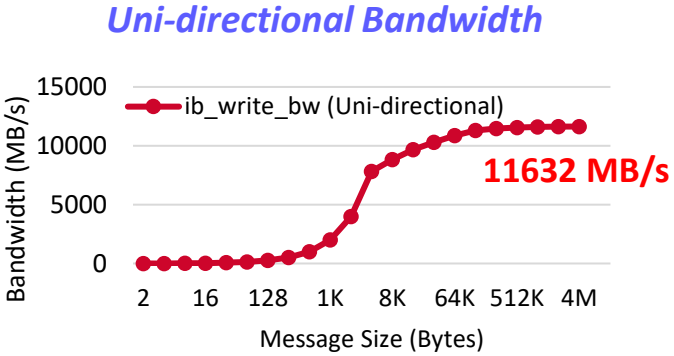
- SRD adds 8-10% overhead compared to UD
- Due to hardware based acks used for reliability

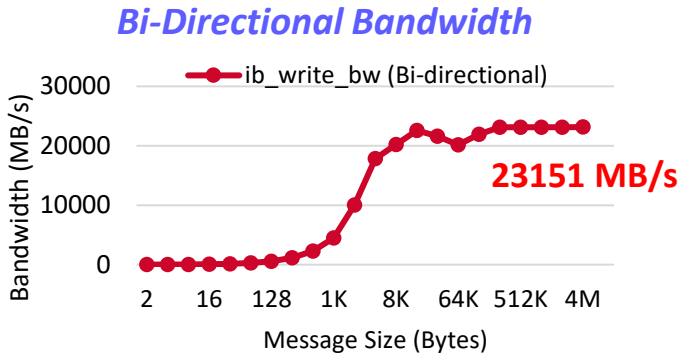
- Instance type: c5n.18xlarge
- CPU: Intel Xeon Platinum 8124M @ 3.00GHz

Verbs Level Evaluation of Broadcom RoCE adapters









Broadcom NetXtreme RoCE HCA (100 Gbps): 2 GHz AMD EPYC 7662 64-Core Processor

Case Studies

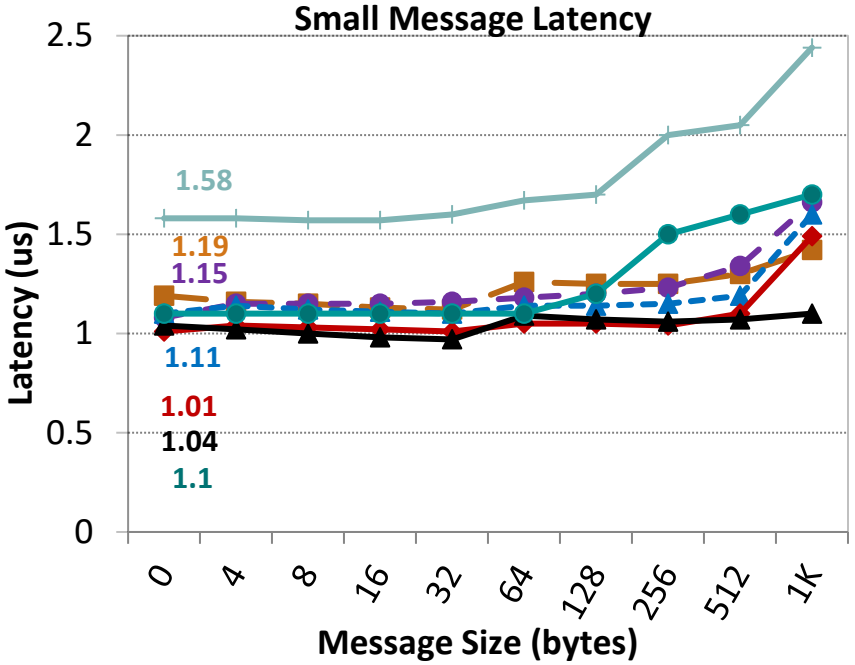
- Low-level Performance
- **Message Passing Interface (MPI)**

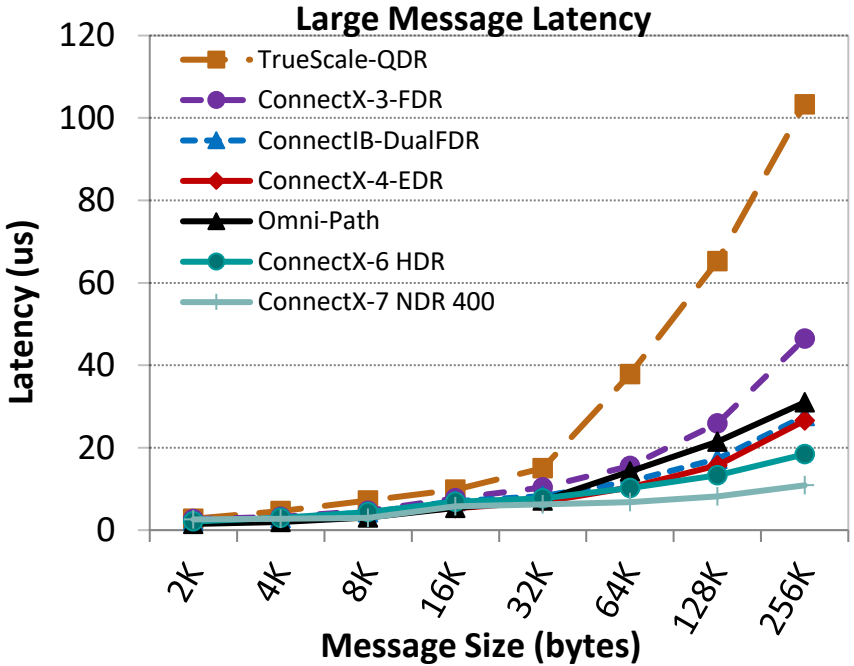
Overview of the MVAPICH Project

- High Performance open-source MPI Library
- Support for multiple interconnects
  - InfiniBand, Omni-Path, Ethernet/iWARP, RDMA over Converged Ethernet (RoCE), AWS EFA, OPX, Broadcom RoCE, Intel Ethernet, Rockport Networks, Slingshot 10/11
- 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-4.1 standard
- http://mvapich.cse.ohio-state.edu
- Additional optimized versions for different systems/environments:
  - MVAPICH-Plus (Unification of MVAPICH2-X and MVAPICH2-GDR), since 2023
  - 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 2004**
  - OSU InfiniBand Network Analysis and Monitoring (INAM), since 2015

- **Used by more than 3,450 organizations in 92 countries (listed under the Users Tab of the MVAPICH page)**
- **More than 1.93 Million downloads from the OSU site directly**
- Empowering many TOP500 clusters (June '25 ranking)
  - 21st , 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China
  - 67th , 448, 448 cores (Frontera) at TACC
  - 88th , 288,288 cores (Lassen) at LLNL
- Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, OpenHPC, and Spack)
- Partner in the 67th ranked TACC Frontera system
- **Empowering Top500 systems for more than 20+ years**

One-way Latency: MPI over IB with MVAPICH2





TrueScale-QDR - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with IB switch

ConnectX-3-FDR - 2.8 GHz 10-core (IvyBridge) Intel PCI Gen3 with IB switch

ConnectIB-Dual FDR - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with IB switch

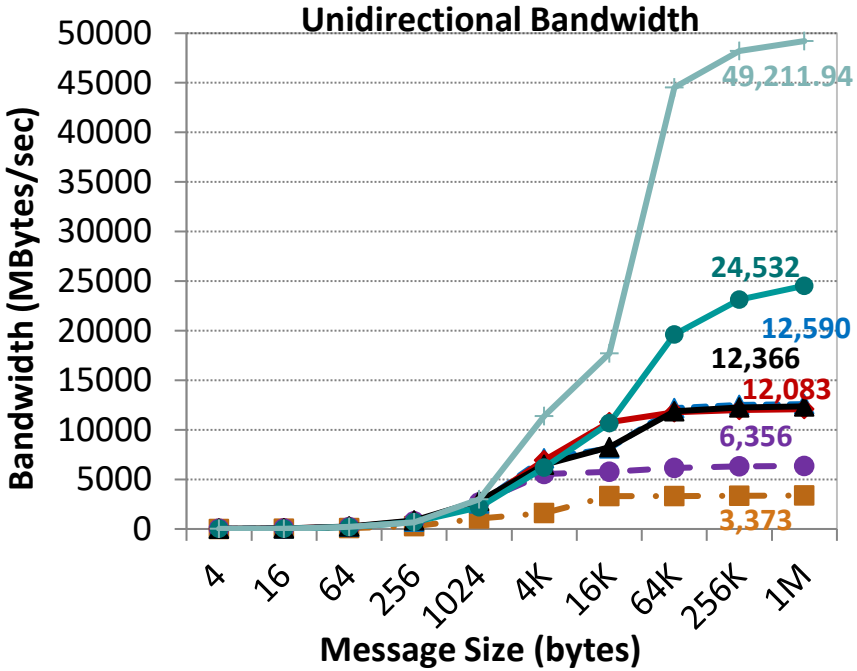
ConnectX-4-EDR - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with IB Switch

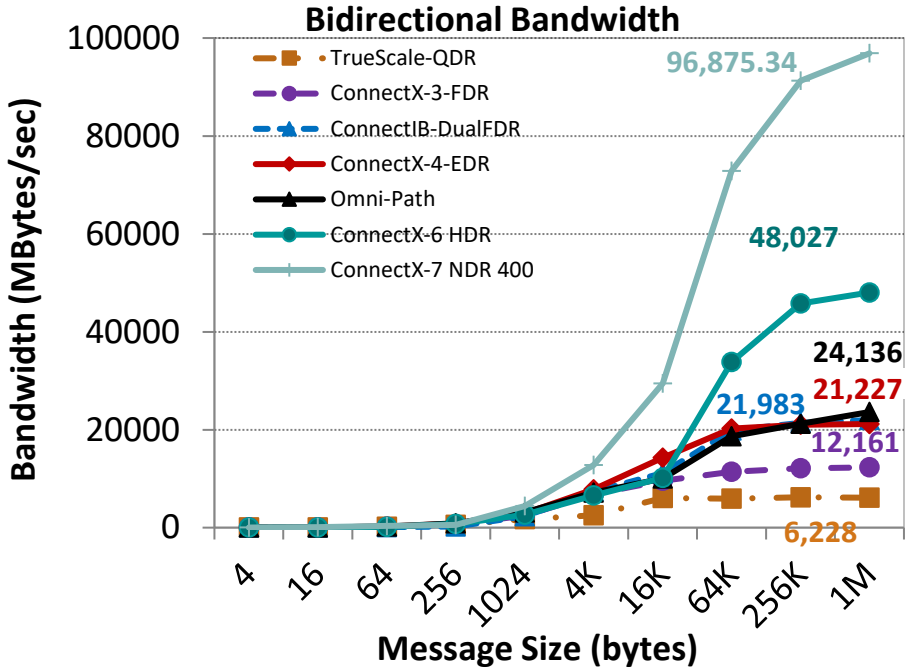
Omni-Path - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with Omni-Path switch

ConnectX-6-HDR - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with IB Switch

ConnectX-7-NDR - 3.4 GHz 72-core (Grace) ARM PCIe Gen5 with IB Switch

Bandwidth: MPI over IB with MVAPICH2





TrueScale-QDR - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with IB switch

ConnectX-3-FDR - 2.8 GHz 10-core (IvyBridge) Intel PCI Gen3 with IB switch

ConnectIB-Dual FDR - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with IB switch

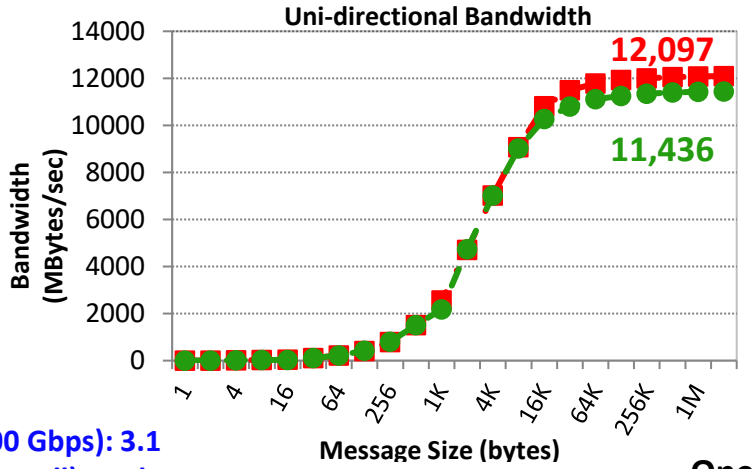
ConnectX-4-EDR - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with IB Switch

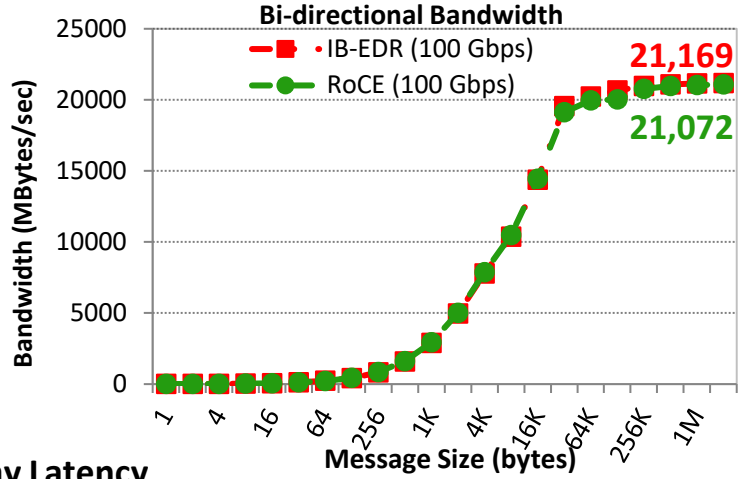
Omni-Path - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with Omni-Path switch

ConnectX-6-HDR - 3.1 GHz 10-core (Haswell) Intel PCI Gen3 with IB Switch

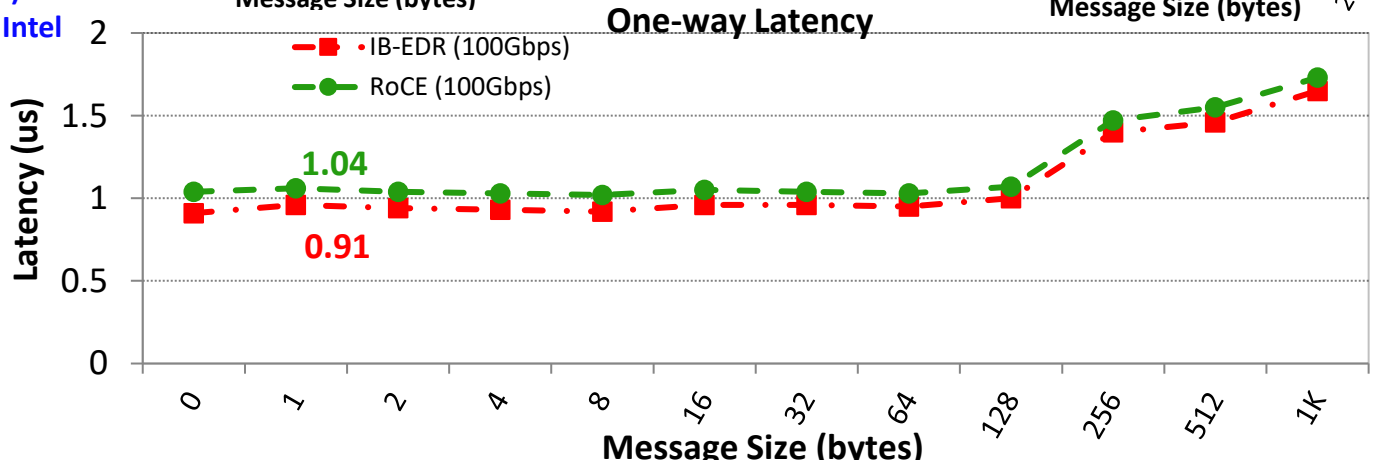
ConnectX-7-NDR - 3.4 GHz 72-core (Grace) ARM PCIe Gen5 with IB Switch

Convergent Technologies: MPI Latency and Uni-/Bi-directional BW



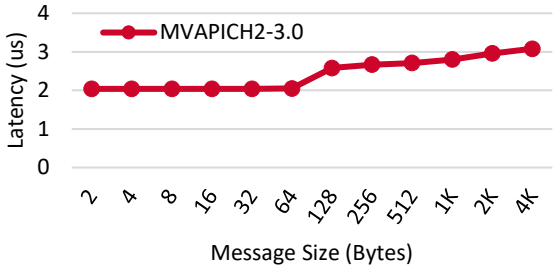


ConnectX-4 EDR (100 Gbps): 3.1 GHz Deca-core (Haswell) Intel Back-to-back

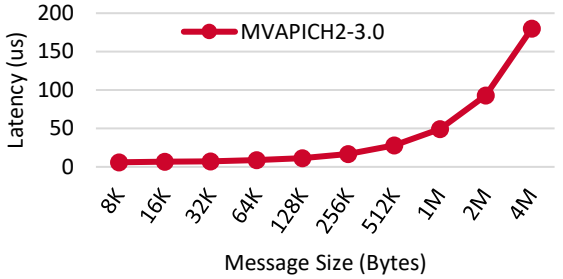


MPI Level Latency/Bandwidth on Slingshot 11

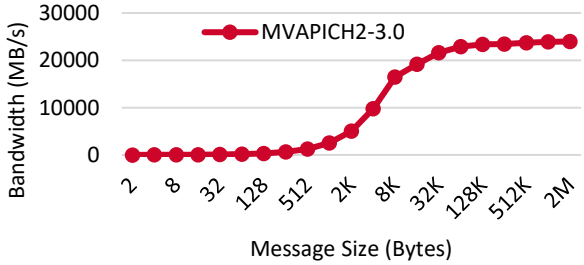
Small message Latency



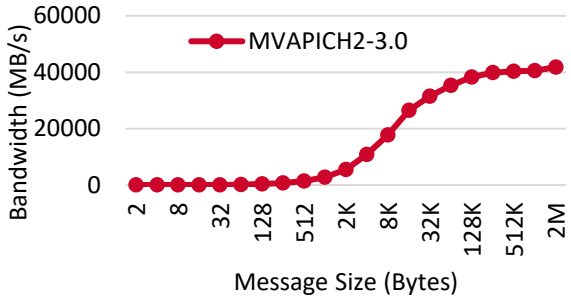
Medium/Large message Latency



Uni-directional Bandwidth



Bi-Directional Bandwidth

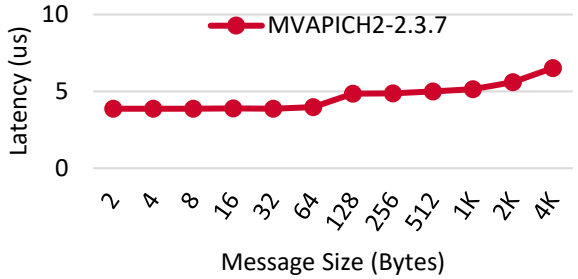


- **2us** inter-node point-to-point latency for small messages
- **23,985 MB/s** uni-directional peak bandwidth
- **42,034 MB/s** bi-directional peak bandwidth

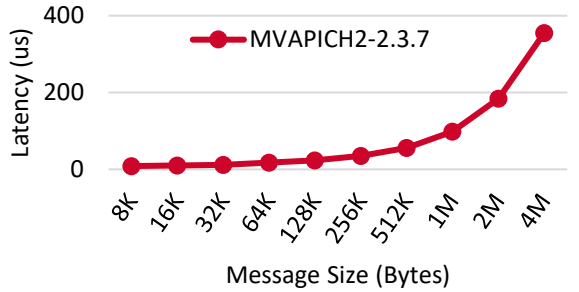
Interconnect : Cray HPE Slingshot 11, Library : MVAPICH2 3.0, CPU : AMD EPYC 7763 (milan) Processor

MPI Level Latency/Bandwidth on Broadcom RoCE

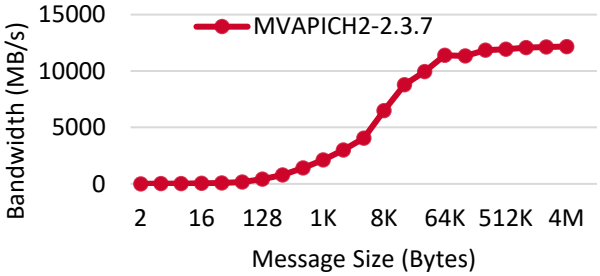
Small message Latency



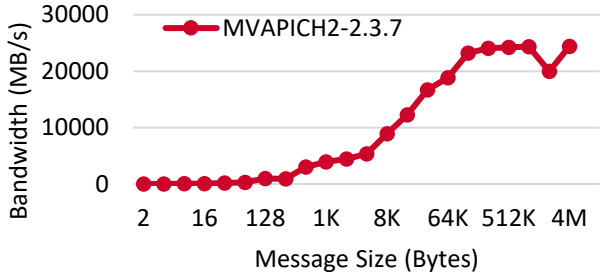
Medium/Large message Latency



Uni-directional Bandwidth



Bi-Directional Bandwidth



- 3.88us inter-node point-to-point latency for small messages
- **12,171 MB/s** uni-directional peak bandwidth
- **24,394 MB/s** bi-directional peak bandwidth

Interconnect : RoCE 100GbE with Broadcom NetXtreme Thor, Library : MVAPICH2 3.0, CPU : 2 GHz AMD EPYC 7662 64-Core Processor

Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
- Overview of Emerging Smart Network Technologies
  - Collectives (NVIDIA SHARP)
  - Overview of SmartNIC Architecture
  - NVIDIA BlueField DPUs
  - AMD Pensando Smart NICs
  - Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras
  - Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- **Hands on Exercises: IB Technologies and MPI Collectives**
- Conclusions and Final Q&A

Getting Set-up for the Hands-on Exercise 1

- You will run the experiments on the OSU RI2 cluster
- **Please use the account name and password from http://go.osu.edu/ibtutorial**

- Open your favorite Terminal

$ ssh ri2tut01@ri2.cse.ohio-state.edu

Enter Password:

- Today's examples are located in the training account home directories:
  - **/opt/tutorials/tutorial-ib**

- Verify files are present

```
[ri2tut01@head tutorial-ib]$ ls -l
total 28
-rw-r--r-- 1 root root 481 Nov 16 23:11 README.TXT
-rwxr-xr-x 1 root root 997 Nov 16 23:11 run_inline.sh
-rwxr-xr-x 1 root root 683 Nov 16 23:11 run_mtu.sh
-rwxr-xr-x 1 root root 395 Nov 16 23:11 run_omb.sh
-rwxr-xr-x 1 root root 498 Nov 16 23:11 run_perftest_bw.sh
-rwxr-xr-x 1 root root 501 Nov 16 23:11 run_perftest_lat.sh
-rwxr-xr-x 1 root root 796 Nov 16 23:11 run_rc_ud.sh
```

Step 2: Benchmarking InfiniBand Latency

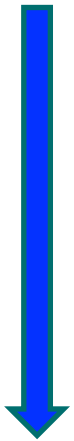
```
$ srun -N2 --reservation=ibtutorial run_perftest_lat.sh

Executing '/usr/bin/ib_send_lat -d mlx5_0 -a' on the server
Executing '/usr/bin/ib_send_lat -d mlx5_0 -a gpu08' on the client
. . . .
```

#bytes	#iterations	t_min[usec]	t_max[usec]	t_typical[usec]	t_avg[usec]	t_stdev[usec]	99% percentile[usec]	99.9% percentile[usec]
2	1000	0.94	3.00	0.98	0.98	0.04	1.08	3.00
4	1000	0.94	3.36	0.96	0.98	0.04	1.04	3.36
8	1000	0.95	2.96	0.99	0.99	0.04	1.09	2.96
16	1000	0.95	2.83	0.97	0.98	0.04	1.01	2.83
32	1000	1.02	2.60	1.04	1.05	0.00	1.14	2.60
64	1000	1.00	2.15	1.03	1.04	0.00	1.07	2.15
128	1000	1.07	3.09	1.12	1.12	0.03	1.14	3.09
256	1000	1.45	3.21	1.53	1.54	0.03	1.67	3.21
512	1000	1.51	4.22	1.54	1.56	0.03	1.69	4.22
1024	1000	1.64	3.45	1.67	1.69	0.03	1.79	3.45….

Benchmarking InfiniBand Latency

#bytes	ib_send_lat	ib_write_lat	ib_read_lat
2	1.01	0.94	1.93
4	1.01	0.94	1.96
8	1.01	0.95	1.97
16	1.02	0.95	1.98
32	1.02	1.02	1.98
64	1.12	1	2
128	1.16	1.07	2.05
256	1.60	1.45	2.1
512	1.64	1.51	2.14
1024	1.78	1.64	2.4
2048	2.01	1.88	2.64
4096	2.48	2.34	3.12
8192	3.16	3.02	3.79
16384	4.34	4.19	5.03
32768	6.41	6.25	7.05



Latency Increases with Message Size

Step 3: Benchmarking InfiniBand Bandwidth

```
$ srun -N2 --reservation=ibtutorial run_perftest_bw.sh

Executing '/usr/bin/ib_send_bw -d mlx5_0 -a' on the server
Executing '/usr/bin/ib_send_bw -d mlx5_0 -a gpu08' on the client
. . . .
#bytes     #iterations    BW peak[MB/sec]    BW average[MB/sec]   MsgRate[Mpps]
 2           1000             12.01              11.85              6.210454
 4           1000             22.17              17.74              4.650813
 8           1000             45.55              35.99              4.716851
 16          1000             90.87              71.80              4.705226
 32          1000             179.96             143.74             4.710197
 64          1000             362.58             286.97             4.701794
 128         1000             726.98             573.51             4.698231
 256         1000             1457.57            1111.64            4.553276
 512         1000             2872.28            2281.59            4.672705
 1024        1000             5488.89            4384.14            4.489363
```

Benchmarking InfiniBand Bandwidth

#Bytes	ib_send_bw	ib_write_bw	ib_read_bw
2	11.59	11.68	12.01
4	15.84	23.78	22.17
8	45.1	47.07	45.55
16	90.2	93.66	90.87
32	176.06	188.28	179.96
64	351.28	377.54	362.58
128	719.83	751.2	726.98
256	1385.21	1498.58	1457.57
512	2694	2974.33	2872.28
1024	5540.84	5888.88	5488.89
2048	9664.95	9868.51	6823.24
4096	10393.64	10825.72	6304.71
8192	11140.84	11180.81	9600.63
16384	11197.53	11363.68	9847.76
32768	11274.91	11295.33	10802.41

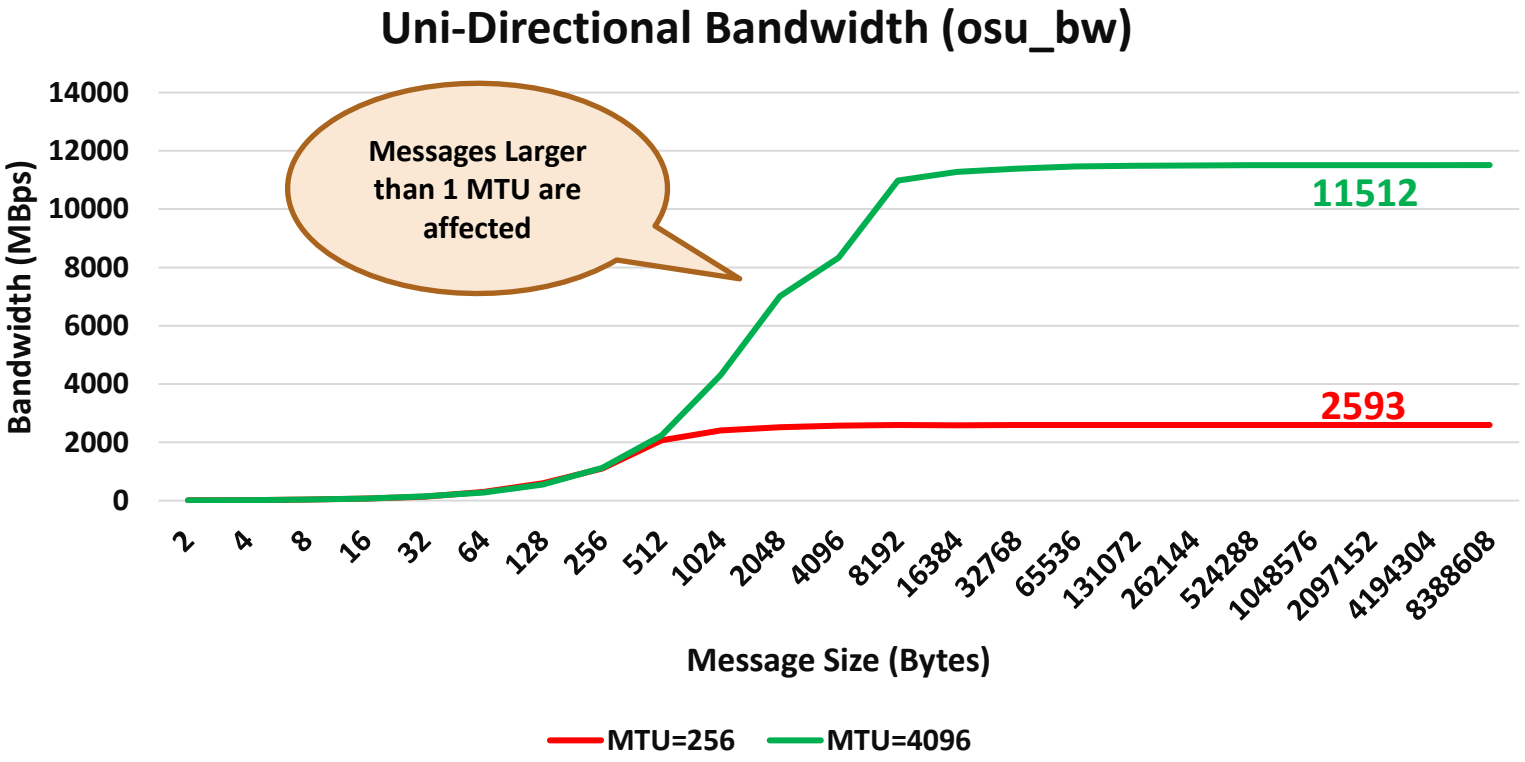
Impact of Maximum Transmission Unit (MTU)

```
$ srun -N2 --reservation=ibtutorial run_mtu.sh 256 4096

IB Read Bandwidth Test using MTU=256
Executing '/usr/bin/ib_read_bw -d mlx5_0 -a -m 256 ' on the server
Executing '/usr/bin/ib_read_bw -d mlx5_0 -a -m 256 gpu05' on the client
...

IB Read Bandwidth Test using MTU=4096
Executing '/usr/bin/ib_read_bw -d mlx5_0 -a -m 4096 ' on the server
Executing '/usr/bin/ib_read_bw -d mlx5_0 -a -m 4096 gpu05' on the client
...
```

Impact of Maximum Transmission Unit (MTU)



ConnectX-5 EDR (100 Gbps), 2 x 14-core Intel Xeon CPU E5-2680 v4 2.40 GHz

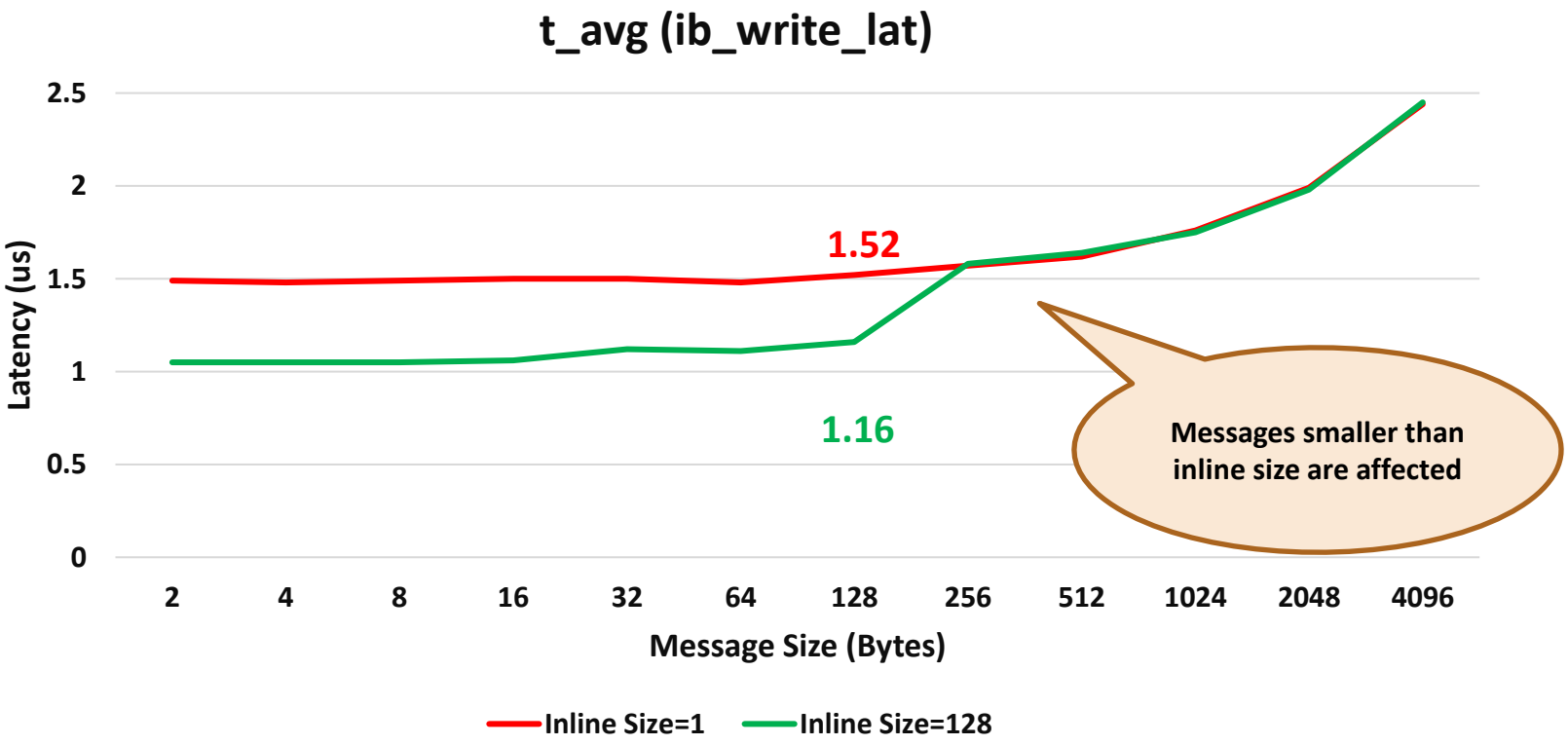
Impact of Changing Inline Size: IB

```
$ srun -N2 --reservation=ibtutorial run_inline_ib.sh 0
128 #(Can explore with other values as well)

IB Write Latency using Inline=0
Executing '/usr/bin/ib_write_lat -d mlx5_0 -a -I 0 ' on the server
Executing '/usr/bin/ib_write_lat -d mlx5_0 -a -I 0 gpu05' on the client
...

IB Write Latency using Inline=128
Executing '/usr/bin/ib_write_lat -d mlx5_0 -a -I 128 ' on the server
Executing '/usr/bin/ib_write_lat -d mlx5_0 -a -I 128 gpu05' on the client
...
```

Impact of Changing Inline Size



ConnectX-5 EDR (100 Gbps), 2 x 14-core Intel Xeon CPU E5-2680 v4 2.40 GHz

Benchmarking MPI Performance

```
$ srun -N2 --reservation=ibtutorial run_omb.sh

+ /opt/mvapich2/mvapich2-2.3.7/bin/mpirun_rsh -np 2 gpu05 gpu06 \
MV2_HOMOGENEOUS_CLUSTER=1 MV2_IBA_HCA=mlx5_0 \
/opt/mvapich2/mvapich2-2.3.7/libexec/osu-micro-benchmarks/mpi/pt2pt/osu_latency
# OSU MPI Latency Test v5.9
# Size          Latency (us)
0                       1.21
1                       1.23
2                       1.21
4                       1.19
8                       1.17
16                      1.19
32                      1.18
64                      1.20
128                     1.26
```

Benchmarking MPI Performance

#Bytes	Latency (us)	Bandwidth (MB/s)	Bi-Bandwidth(MB/s)
1	1.21	5.43	4
2	1.23	10.69	8.17
4	1.21	21.08	16.75
8	1.19	41.79	34.59
16	1.17	82.29	71.42
32	1.19	163.06	162.32
64	1.18	353.42	518.72
128	1.2	655.47	665.11
256	1.26	1333.58	1875.53
512	1.66	2291.8	3225.89
1024	1.73	3569.84	4724.21
2048	1.88	5250.27	6284.36
4096	2.22	6806.01	7664.77
8192	2.93	9669.86	10540.81
16384	4.37	7736.91	11650.45

Latency is close to IB Latency

MPI Bandwidth for large messages is close to Line Rate

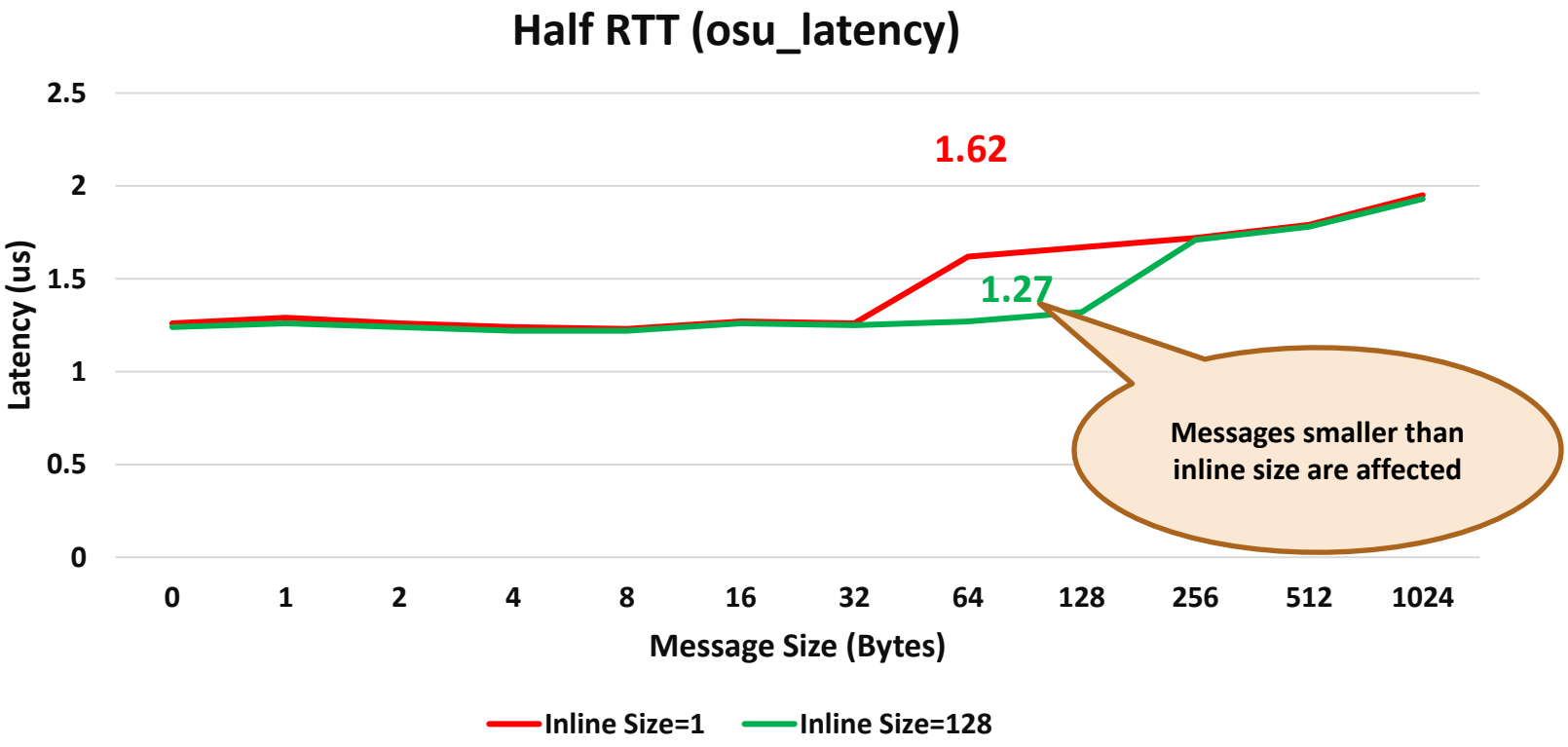
Impact of Changing Inline Size: MPI

```
$ srun -N2 --reservation=ibtutorial run_inline_mpi.sh 0
128 #(Can explore with other values as well)

MPI osu_latency Latency using Inline=0
MPI Send/Recv Latency using Inline=0 on hosts gpu05 and gpu06
...

MPI osu_latency Latency using Inline=128
MPI Send/Recv Latency using Inline=128 on hosts gpu05 and gpu06
...
```

Impact of Changing Inline Size



ConnectX-5 EDR (100 Gbps), 2 x 14-core Intel Xeon CPU E5-2680 v4 2.40 GHz

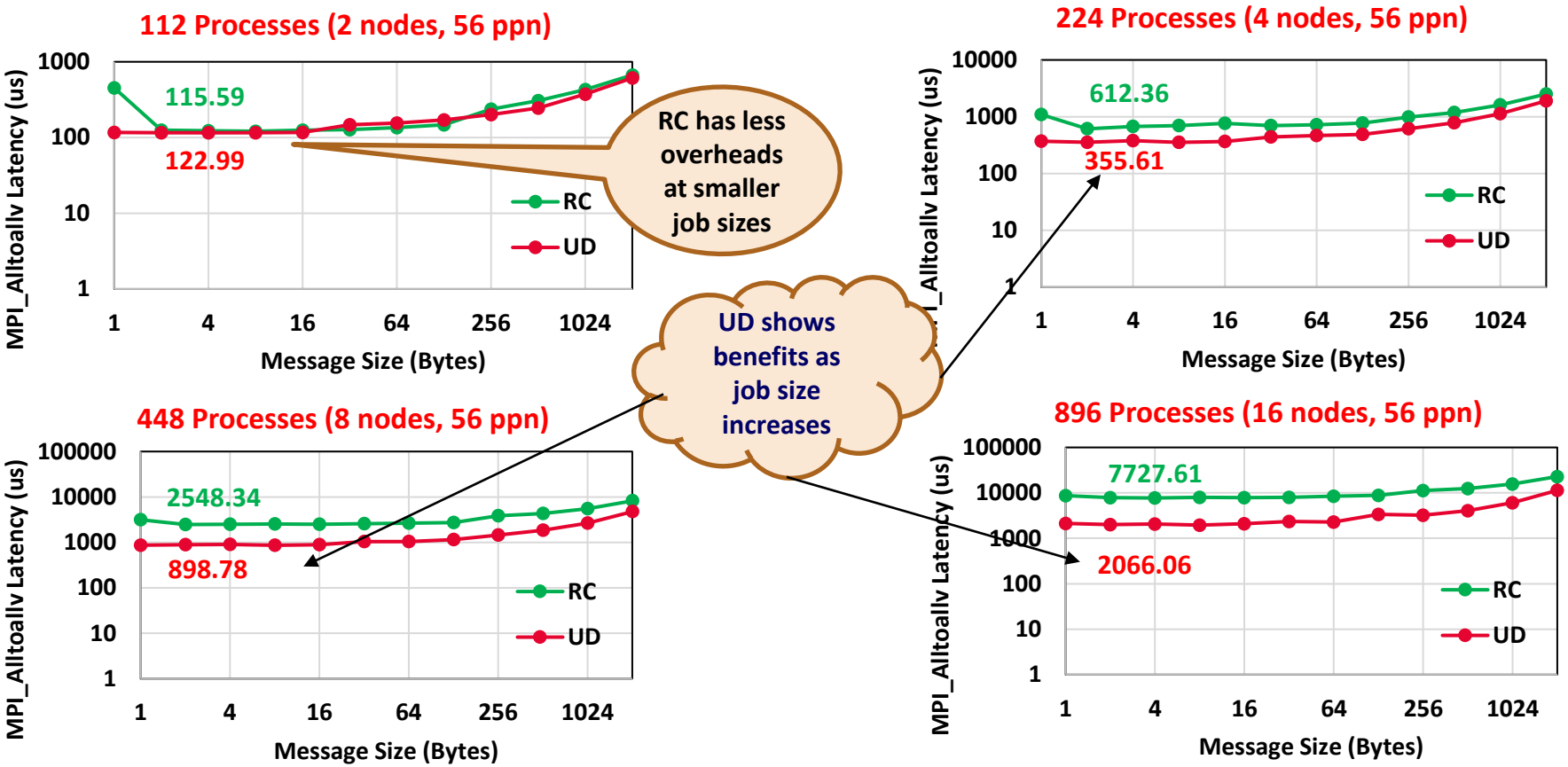
MPI Collective Performance with RC vs. UD

```
$ srun -N4 --reservation=ibtutorial run_rc_ud.sh

Running MPI_Alltoallv with 112 processes with RC
+ /opt/mvapich2/mvapich2-2.3.7/bin/mpirun_rsh -np 112 -hostfile $HOME/hosts-10291
MV2_HOMOGENEOUS_CLUSTER=1 MV2_IBA_HCA=mlx5_0 MV2_USE_ONLY_UD=0
/opt/mvapich2/mvapich2-2.3.7/libexec/osu-micro-
benchmarks/mpi/collective/osu_alltoallv -m 2048
...

Running MPI_Alltoallv with 112 processes with UD
+ /opt/mvapich2/mvapich2-2.3.7/bin/mpirun_rsh -np 112 -hostfile $HOME/hosts-10291
MV2_HOMOGENEOUS_CLUSTER=1 MV2_IBA_HCA=mlx5_0 MV2_USE_ONLY_UD=1
MV2_UD_MAX_RECV_WQE=512 MV2_UD_MAX_SEND_WQE=512 /opt/mvapich2/mvapich2-
2.3.7/libexec/osu-micro-benchmarks/mpi/collective/osu_alltoallv -m 2048
...
```

MPI Collective Performance with RC vs. UD



ConnectX-6-HDR100 (100 Gbps), 2x2.70 GHz 28-core Intel Cascade Lake with IB (HDR) switches

Presentation Overview

- Introduction
- Why High-Performance Networking for HPC and AI?
- Communication Model and Semantics of High-Performance Networks
- Architectural Overview of High-Performance Networks
- Overview of Emerging Smart Network Technologies
  - Collectives (NVIDIA SHARP)
  - Overview of SmartNIC Architecture
  - NVIDIA BlueField DPUs
  - AMD Pensando Smart NICs
  - Intel Columbiaville IPUs
- High-Performance Network Deployments for AI Workloads
  - Cerebras
  - Habana-Gaudi
- Overview of Software Stacks for Commodity High-Performance Networks
- Sample Case Studies and Performance Numbers
- Hands on Exercises: IB Technologies and MPI Collectives
- **Conclusions and Final Q&A**

Concluding Remarks

- Presented smart networking architectures & trends in Clusters
- Presented background and details of various smart network for HPC
  - Highlighted the main features of high-performance networks
  - Gave an overview of high-performance network hardware/software ecosystem
  - Discussed sample performance numbers in designing various high-end systems
- Smart networking architectures are leading to a new generation of networked computing systems, opening many research issues needing novel solutions
- Will see many more innovations in the coming years for exascale/zetascale systems

Funding Acknowledgments

Funding Support by

Equipment Support by

Acknowledgments to all the Heroes (Past/Current Students and Staffs)

Current Students (Under/Graduate)

- N. Alnaasan (Ph.D.)
- Q. Anthony (Ph.D.)
- C.-C. Chen (Ph.D.)
- T. Chen (Ph.D.)
- N. Contini (Ph.D.)
- S. Gumaste (Ph.D.)
- J. Hatef (Ph.D.)
- G. Kuncham (Ph.D.)
- S. Lee (Ph.D.)
- B. Michalowicz (Ph.D.)
- J. Oswal (Ph.D.)
- T. Tran (Ph.D.)
- L. Xu (P.h.D.)
- S. Xu (Ph.D.)
- J. Yao (Ph.D.)
- S. Zhang (Ph.D.)
- S. Mohammad (M.S.)
- B. Lampe (B.S.)
- N. Klein (B.S.)

Current Research Specialist

- R. Motlagh

Current Software Engineers

- N. Shineman
- M. Lieber

Past Research Scientists

- K. Hamidouche
- S. Sur
- X. Lu
- M. Abduljabbar
- A. Shafi

Past Students

- A. Awan (Ph.D.)
- A. Augustine (M.S.)
- P. Balaji (Ph.D.)
- M. Bayatpour (Ph.D.)
- R. Biswas (M.S.)
- S. Bhagvat (M.S.)
- A. Bhat (M.S.)
- D. Buntinas (Ph.D.)
- L. Chai (Ph.D.)
- B. Chandrasekharan (M.S.)
- S. Chakraborthy (Ph.D.)
- N. Dandapanthula (M.S.)
- V. Dhanraj (M.S.)
- C.-H. Chu (Ph.D.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- R. Gulhane (M.S.)
- J. Hashmi (Ph.D.)
- M. Han (M.S.)
- W. Huang (Ph.D.)
- A. Jain (Ph.D.)
- J. Jani (M.S.)
- W. Jiang (M.S.)
- J. Jose (Ph.D.)
- M. Kedia (M.S.)
- K. S. Khorassani (Ph.D.)
- S. Kini (M.S.)
- M. Koop (Ph.D.)
- P. Kousha (Ph.D.)
- K. Kulkarni (M.S.)
- R. Kumar (M.S.)
- S. Krishnamoorthy (M.S.)
- K. Kandalla (Ph.D.)
- M. Li (Ph.D.)
- P. Lai (M.S.)
- J. Liu (Ph.D.)
- M. Luo (Ph.D.)
- A. Mamidala (Ph.D.)
- G. Marsh (M.S.)
- V. Meshram (M.S.)
- A. Moody (M.S.)
- S. Naravula (Ph.D.)
- R. Noronha (Ph.D.)
- X. Ouyang (Ph.D.)
- S. Pai (M.S.)
- S. Potluri (Ph.D.)
- J. Queiser (M.S.)
- K. Raj (M.S.)
- R. Rajachandrasekar (Ph.D.)
- B. Ramesh (Ph.D.)
- D. Shankar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- N. Sarkauskas (B.S. and M.S)
- V. Sathu (M.S.)
- N. Senthil Kumar (M.S.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Srivastava (M.S.)
- H. Subramoni (Ph.D.)
- S. Sur (Ph.D.)
- K. K. Suresh (Ph.D.)
- K. Vaidyanathan (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)
- J. Zhang (Ph.D.)
- Q. Zhou (Ph.D.)
- N. Chmura (B.S.)

Past Faculty

- H. Subramoni

Past Senior Research Associate

- J. Hashmi

Past Programmers

- A. Reifsteck
- D. Bureddy
- J. Perkins
- B. Seeds
- A. Guptha
- N. Pavuk

Past Post-Docs

- D. Banerjee
- X. Besseron
- M. S. Ghazimirsaeed
- H.-W. Jin
- J. Lin
- M. Luo
- E. Mancini
- K. Manian
- S. Marcarelli
- A. Ruhela
- J. Vienne
- H. Wang

Past Research Specialist

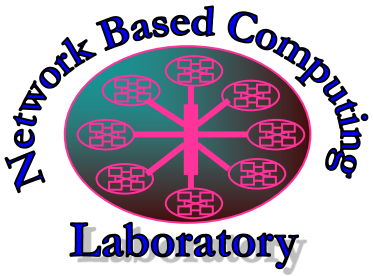
- M. Arnold
- J. Smith

Thank You!

panda@cse.ohio-state.edu, subramoni.1@osu.edu, michalowicz.2@osu.edu

Follow us on

https://x.com/mvapich



Network-Based Computing Laboratory

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

The High-Performance MPI/PGAS Project

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

The High-Performance Big Data Project

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

The High-Performance Deep Learning Project

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