MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

The Ohio State University

ENGILITY
Engineered to Make a Difference

Efficient Reliability Support for Hardware Multicast-based Broadcast in GPU-enabled Streaming Applications

[1]**Ching-Hsiang Chu,** [1]Khaled Hamidouche, [1]Hari Subramoni,
[1]Akshay Venkatesh, [2]Bracy Elton and [1]Dhabaleswar K. (DK) Panda

[1]Department of Computer Science and Engineering, The Ohio State University
[2]Engility Corporation

Outline

- Introduction
- Proposed Designs
- Performance Evaluation
- Conclusion and Future Work

Drivers of Modern HPC Cluster Architectures

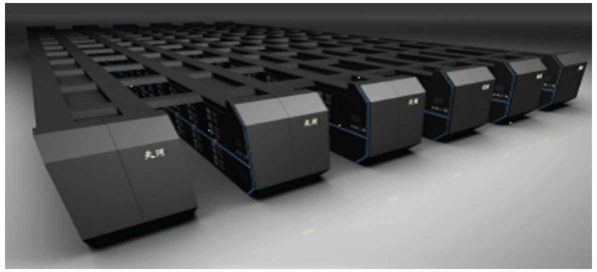
Multi-core Processors

High Performance Interconnects – InfiniBand
<1 μs latency, >100 Gbps Bandwidth

Accelerators / Coprocessors
high compute density, high performance/watt
>1 Tflop/s DP on a chip

- Multi-core processors are ubiquitous
- **InfiniBand (IB) is very popular in HPC clusters**
- **Accelerators/Coprocessors are becoming common in high-end systems**

➠ Pushing the envelope towards Exascale computing



Tianhe – 2

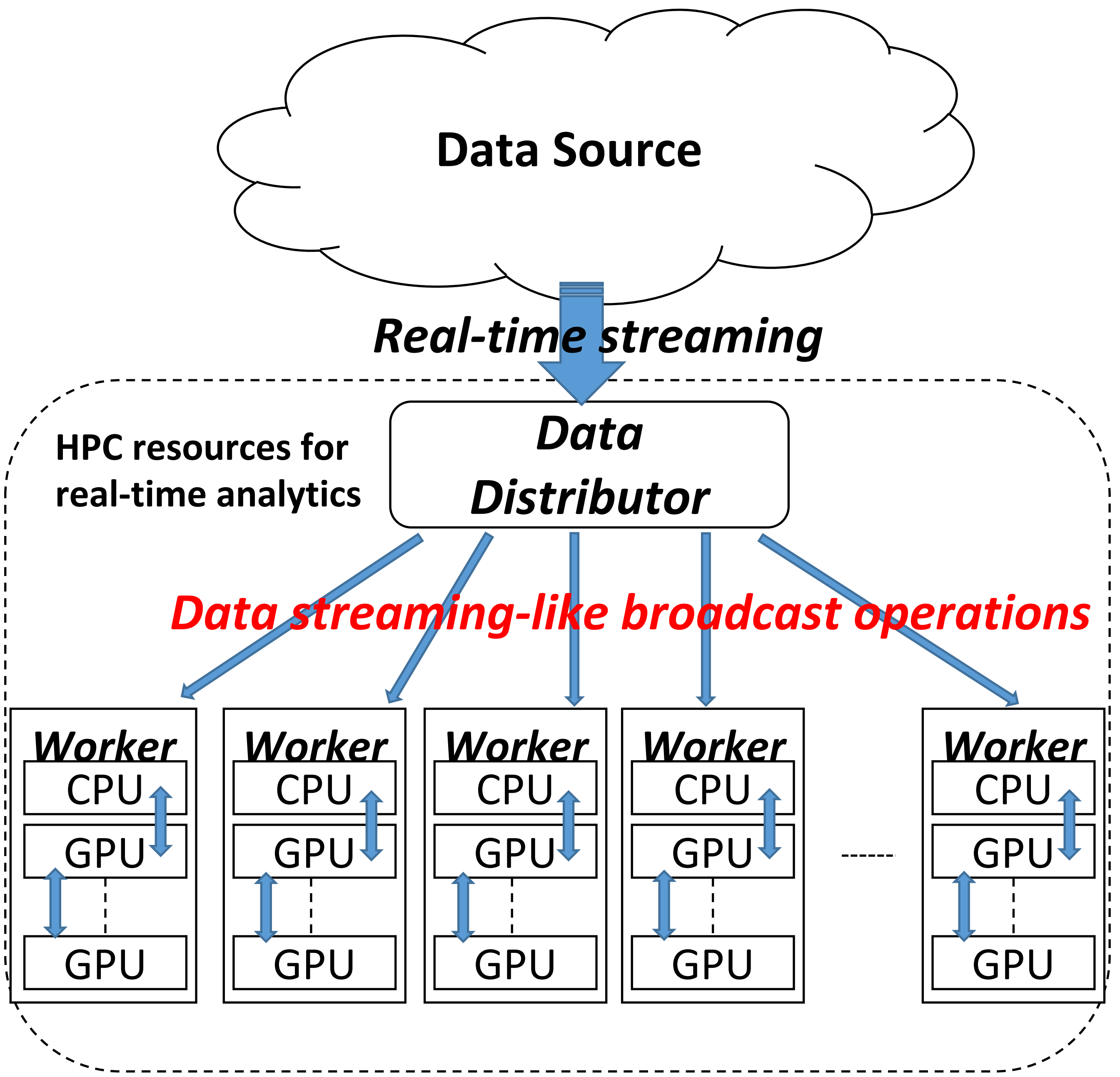
Titan

Stampede

Tianhe – 1A

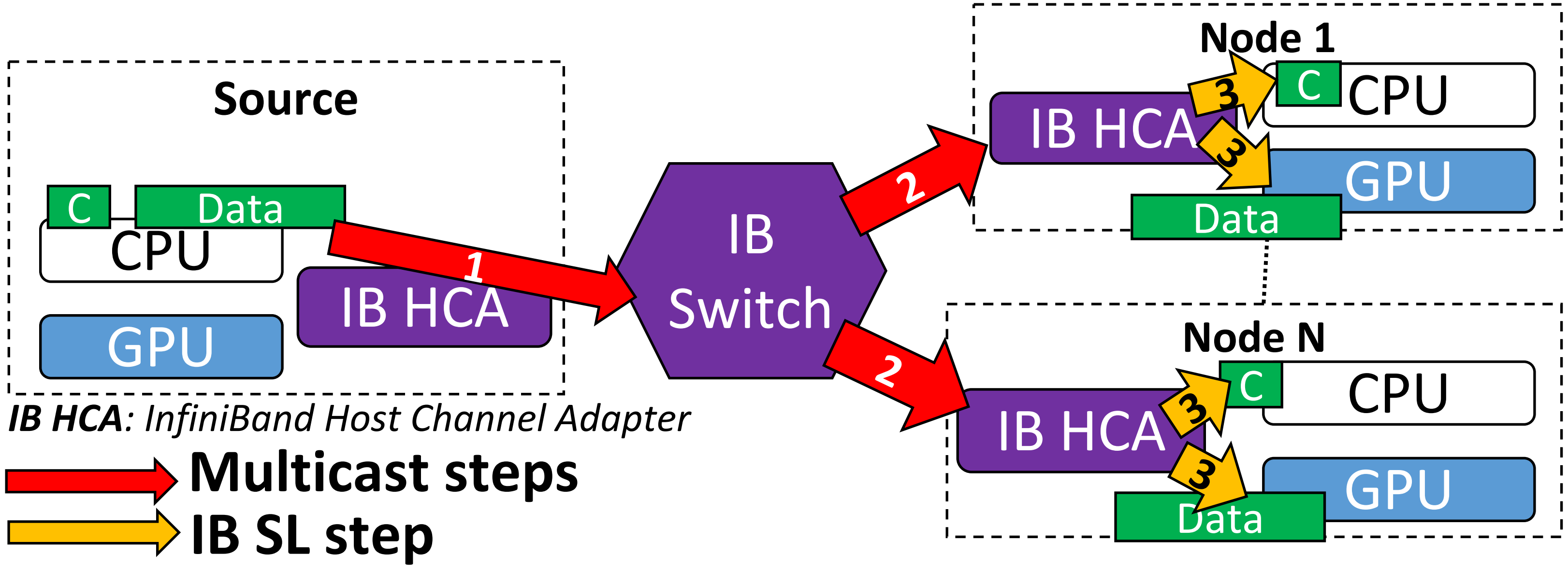
Motivation

- **Streaming applications on HPC systems**
  1. **Communication (MPI)**
     - Pipeline of broadcast-type operations
  2. **Computation (CUDA)**
     - Multiple **GPU** nodes as workers
  – Examples
     - Deep learning frameworks
     - Proton computed tomography (pCT)



Communication for Streaming Applications

- **High-performance Heterogeneous Broadcast***
  - Leverages NVIDIA GPUDirect and IB hardware multicast (MCAST) features
  - Eliminates unnecessary data staging through host memory



*Ching-Hsiang Chu, Khaled Hamidouche, Hari Subramoni, Akshay Venkatesh, Bracy Elton, and D. K. Panda. "Designing High Performance Heterogeneous Broadcast for Streaming Applications on GPU Clusters, " SBAC-PAD'16, Oct 2016.

Limitations of the Existing Scheme

- IB hardware multicast significantly improves the performance, however, it is a Unreliable Datagram (UD)-based scheme
  - ➢ **Reliability needs to be handled explicitly**
- Existing Negative ACKnowledgement (NACK)-based Design
  - Sender must stall to check receipt of NACK packets
    - ➢ **Breaks the pipeline of broadcast operations**
  - Re-send MCAST packets even if it is not necessary for some receivers
    - ➢ **Wastes network resource, degrades throughput/bandwidth**

Problem Statement

- **How to provide reliability support while leveraging UD-based IB hardware multicast to achieve high-performance broadcast for GPU-enabled streaming applications?**
  - Maintains the pipeline of broadcast operations
  - Minimizes the consumption of Peripheral Component Interconnect Express (PCIe) resources

Outline

- Introduction
- Proposed Designs
  - Remote Memory Access (RMA)-based Design
- Performance Evaluation
- Conclusion and Future Work

Overview: RMA-based Reliability Design

- **Goals of the proposed design**
  - Allows the receivers to retrieve lost MCAST packets through the RMA operations without interrupting sender
  - ➢ Maintains pipelining of broadcast operations
  - ➢ Minimizes consumption of PCIe resources
- **Major Benefit of MPI-3 Remote Memory Access (RMA)***
  - Supports one-sided communication ➔ broadcast sender won't be interrupted
- **Major Challenge**
  - How and where receivers can retrieve the correct MCAST packets through RMA operations

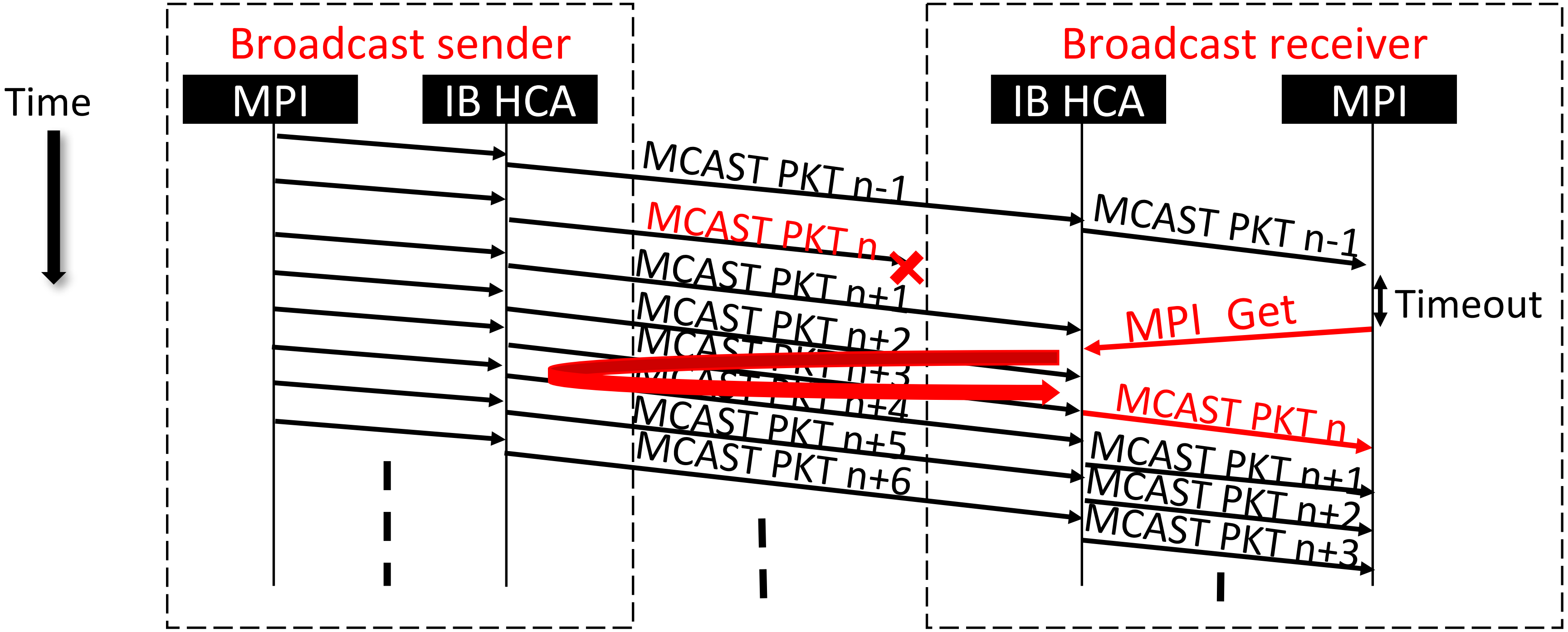
*"MPI Forum", http://mpi-forum.org/

Implementing MPI_Bcast: Sender Side

- Maintains an additional window of a **circular backup buffer** for MCAST packets
- Exposes this window to other processes in the MCAST group, e.g., performs MPI_Win_create
- Utilizes an additional **helper thread** to copy MCAST packets to the backup buffer → we can overlap with broadcast communication

Implementing MPI_Bcast: Receiver Side

- When a receiver experiences **timeout** (lost MCAST packet)
  - Performs the RMA Get operation to the sender's backup buffer to retrieve lost MCAST packets
  - **Sender is not interrupted**



Backup Buffer Requirements

- Large enough to keep the MCAST packets available when it is needed
- As small as possible to limit size of memory footprint

Bandwidth — *Constant* — *Round-Trip Time between sender and receiver*

$$W > \frac{B \times (K \times RTT)}{f}$$

Frame size: Size of a single MCAST packet

Outline

- Introduction
- Proposed Designs
- Performance Evaluation
  - Experimental Environments
  - Streaming Benchmark Level Evaluation
- Conclusion and Future Work

Experimental Environments

1. **RI2 cluster** @ The Ohio State University*
   - Mellanox EDR InfiniBand HCAs
   - 2 NVIDIA K80 GPUs per node
   - Used up to 16 GPU nodes
2. **CSCS cluster** @ Swiss National Supercomputing Centre
   http://www.cscs.ch/computers/kesch_escha/index.html
   - Mellanox FDR InfiniBand HCAs
   - Cray CS-Storm system, 8 NVIDIA K80 GPU cards per node
   - Used up to 88 NVIDIA K80 GPU cards over 11 nodes

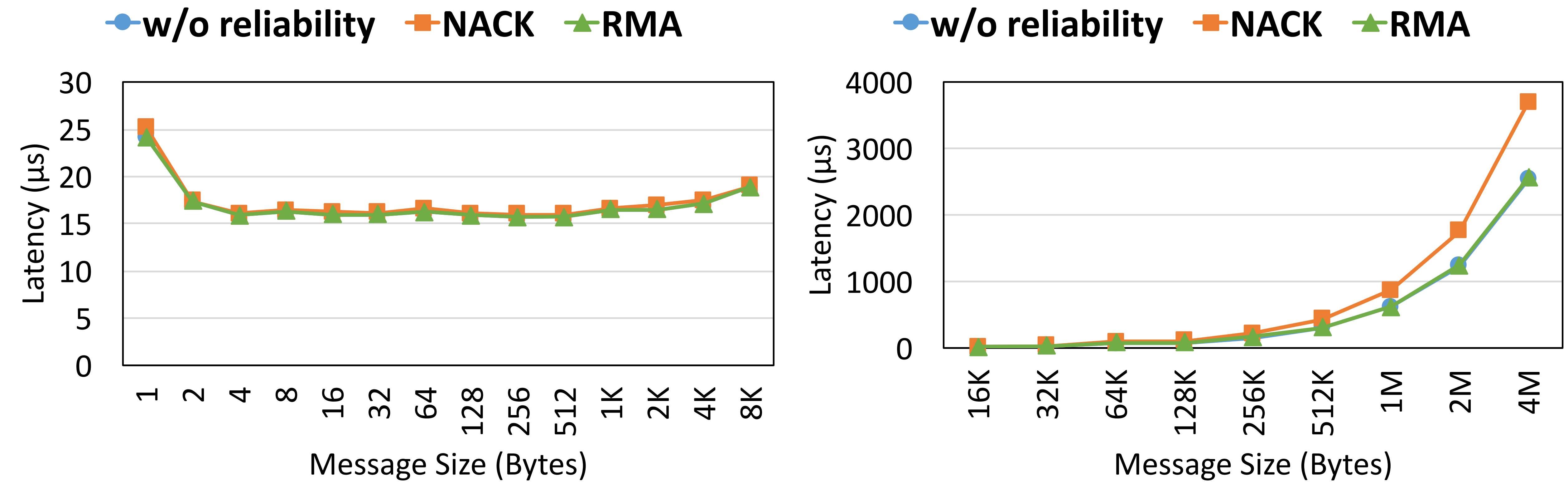
- Modified Ohio State University (OSU) Micro-Benchmark (OMB)*
  - http://mvapich.cse.ohio-state.edu/benchmarks/
  - osu_bcast - MPI_Bcast Latency Test
  - Modified to support heterogeneous broadcast
- **Streaming benchmark**
  - Mimics real streaming applications
  - Continuously broadcasts data from a source to GPU-based compute nodes
  - Includes a computation phase that involves host-to-device and device-to-host copies

**Results from RI2 and OMB are omitted in this presentation due to time constraints*

Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Enhanced Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Available since 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - **Support for GPGPUs (MVAPICH2-GDR), Available since 2014**
  - **Support for MIC (MVAPICH2-MIC), Available since 2014**
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - **Used by more than 2,675 organizations in 83 countries**
  - **More than 400,000 (> 0.4 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (June 2016 ranking)
    - 12th ranked 462,462-core cluster (Stampede) at TACC
    - 15th ranked 185,344-core cluster (Pleiades) at NASA
    - 31th ranked 74520-core cluster (Tsubame 2.5) at Tokyo Institute of Technology
  - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
  - http://mvapich.cse.ohio-state.edu
- Empowering Top500 systems for over a decade
  - System-X from Virginia Tech (3rd in Nov 2003, 2,200 processors, 12.25 Tflop/s) ⇒
  - Stampede at TACC (12th in June 2016, 462,462 cores, 5.168 Pflop/s)

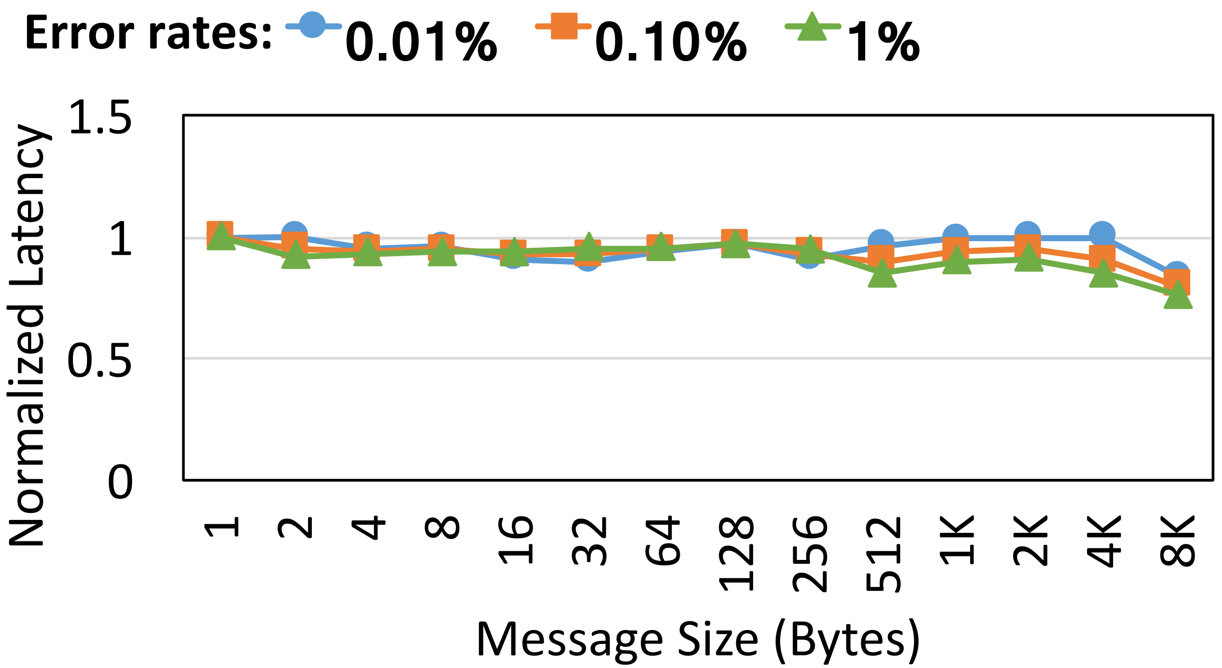
Evaluation: Overhead

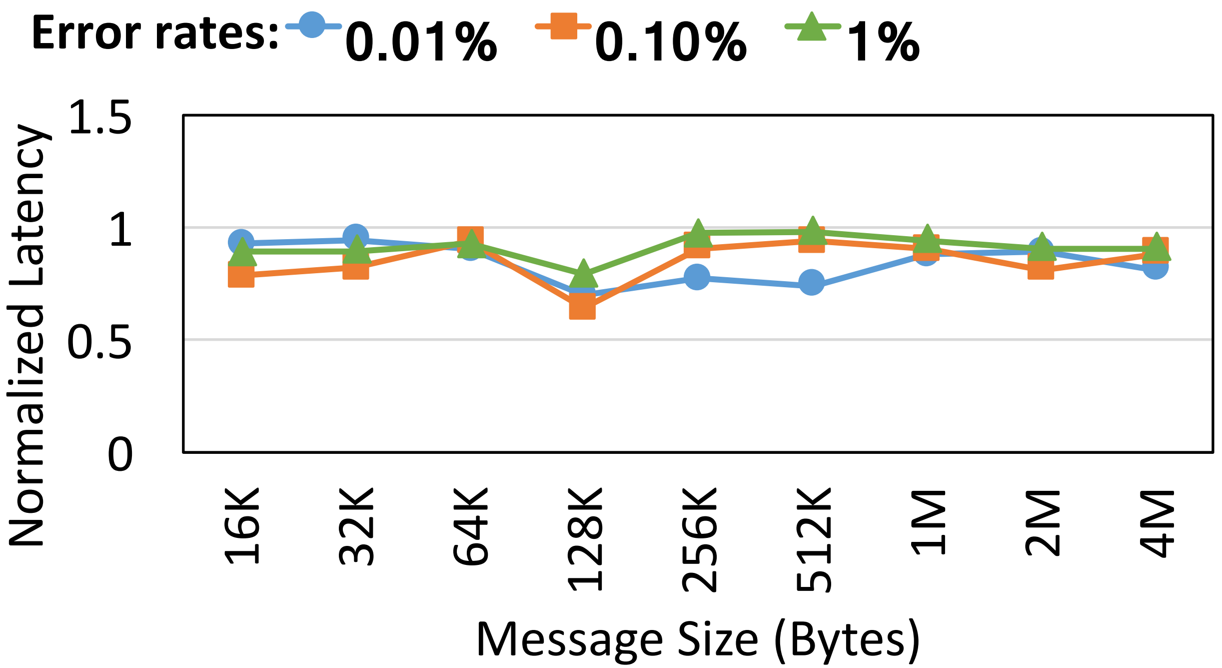


- **Negligible overhead compared to existing NACK-based design**
- **RMA-based design outperforms NACK-based scheme for large messages**
  - **A helper thread in the background performs backups of MCAST packets**

Evaluation: Latency on Streaming Benchmark

Normalized to SL-based MCAST with NACK-based retransmission scheme



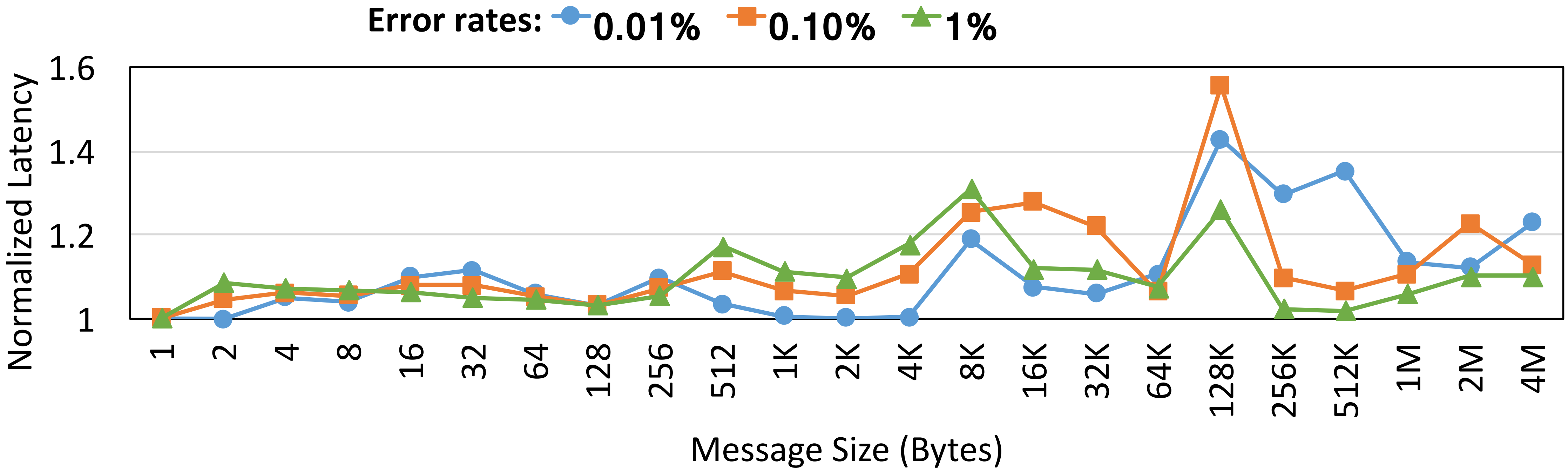


Latency reduction of proposed RMA-based design compared to the existing NACK-based scheme

		Message Size		
		8KB	128KB	2MB
Error Rate	0.01%	16%	31%	11%
	0.1%	21%	36%	19%
	1%	24%	21%	10%

Evaluation: Broadcast Rate (Throughput)

- Equal or better than the leading NACK-based design for different message sizes and error rates
- Always yields **(up to 56%) a higher broadcast rate** than the existing NACK-based design



Normalized to SL-based MCAST with NACK-based retransmission scheme

Outline

- Introduction
- Proposed Designs
- Performance Evaluation
- Conclusion and Future Work

Conclusion

- **Propose an RMA-based reliability design on top of IB hardware multicast based broadcast for streaming applications**
  - Maintains pipelining of broadcast operations
  - Minimizes consumption of PCIe resources
  - Provides good performance with streaming benchmarks, which is promising for real streaming applications
- **Future work**
  - Include the proposed design in future releases of the MVAPICH2-GDR library
  - Evaluate effectiveness with real streaming applications

Thank You!

Ching-Hsiang Chu

chu.368@osu.edu

THE OHIO STATE UNIVERSITY

The MVAPICH2 Project
http://mvapich.cse.ohio-state.edu/

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

This project is supported under the United States Department of Defense (DOD) High Performance Computing Modernization Program (HPCMP) User Productivity Enhancement and Technology Transfer (PETTT) activity (Contract No. GS04T09DBC0017 through Engility Corporation). The opinions expressed herein are those of the authors and do not necessarily reflect the views of the DOD or the employer of the author.