MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

MVAPICH Touches the Cloud New Frontiers for MPI in High Performance Clouds

Presenter: Shulei Xu

xu.2452@osu.edu

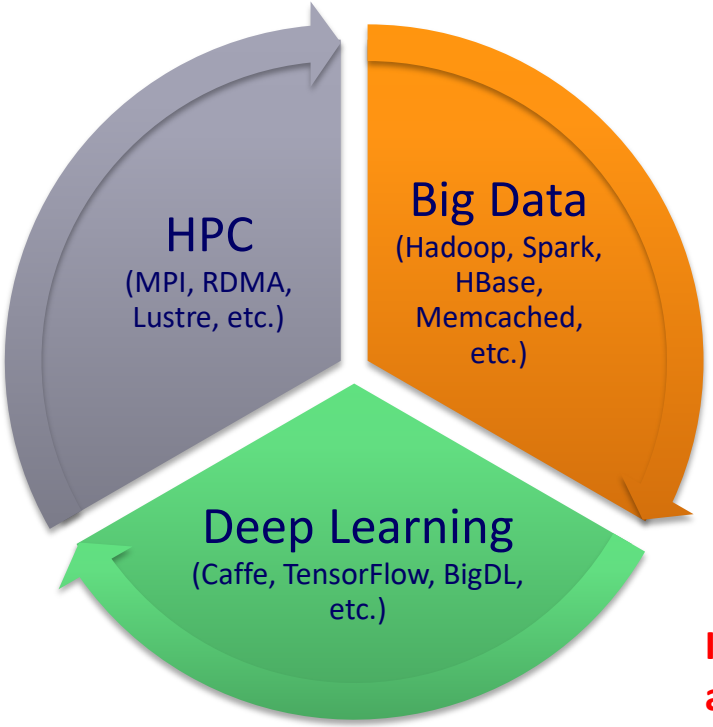
Network Based Computing Laboratory (NBCL)

The Ohio State University

Agenda

- **Introduction**
- Support to AWS EFA
  - Overview of AWS EFA
  - Designing MPI Libraries for EFA
  - Experimental Evaluations
- Support to Azure VM
  - Dedicated Performance Evaluation & Tuning
  - One-click quick deployment

Increasing Usage of HPC, Big Data and Deep Learning



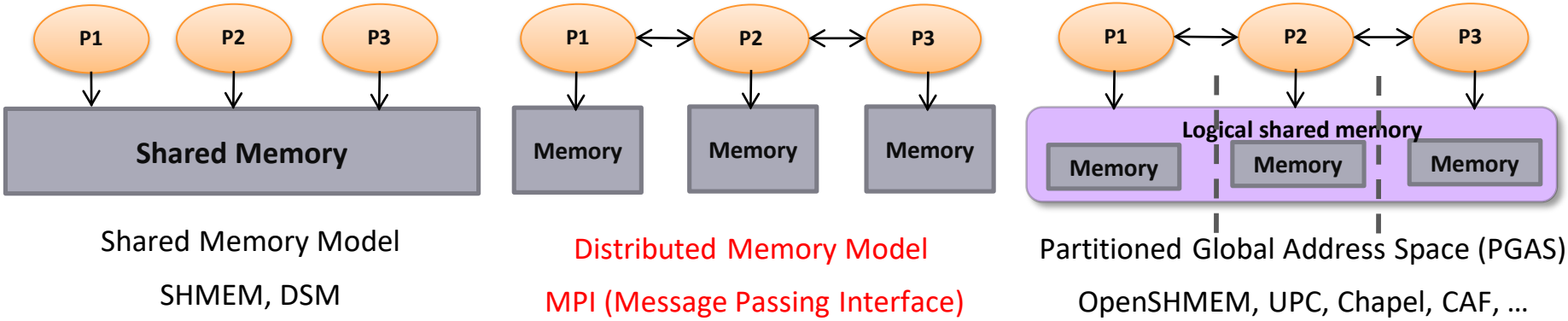
Convergence of HPC, Big Data, and Deep Learning!

Increasing Need to Run these applications on the Cloud!!

HPC, Deep Learning, Big Data and Cloud

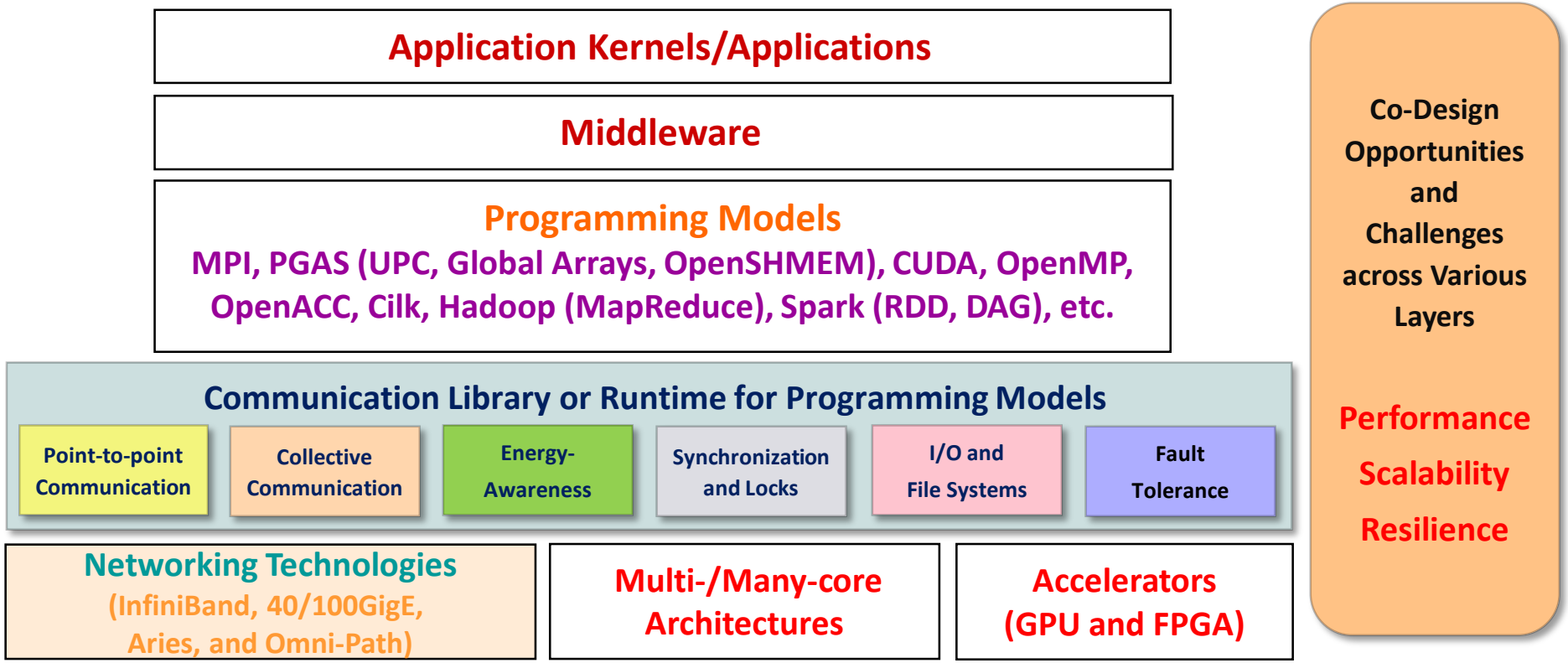
- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Performance Evaluation on Azure HB and HC Systems
  - Exploiting Accelerators (NVIDIA GPGPUs)
- Big Data/Enterprise/Commercial Computing
  - Spark and Hadoop (HDFS, HBase, MapReduce)
- Deep Learning
  - Caffe, CNTK, TensorFlow, and many more
- Cloud for HPC
  - Virtualization and Live Migration
- InfiniBand Network Analysis and Monitoring (INAM) Tool

Parallel Programming Models Overview



- Programming models provide abstract machine models
- Models can be mapped on different types of systems
  - e.g. Distributed Shared Memory (DSM), MPI within a node, etc.
- PGAS models and Hybrid MPI+PGAS models are gradually receiving importance

Supporting Programming Models for Multi-Petaflop and Exaflop Systems: Challenges



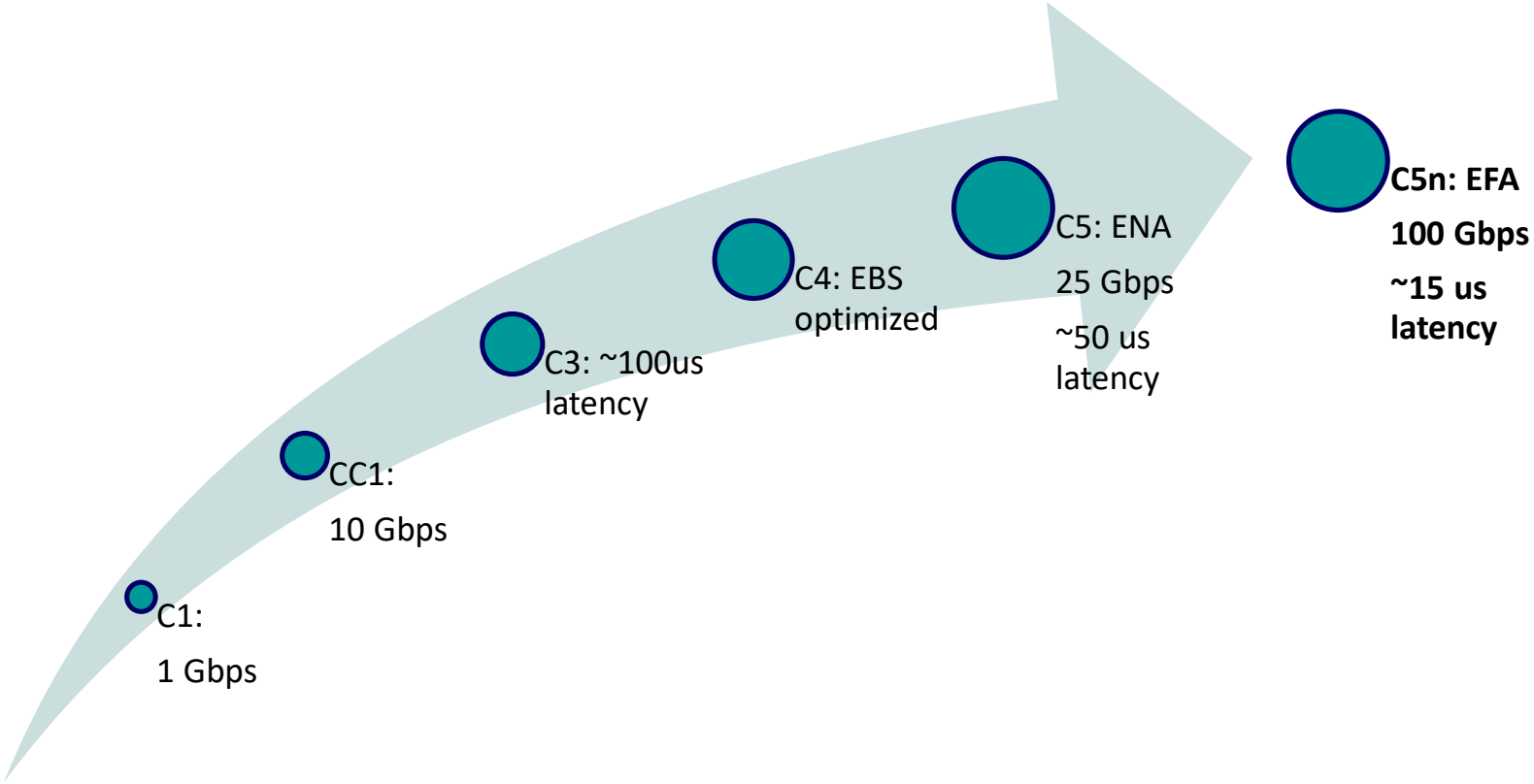
Broad Challenges in Designing Runtimes for (MPI+X) at Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
  - Scalable job start-up
  - Low memory footprint
- Scalable Collective communication
  - Offload
  - Non-blocking
  - Topology-aware
- Balancing intra-node and inter-node communication for next generation nodes (128-1024 cores)
  - Multiple end-points per node
- Support for efficient multi-threading
- Integrated Support for Accelerators (GPGPUs and FPGAs)
- Fault-tolerance/resiliency
- QoS support for communication and I/O
- Support for Hybrid MPI+PGAS programming (MPI + OpenMP, MPI + UPC, MPI + OpenSHMEM, MPI+UPC++, CAF, …)
- Virtualization
- Energy-Awareness

Agenda

- Introduction
- Support to AWS EFA
  - **Overview of AWS EFA**
  - Designing MPI Libraries for EFA
  - Experimental Evaluations
- Support to Azure VM
  - Dedicated Performance Evaluation & Tuning
  - One-click quick deployment

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)
- Allows OS bypass, up to 100 Gbps bandwidth
- 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)

Scalable Reliable Datagrams (SRD): Features & Limitations

Feature	UD	SRD
Send/Recv	✓	✓
Send w/ Immediate	✗	✗
RDMA Read/Write/Atomic	✗	✗
Scatter Gather Lists	✓	✓
Shared Receive Queue	✗	✗
Reliable Delivery	✗	✓
Ordering	✗	✗
Inline Sends	✗	✗
Global Routing Header	✓	✗
Max Message Size	4KB	8KB

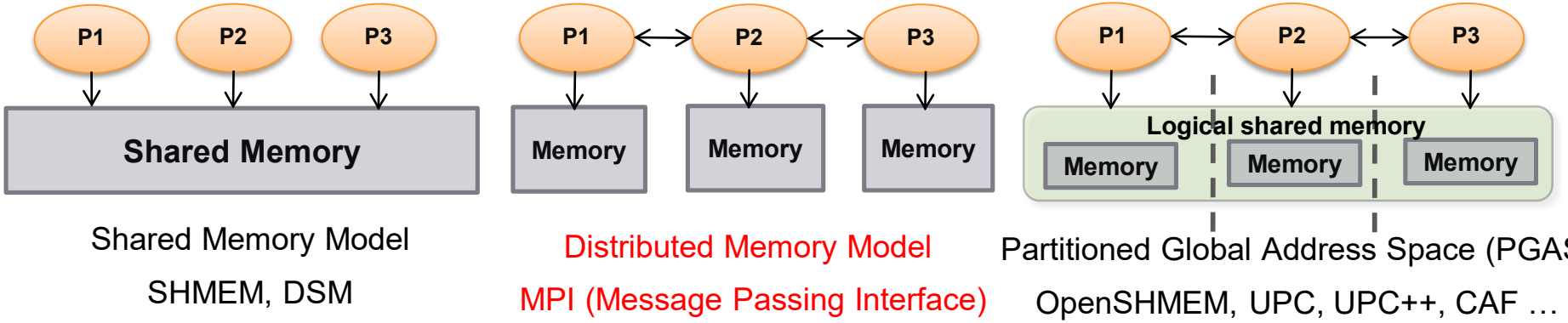
- Similar to IB Reliable Datagram
  - No limit on number of outstanding messages per context
- Out of order delivery
  - No head-of-line blocking
  - Bad fit for MPI, can suit other workloads
- Packet spraying over multiple ECMP paths
  - No hotspots
  - Fast and transparent recovery from network failures
- Congestion control designed for large scale
  - Minimize jitter and tail latency

Amazon Elastic Fabric Adapter: Anatomy, Capabilities, and the Road Ahead, Raghu Raja, OpenFabrics Workshop 2019

Agenda

- Introduction
- Support to AWS EFA
  - Overview of AWS EFA
  - **Designing MPI Libraries for EFA**
  - Experimental Evaluations
- Support to Azure VM
  - Dedicated Performance Evaluation & Tuning
  - One-click quick deployment

Designing MPI libraries for EFA



- MPI offer various communication primitives
  - Point-to-point, Collective, Remote Memory Access
  - Provides strict guarantees about reliability and ordering
  - Allows message sizes much larger than allowed by the network
- How to address these semantic mismatches between the network and programming model in a scalable and high-performance manner?

Challenge 1: Reliable and in-order delivery

- MPI guarantees reliable and in-order message matching to applications
- UD does not provide reliability or ordering
- SRD provides reliability but not in-order delivery

- Solution: use acknowledgements and retransmissions for reliability
- Piggy back acks on application messages for reducing overhead
- Use sequence number and sliding window for re-ordering packets at the receiver process

M. J. Koop, S. Sur, Q. Gao, and D. K. Panda, "High Performance MPI Design using Unreliable Datagram for Ultra-scale InfiniBand Clusters," in *Proceedings of the 21st annual international conference on Supercomputing*.

Challenge 2: Zero-copy transmission of large messages

- MPI allows sending and receiving very large messages
- Network message size bound by MTU size (4KB for UD, 8KB for SRD)
- Need to handle segmentation and reassembly
- Existing zero-copy designs* can not be used
  - Utilizes send-with-immediate for sequence numbers (not supported by EFA)
  - Retransmits entire message if out-of-order arrival is detected
- Solution: propose new design for zero-copy rendezvous transfers
  - Maintain a pool of dedicated QPs for zero-copy transfers
  - Use scatter gather lists for sequence numbers
  - Reorder out-of-order packets at the receiver

* M. J. Koop, S. Sur, and D. K. Panda, "Zero-copy Protocol for MPI using Infiniband Unreliable Datagram," in *2007 IEEE International Conference on Cluster Computing*.

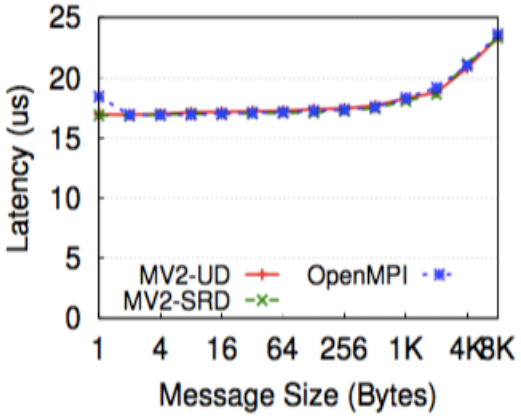
Agenda

- Introduction
- Support to AWS EFA
  - Overview of AWS EFA
  - Designing MPI Libraries for EFA
  - **Experimental Evaluations**
- Support to Azure VM
  - Dedicated Performance Evaluation & Tuning
  - One-click quick deployment

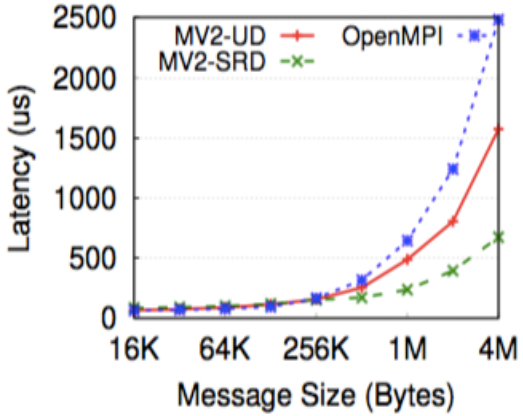
Experimental Setup

- Instance type: c5n.18xlarge
- CPU: Intel Xeon Platinum 8124M @ 3.00GHz
- Cores: 2 Sockets, 18 cores / socket
- KVM Hypervisor, 192 GB RAM, One EFA adapter / node
- MVAPICH2 version: MVAPICH2-X + SRD support
- OpenMPI version: OpenMPI-4.0.2 with libfabric 1.8

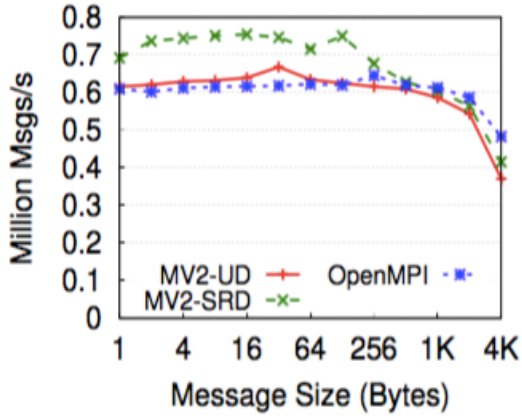
Point-to-Point Performance



(a) Small Message Latency



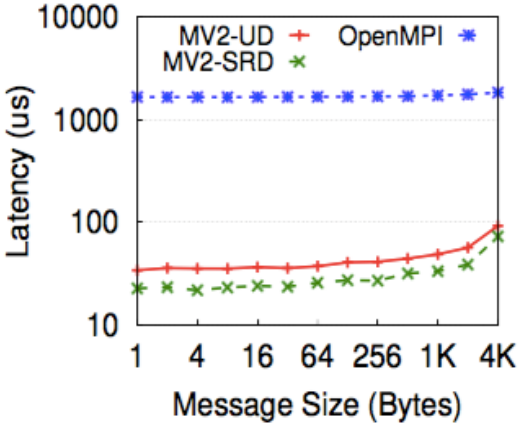
(b) Large Message Latency



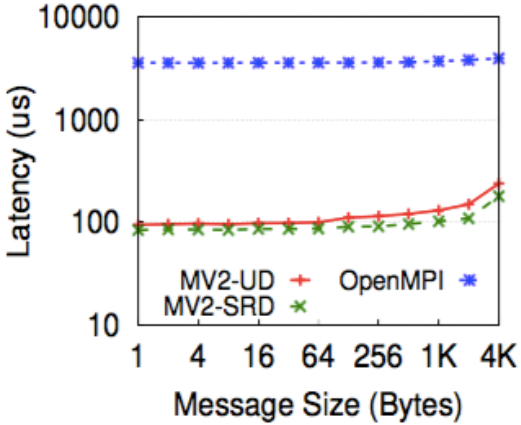
(c) Unidirectional Message Rate

- Both UD and SRD shows similar latency for small messages
- SRD shows higher message rate due to lack of software reliability overhead
- SRD is faster for large messages due to larger MTU size

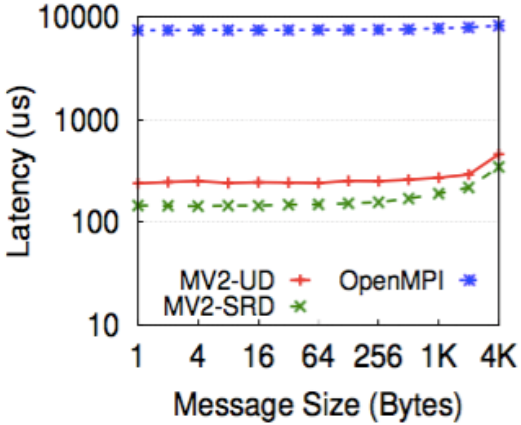
Collective Performance: MPI Scatterv



(a) 2 Nodes, 72 Processes



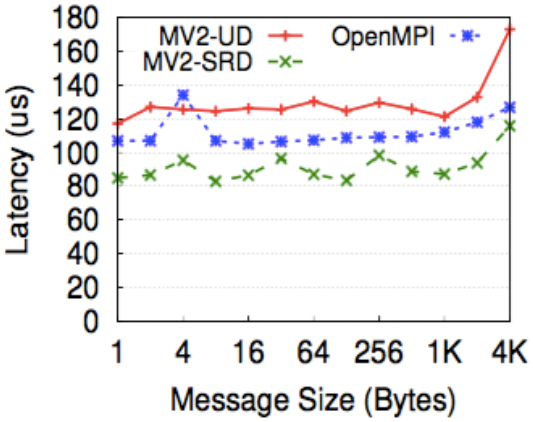
(b) 4 Nodes, 144 Processes



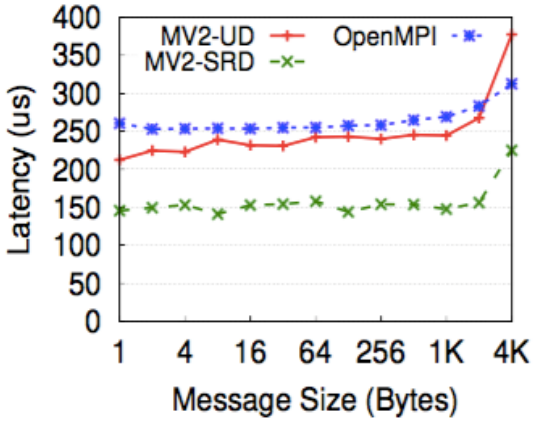
(c) 8 Nodes, 288 Processes

- SRD shows up to 60% improvement over UD
- Non-roots do not need to send back explicit acknowledgments
- Root does not need to buffer messages until ack is received

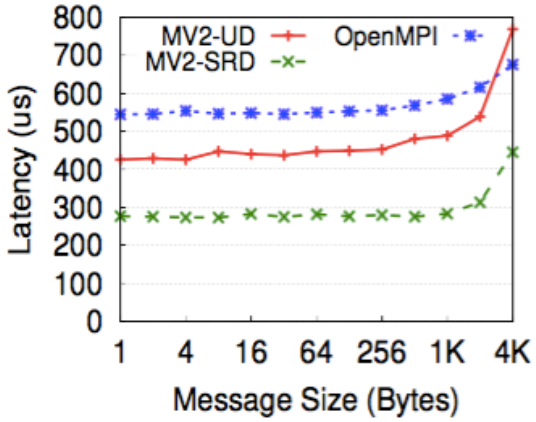
Collective Performance: MPI Gatherv



(a) 2 Nodes, 72 Processes



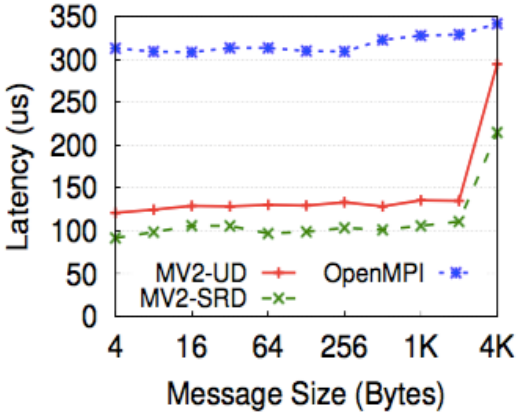
(b) 4 Nodes, 144 Processes



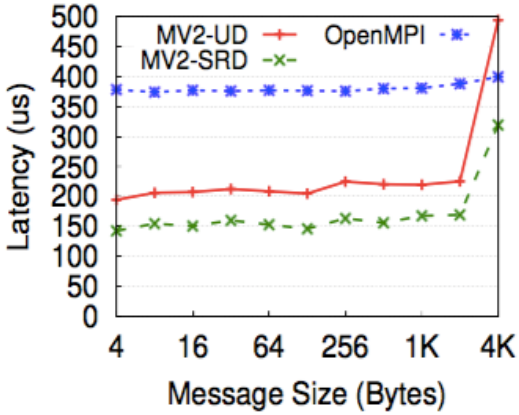
(c) 8 Nodes, 288 Processes

- Up to 33% improvement with SRD compared to UD
- Root does not need to send explicit acks to non-root processes
- Non-roots can exit as soon as the message is sent (no need to wait for acks)

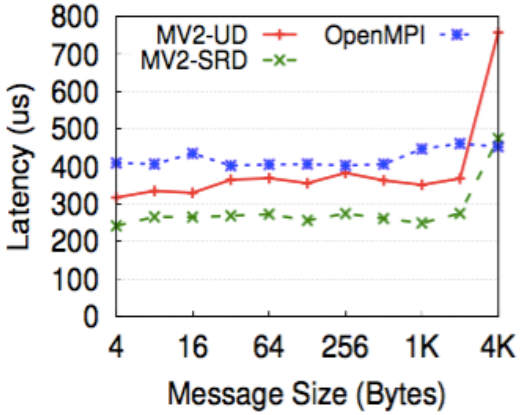
Collective Performance: MPI Allreduce



(a) 2 Nodes, 72 Processes



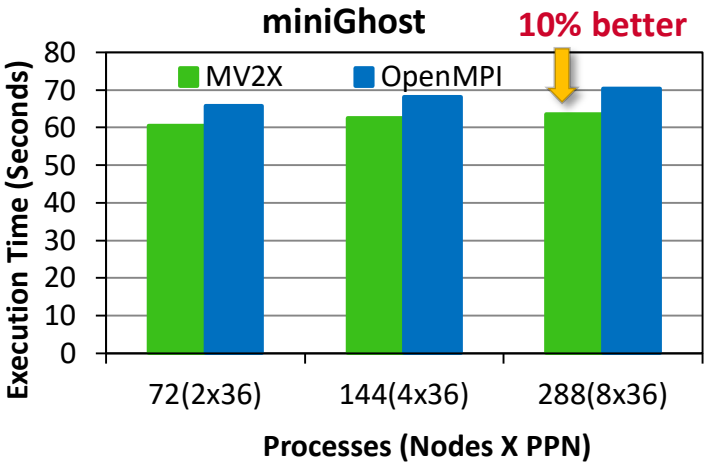
(b) 4 Nodes, 144 Processes

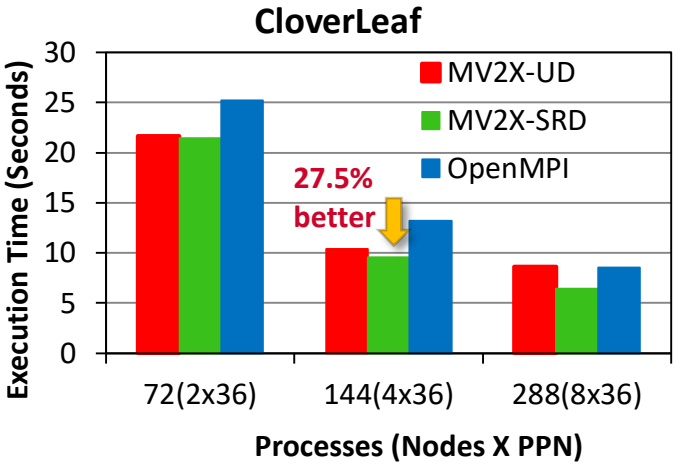


(c) 8 Nodes, 288 Processes

- Up to 18% improvement with SRD compared to UD
- Bidirectional communication pattern allows piggybacking of acks
- Modest improvement compared to asymmetric communication patterns

Application Performance





- Up to 10% performance improvement for MiniGhost on 8 nodes
- Up to 27% better performance with CloverLeaf on 8 nodes

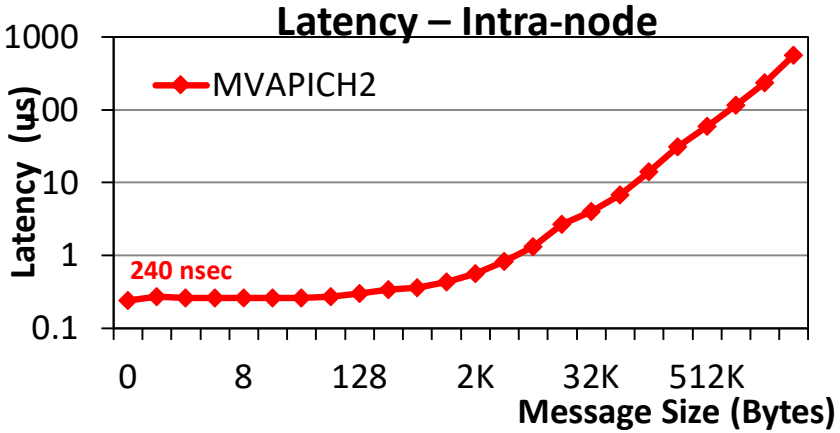
Conclusion

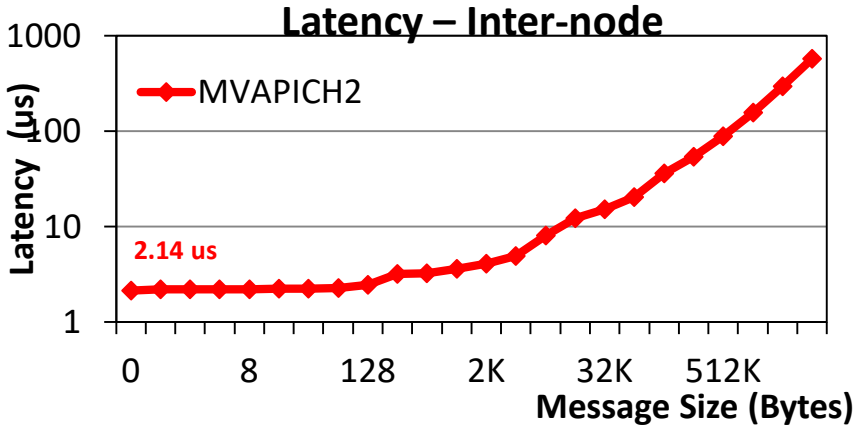
- HPC workloads are being run on cloud environments
  - Networks differ significantly from traditional on-premise HPC clusters
  - MPI libraries need to leverage the features and address the limitations
- Amazon Elastic Fabric Adapter provides lower latency
  - Introduces Scalable Reliable Datagram transport
  - Take advantage of hardware level reliable delivery in MPI
  - Proposed designs for zero-copy transmission of large messages
  - Show significant improvement in microbenchmarks and applications
- MVAPICH2-X for AWS 2.3 released on 04/12/2019
  - Includes support for SRD and XPMEM based transports
  - Available for download from http://mvapich.cse.ohio-state.edu/downloads/

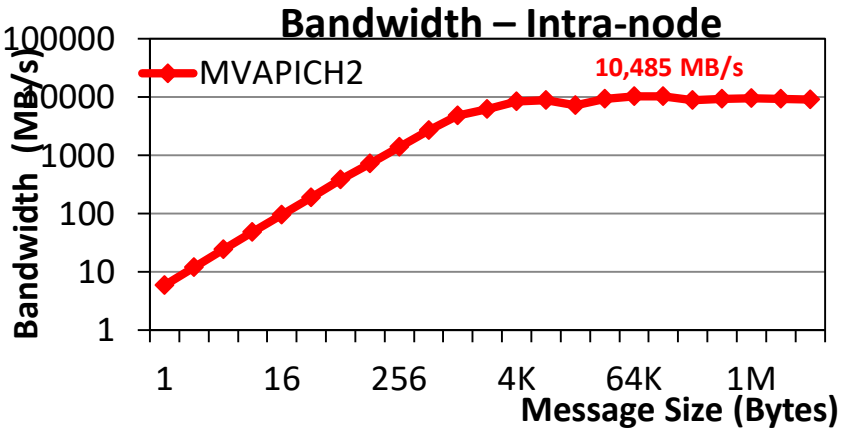
Agenda

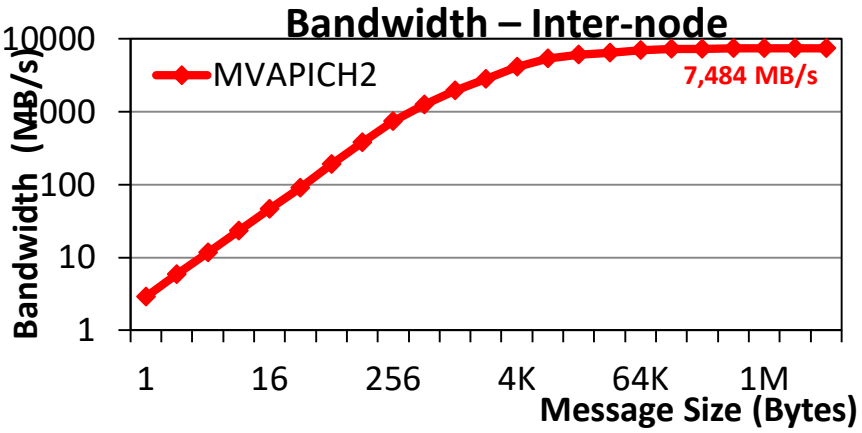
- Introduction
- Support to AWS EFA
  - Overview of AWS EFA
  - Designing MPI Libraries for EFA
  - Experimental Evaluations
- **Support to Azure VM**
  - Dedicated Performance Evaluation & Tuning
  - One-click quick deployment

MVAPICH Azure Point-to-Point Performance (HB Instances + InfiniBand EDR)

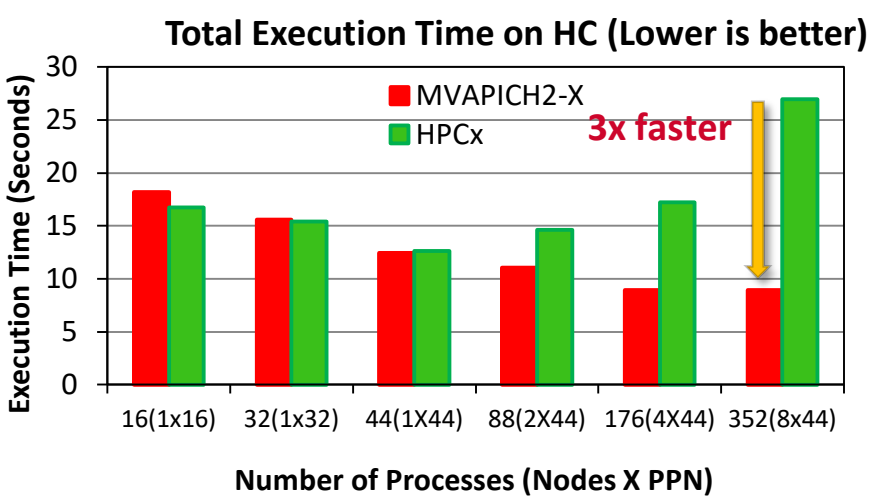


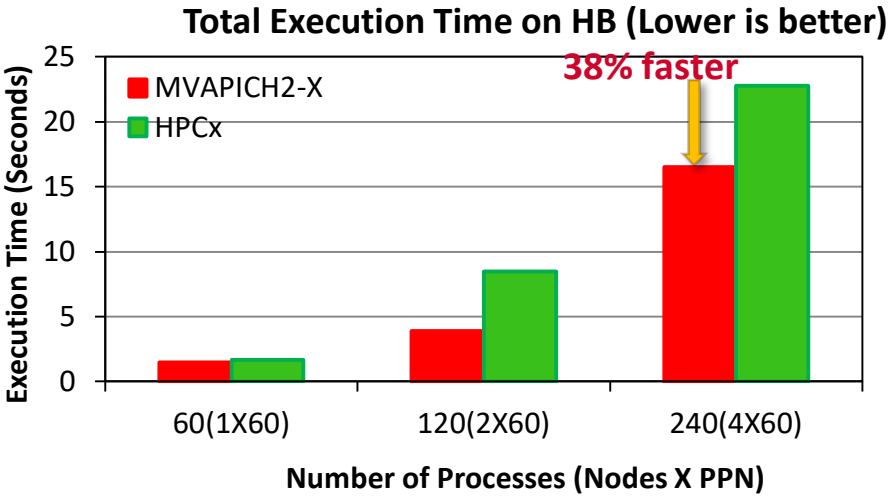






Performance of Radix

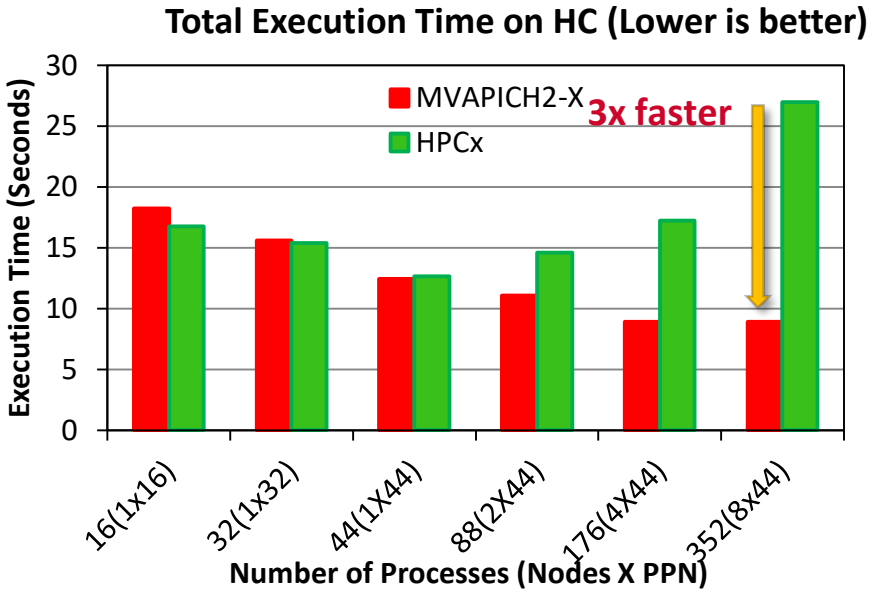


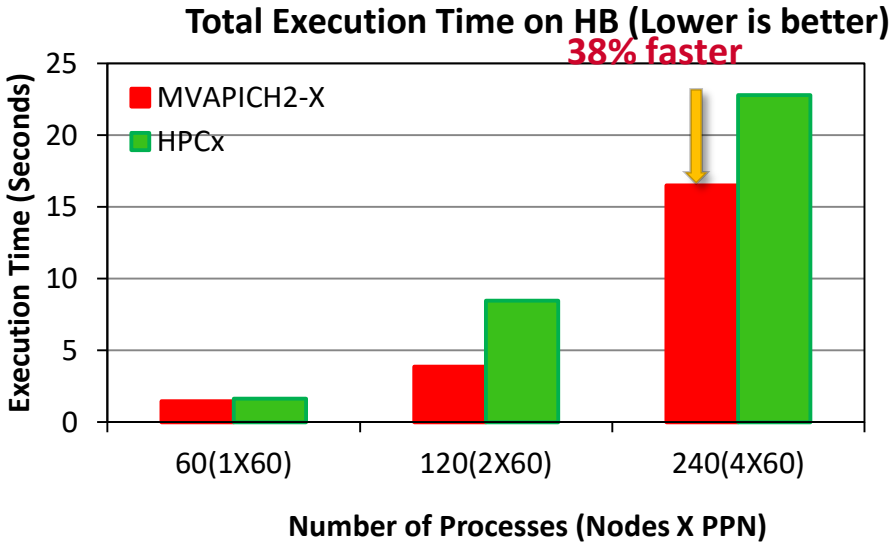


MVAPICH2-Azure 2.3.2 – One click Deployment

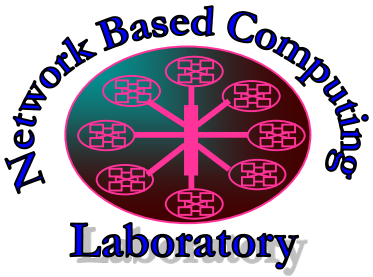
- **Released on 08/16/2019**
- Major Features and Enhancements
  - **Based on MVAPICH2-2.3.2**
  - **Enhanced tuning for point-to-point and collective operations**
  - **Targeted for Azure HB & HC virtual machine instances**
  - **Flexibility for 'one-click' deployment**
  - **Tested with Azure HB & HC VM instances**

Performance of Radix





Thank You!



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/