Designing High Performance Heterogeneous Broadcast for Streaming Applications on GPU Clusters

[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

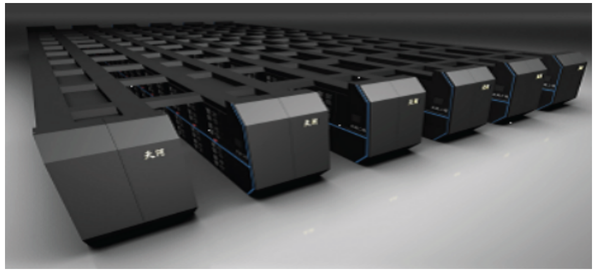
Multi-core Processors

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 is very popular in HPC clusters**
- **Accelerators/Coprocessors are becoming common in high-end systems**

➠ Pushing the envelope towards Exascale computing



Tianhe – 2

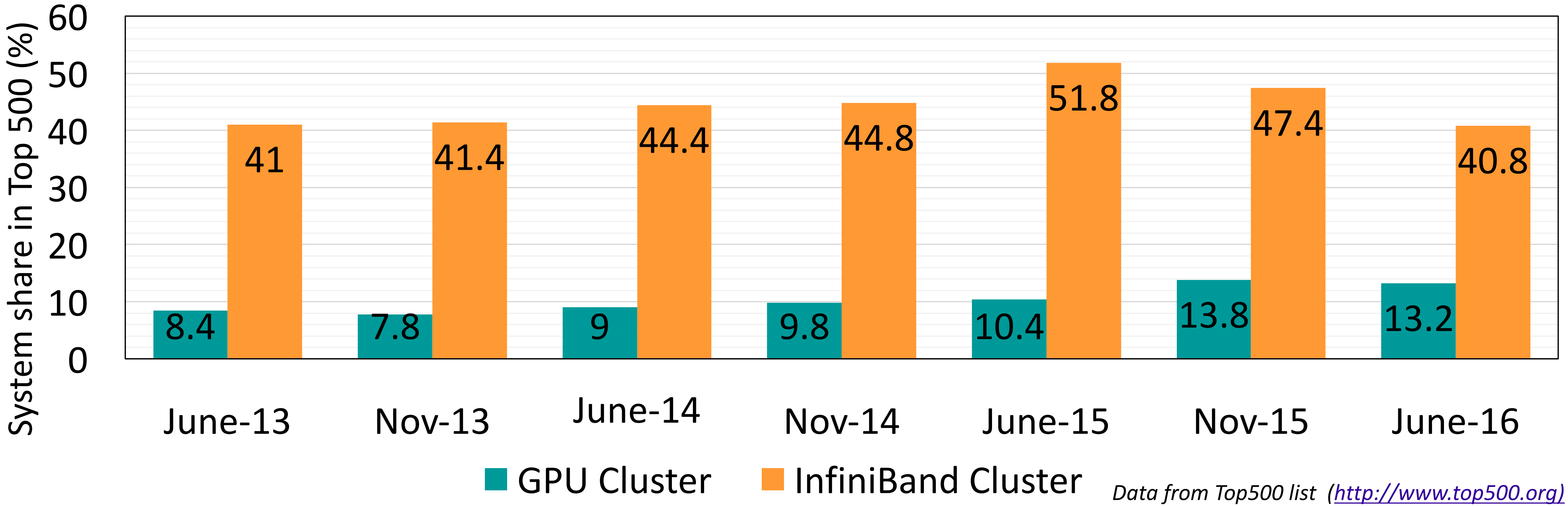
Titan

Stampede

Tianhe – 1A

IB and GPU in HPC Systems

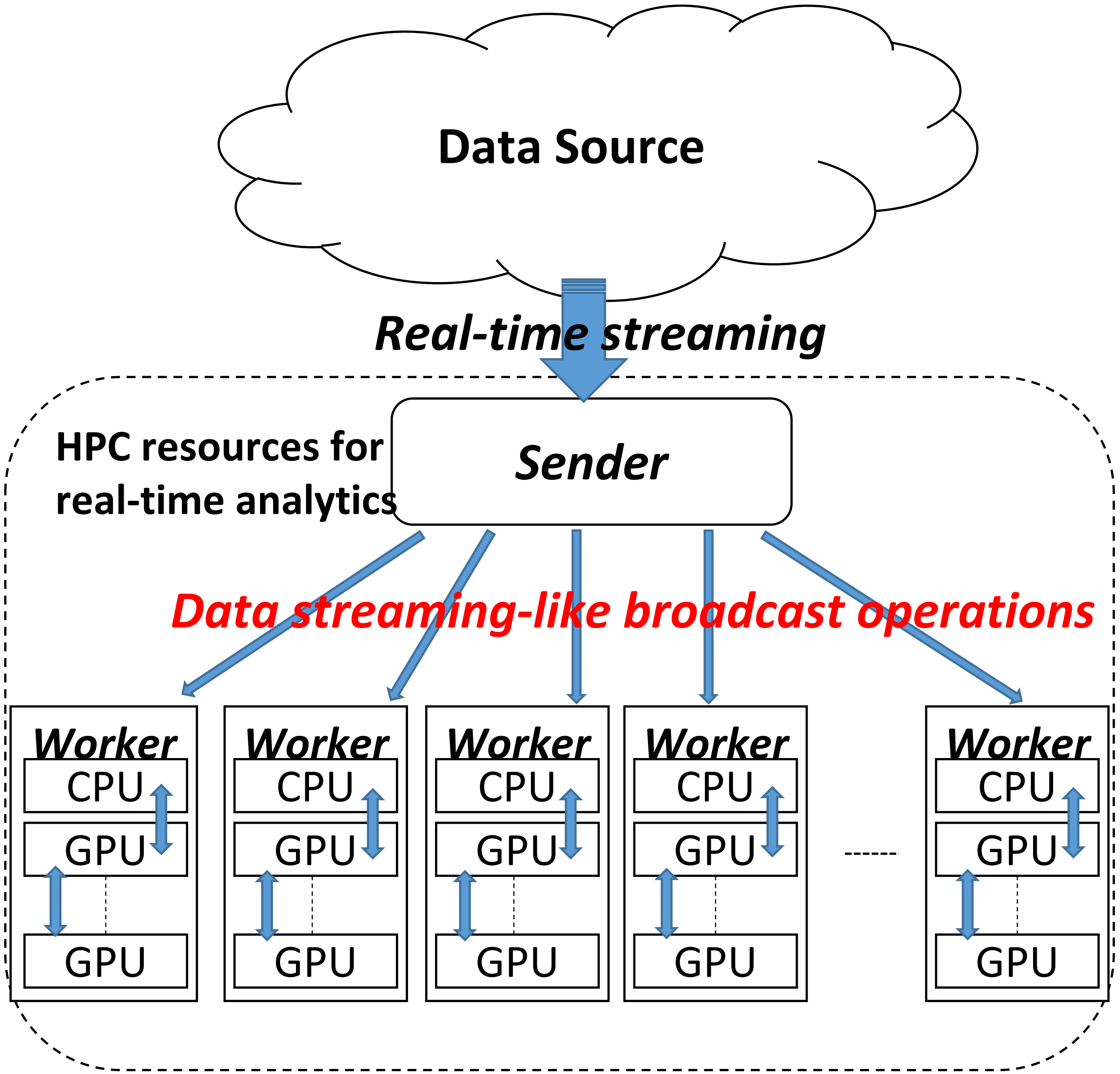
- Growth of IB and GPU clusters in the last 3 years
  - IB is the major commodity network adapter used
  - **NVIDIA GPUs boost 18% of the top 50** of the "Top 500" systems as of June 2016



Data from Top500 list (http://www.top500.org)

Motivation

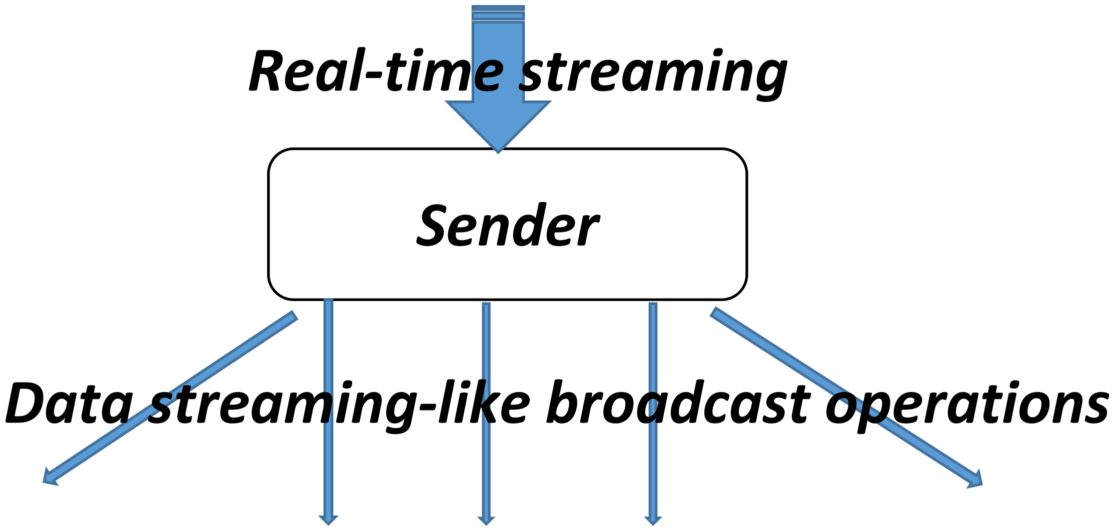
- **Streaming applications on HPC systems**
  1. **Communication (MPI)**
     - Broadcast-type operations
  2. **Computation (CUDA)**
     - Multiple GPU nodes as workers
  – Examples
     - Deep learning frameworks
     - Proton computed tomography (pCT)



Motivation

- **Streaming applications on HPC systems**

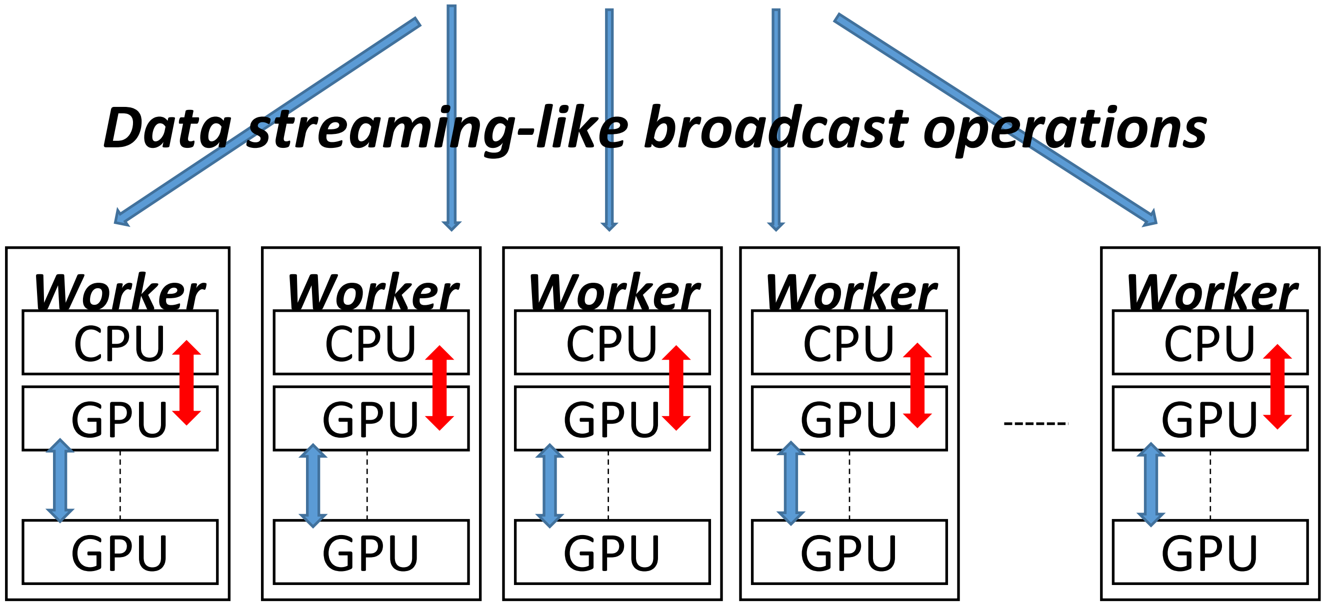
  1. **Communication — Heterogeneous Broadcast-type operations**
     - Data are usually from a **live source** and stored in Host memory
     - Data need to be sent to remote GPU memories for computing



Requires data movement from Host memory to remote GPU memories, i.e., host-device (H-D) heterogeneous broadcast
⇒ Performance bottleneck

Motivation

- **Requirements for streaming applications on HPC systems**
  - Low latency, high throughput and scalability
  - Free up Peripheral Component Interconnect Express (PCIe) bandwidth for application needs



Motivation – Technologies we have

- **NVIDIA GPUDirect[1]**
  - Use remote direct memory access (RDMA) transfers between GPUs and other PCIe devices ⇒ **GDR**
  - Peer-to-peer transfers between GPUs
  - and more…

- **InfiniBand (IB) hardware multicast (IB MCAST)[2]**
  - Enables efficient designs of homogeneous broadcast operations
    - Host-to-Host[3]
    - GPU-to-GPU[4]

[1] https://developer.nvidia.com/gpudirect
[2] Pfister GF., "An Introduction to the InfiniBand Architecture. " High Performance Mass Storage and Parallel I/O, Chapter 42, pp 617-632, Jun 2001.
[3] J. Liu, A. R. Mamidala, and D. K. Panda, "Fast and Scalable MPI-level Broadcast using InfiniBand's Hardware Multicast Support," in *IPDPS 2004*, p. 10, April 2004.
[4] A. Venkatesh, H. Subramoni, K. Hamidouche, and D. K. Panda, "A High Performance Broadcast Design with Hardware Multicast and GPUDirect RDMA for Streaming Applications on InfiniBand Clusters," in *HiPC 2014*, Dec 2014.

Problem Statement

- **Can we design a high-performance heterogeneous broadcast for streaming applications?**
  - Supports **Host-to-Device** broadcast operations
- **Can we also design an efficient broadcast for multi-GPU systems?**
- **Can we combine GPUDirect and IB technologies to**
  - Avoid extra data movements to achieve better performance
  - Increase available Host-Device (H-D) PCIe bandwidth for application use

Outline

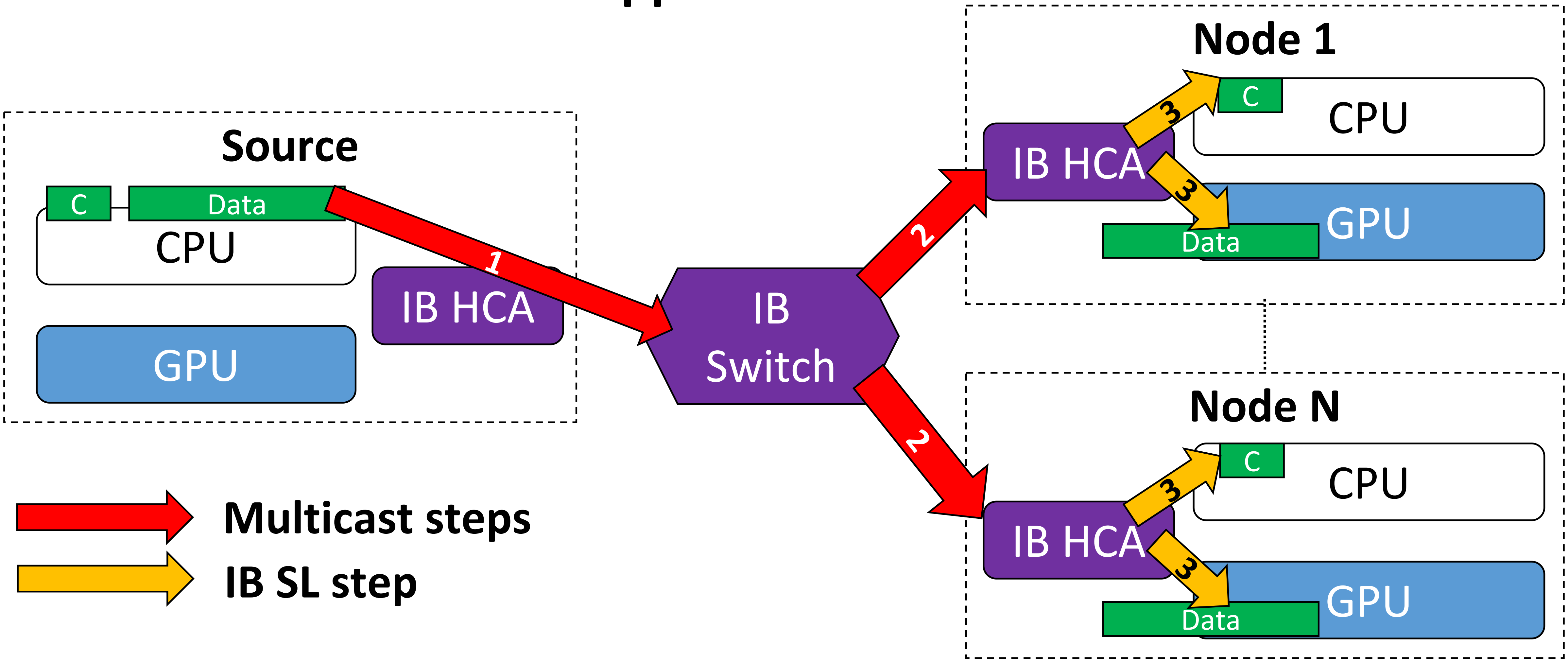
- Introduction
- Proposed Designs
  - Heterogeneous Broadcast with GPUDirect RDMA (GDR) and InfiniBand (IB) Hardware Multicast
  - Intra-node Topology-Aware Broadcast for Multi-GPU Systems
- Performance Evaluation
- Conclusion and Future Work

Proposed Heterogeneous Broadcast

- **Key requirement of IB MCAST**
  - Control header needs to be stored in host memory
- **SL-based approach: Combine CUDA GDR and IB MCAST features**
  - Also, take advantage of **IB Scatter-Gather List (SGL)** feature:
    - Multicast two separate addresses (control on the host + data on GPU)—**in but *one* IB message**
  - Directly IB read/write from/to GPU using GDR feature ⇒ **low-latency zero-copy based schemes**
  - Avoiding extra copy between Host and GPU ⇒ **frees up PCIe bandwidth resource for application needs**
  - Employing IB MCAST feature increases **scalability**

Proposed Heterogeneous Broadcast

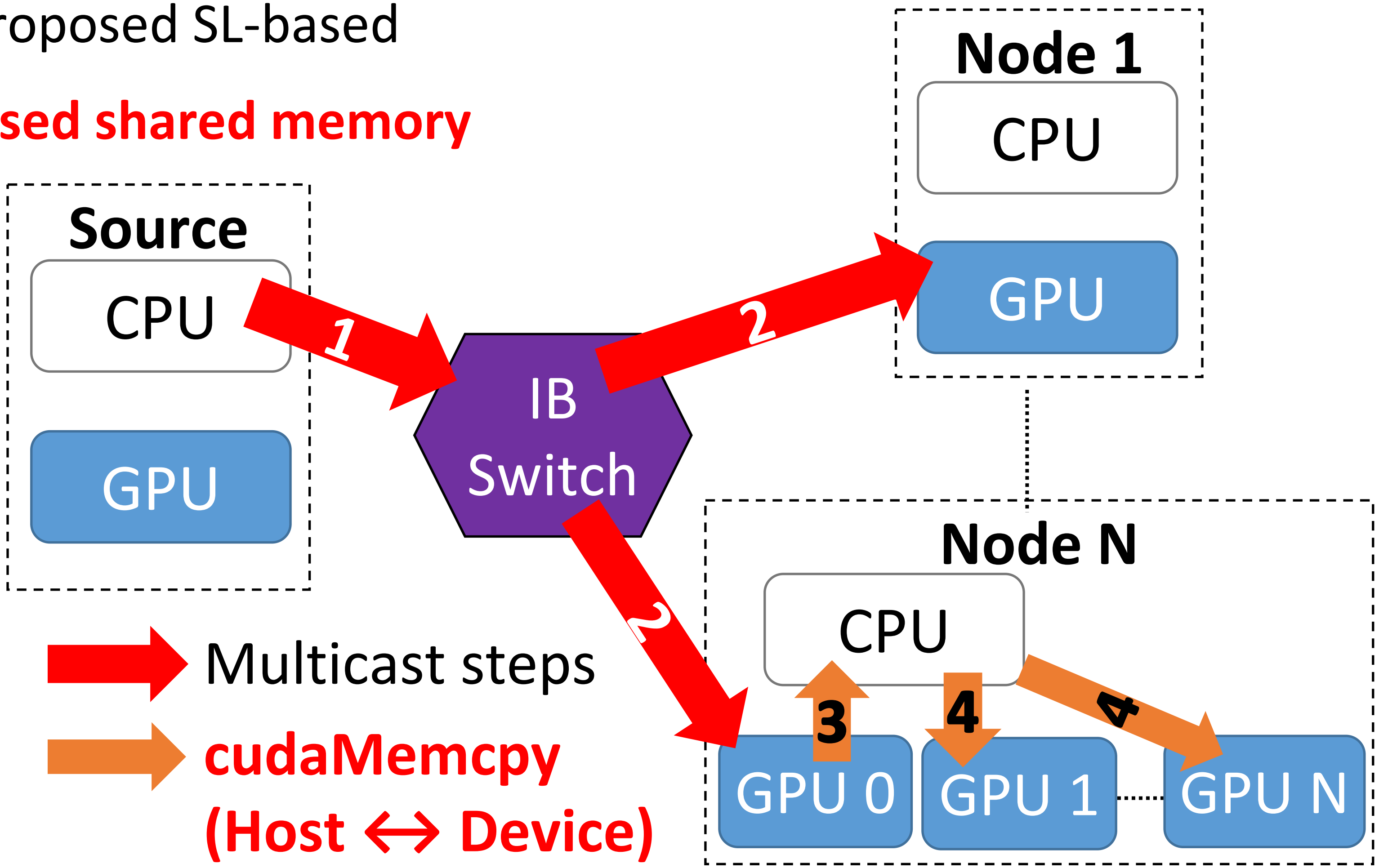
- **Overview of SL-based approach**



Broadcast on Multi-GPU systems

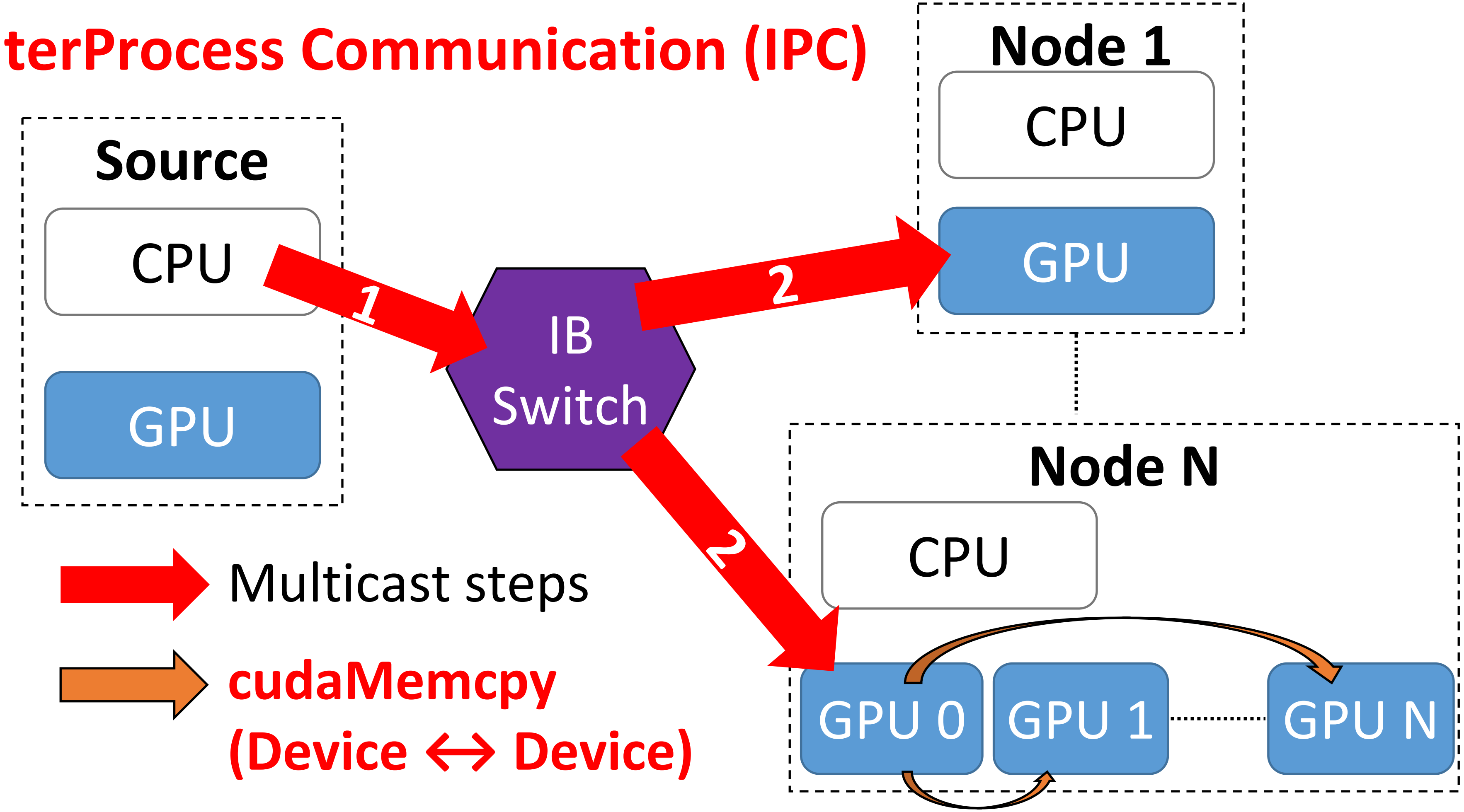
- **Existing two-level approach**
  - Inter-node: Can apply proposed SL-based
  - Intra-node: **Use host-based shared memory**

Issues of H-D cudaMemcpy :
1. Expensive
2. Consumes PCIe bandwidth between CPU and GPUs!



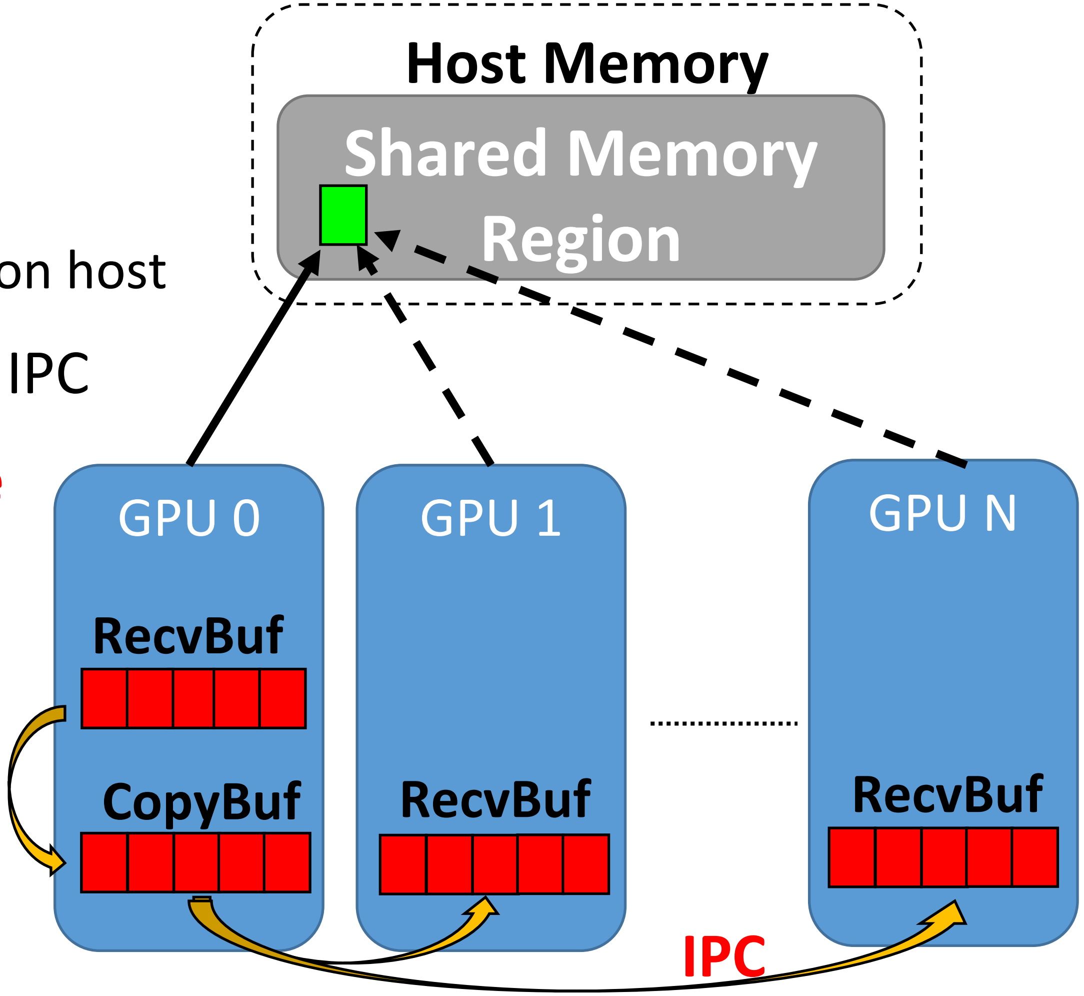
Broadcast on Multi-GPU systems

- **Proposed Intra-node Topology-Aware Broadcast**
  - **CUDA InterProcess Communication (IPC)**



Broadcast on Multi-GPU systems

- **Proposed Intra-node Topology-Aware Broadcast**
  - Leader keeps a copy of the data
  - Synchronization between GPUs
    - Use a one-byte flag in shared memory on host
  - Non-leaders copy the data using CUDA IPC
    - ➢ **Frees up PCIe bandwidth resource**
- **Other Topology-Aware designs**
  - **Ring, K-nomial…etc.**
  - **Dynamic tuning selection**



Outline

- Introduction
- Proposed Designs
- Performance Evaluation
  - OSU Micro-Benchmark (OMB) level evaluation
  - Streaming benchmark level evaluation
- Conclusion and Future Work

Experimental Environments

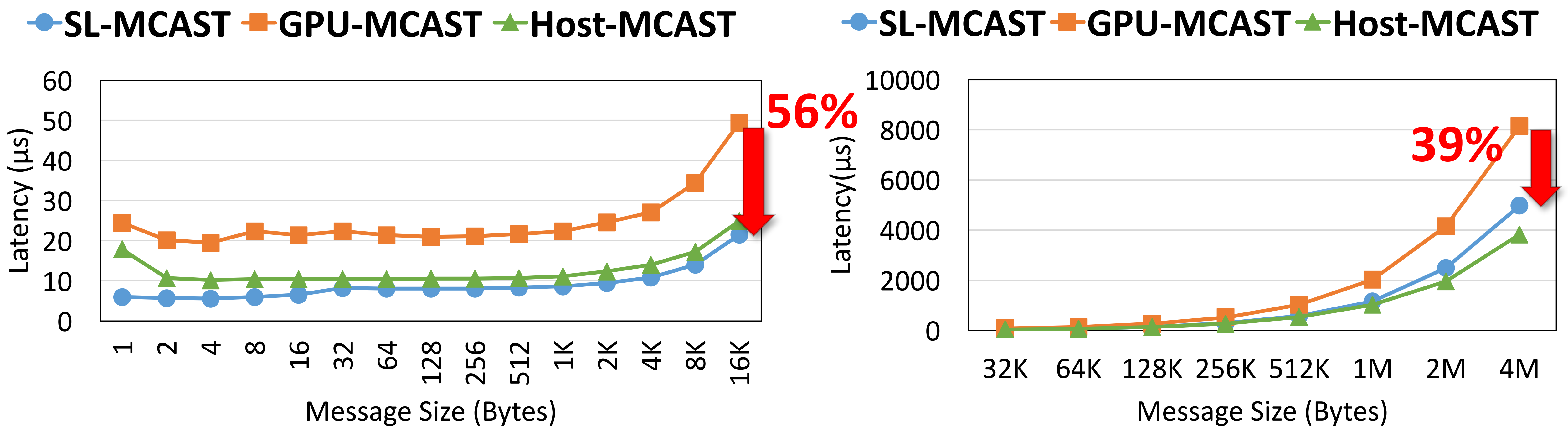
1. **Wilkes cluster** @ University of Cambridge
   http://www.hpc.cam.ac.uk/services/wilkes
   - 2 NVIDIA K20c GPUs per node
   - Used Up to 32 GPU nodes
2. **CSCS cluster** @ Swiss National Supercomputing Centre
   http://www.cscs.ch/computers/kesch_escha/index.html
   - Cray CS-Storm system
   - 8 NVIDIA K80 GPU cards per node (= 16 NVIDIA Kepler GK210 GPU chips per node)
   - Used Up to 88 NVIDIA K80 GPU cards (176 GPU chips) 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
  - Mimic 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

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 391,000 (> 0.39 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (June '16 ranking)
    - 12th ranked 462,462-core cluster (Stampede) at TACC
    - 15th ranked 185,344-core cluster (Pleiades) at NASA
    - 31st 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)

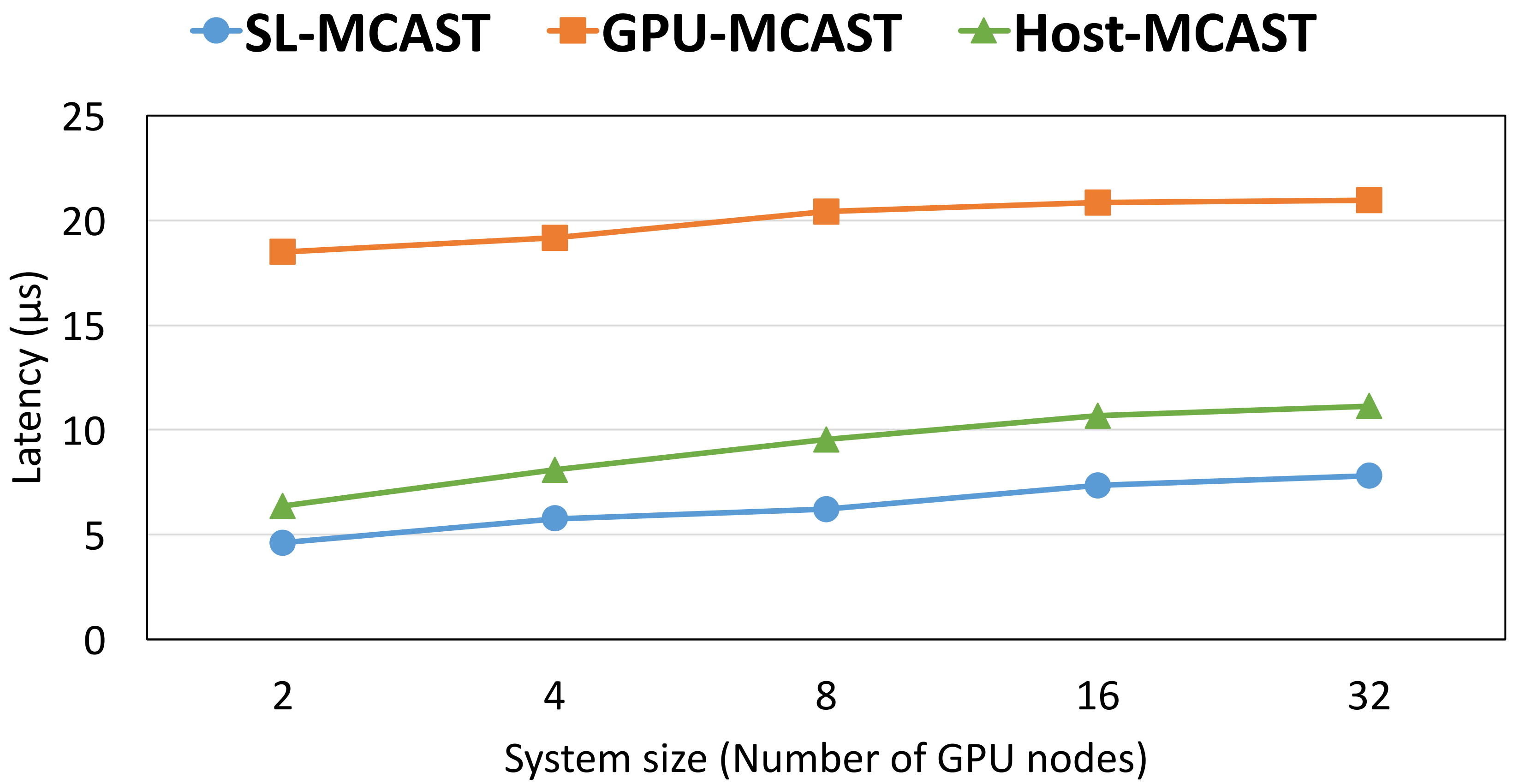
OMB – Heterogeneous Inter-node Broadcast @ Wilkes



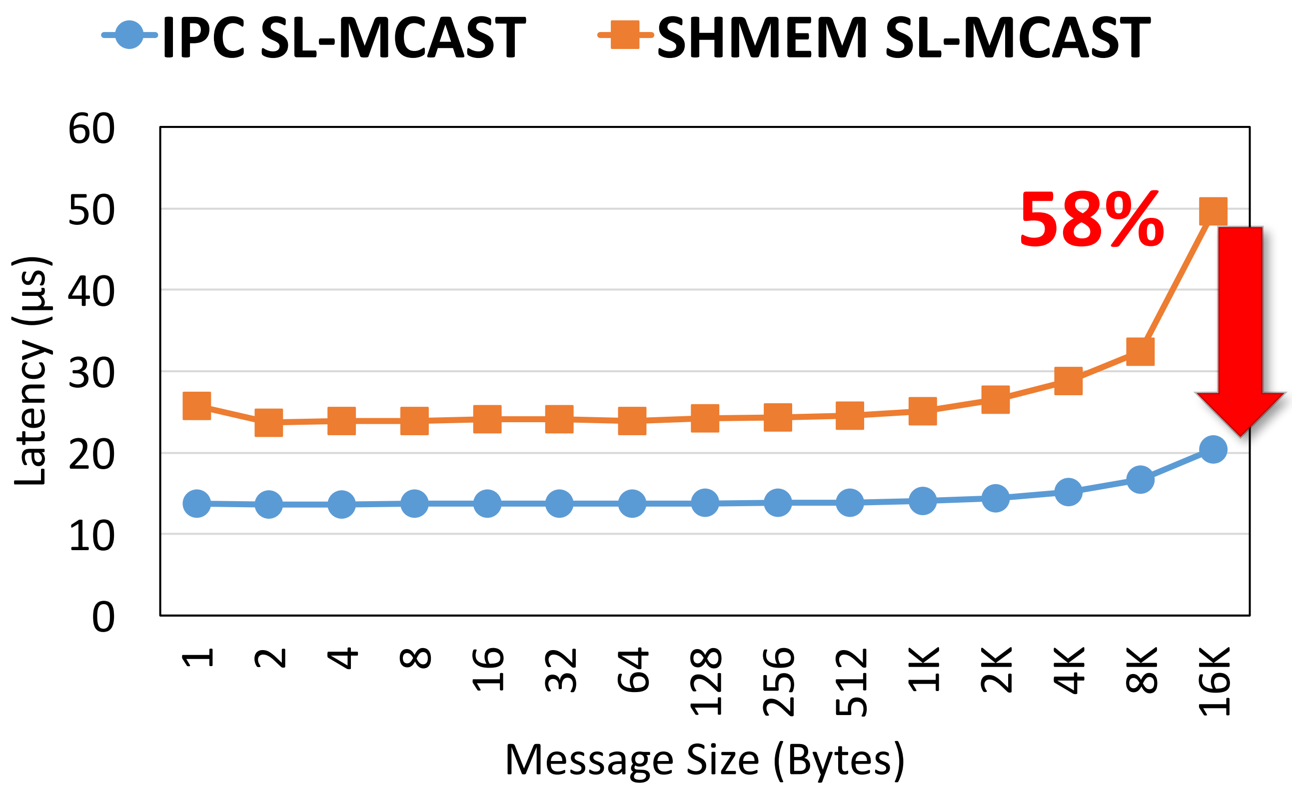
- Compared proposed SL-based design to homogeneous broadcast designs with explicitly data transfers
- Reduces latency up to **56% and 39%** for small and large messages
  - No extra data transfers between Host and GPU memories

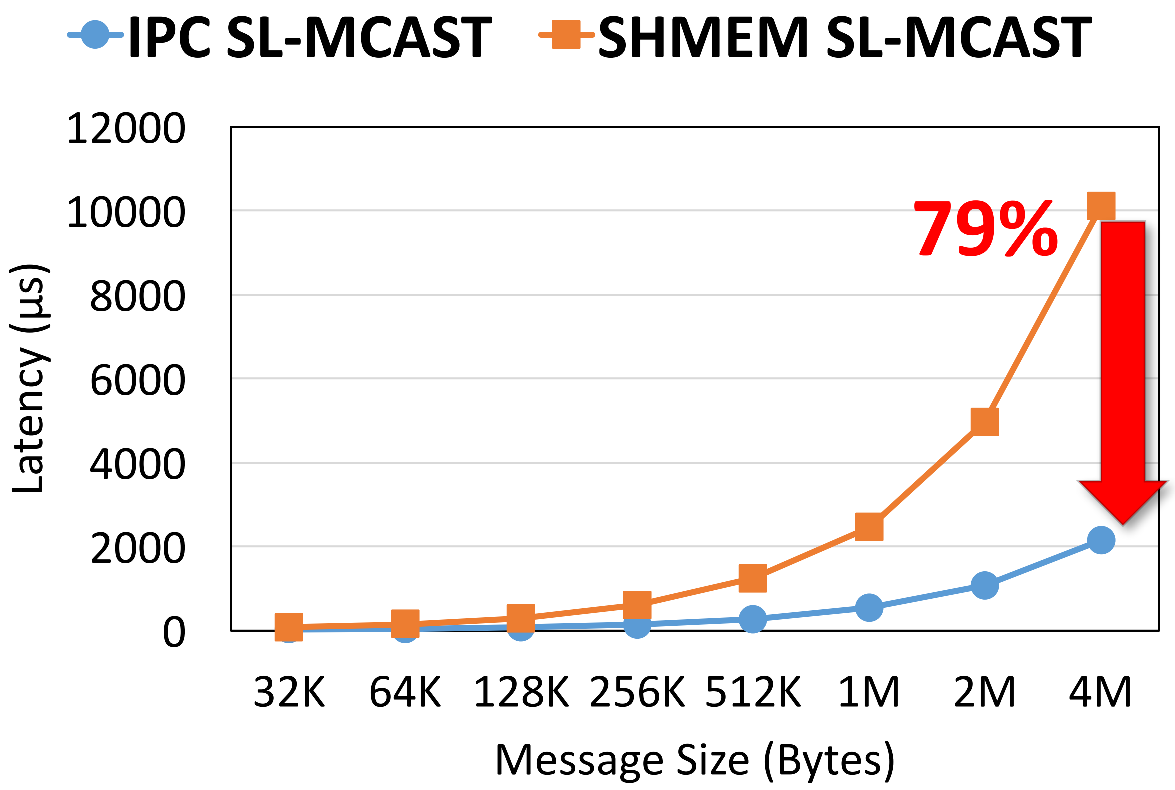
OMB – SL-based Approach

- **Inter-node Broadcast on Wilkes**
  - IB Hardware Multicast provides good scalability



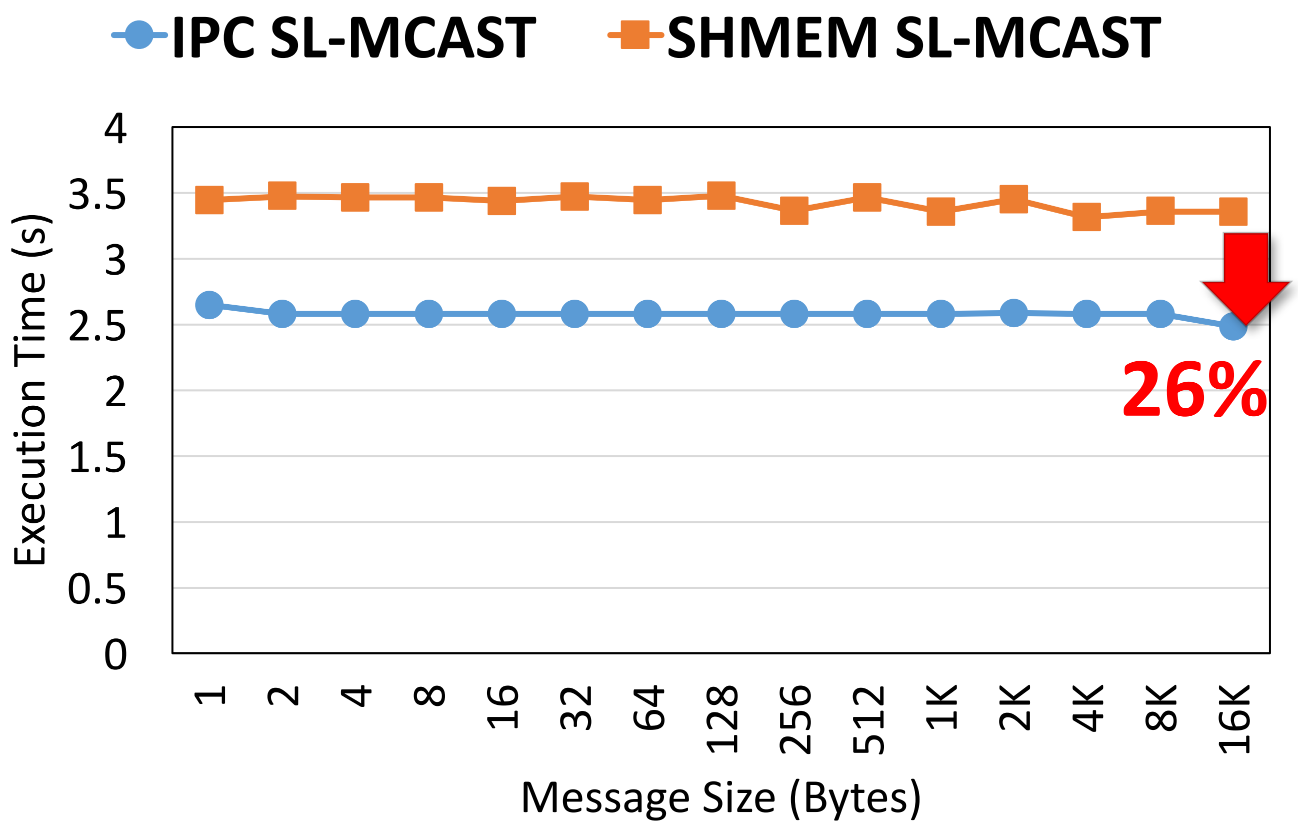
OMB – Inter- and Intra-node Broadcast @ CSCS

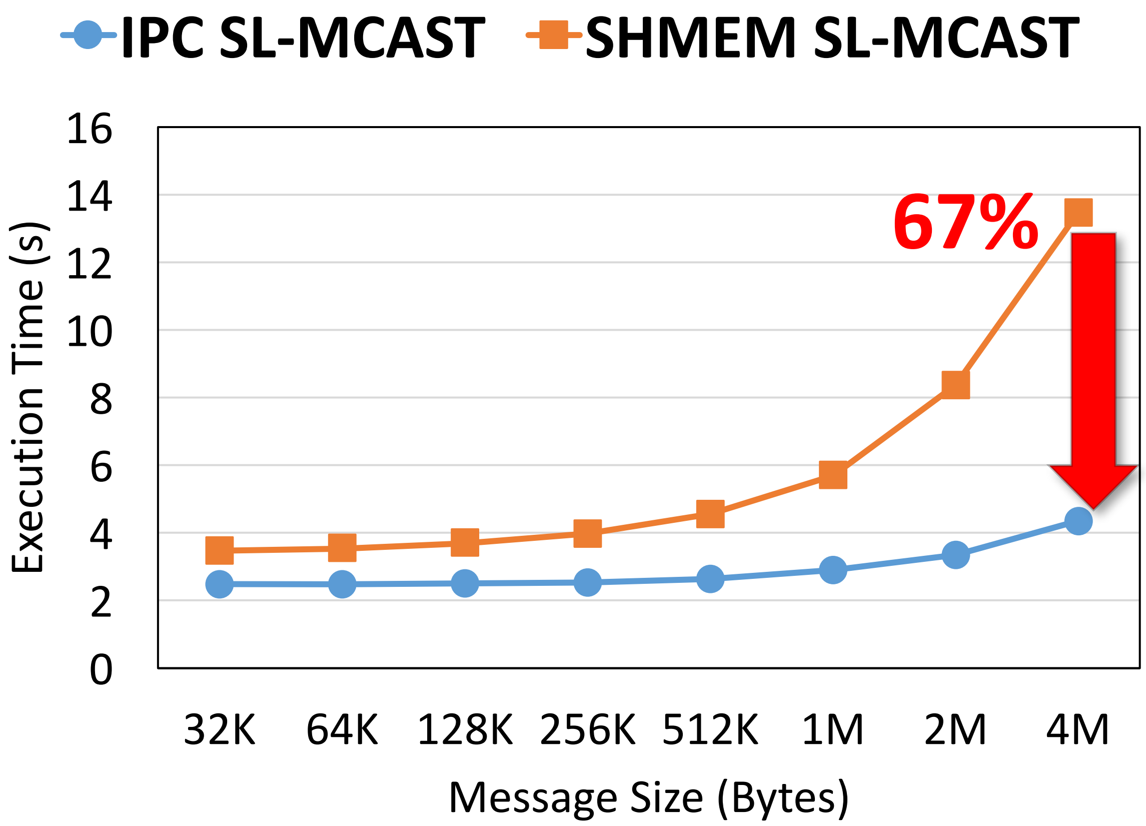




- **SL-based inter-node + Topology-aware intra-node on CSCS**
  - Up to **58% and 79% reduction** for small and large messages
    - No extra data transfers between Host and GPU memories

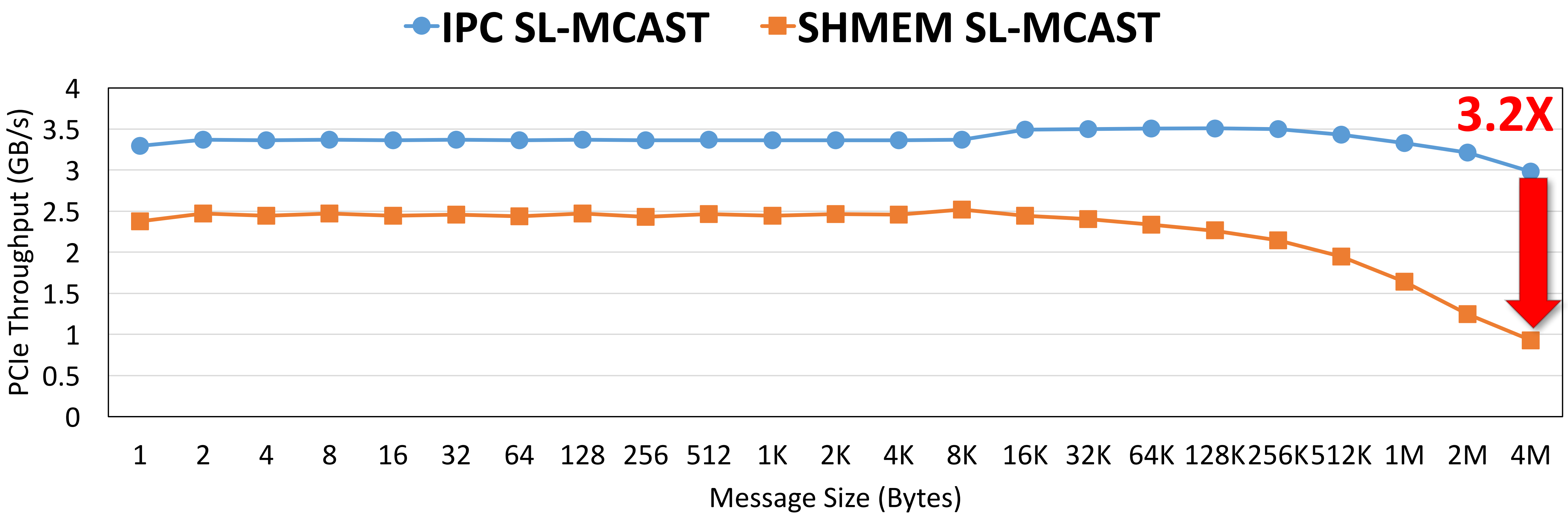
Streaming Benchmark – Execution Time @ CSCS





- Utilizes IPC-based Device-To-Device data transfer for streaming applications on multi-GPU systems
  - Up to 26% and 67% improvement for small and large messages

Streaming Benchmark – Throughput @ CSCS



- **Increases availability of PCIe Host-Device Resources**
  - Utilize IPC-based Device-to-Device data transfers
  - Free up PCIe bandwidth resources between Host and Devices for applications

Outline

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

Conclusion

- Combines **NVIDIA GPUDirect technology and InfiniBand (IB) hardware multicast** for GPU-enabled streaming applications
- Further proposes an **intra-node topology-aware** scheme that exploits **CUDA IPC** for multi-GPU systems
  - Achieves 2X improvement over state-of-the-art schemes with Ohio State University (OSU) Micro-Benchmarks (OMBs)
  - Achieves up to a 67% improvement in execution time and 3.5X of throughput in a synthetic streaming benchmark
  - Indicates applying this approach to a streaming application, such as photon computed tomography (pCT) or deep learning framework, is promising

Future Work

- Include in future releases of MVAPICH2-GDR library
- Improve reliability
- Evaluate effectiveness with streaming applications, such as, photon computed tomography (pCT) and deep learning frameworks
- Extend the designs for other collective operations as well as non-blocking operations
  - Allreduce, gather…etc.

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 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.

Streaming Benchmark

- **Mimics behavior of a streaming application**
  - 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

```
/* h_buf and d_buf: buffer on Host and GPU memory. */
for iter=0 to max_iter do
    cudaMemcpyAsync(..., cudaMemcpyHostToDevice, cpy_stream);
    if rank == root then
        MPI Bcast(h_buf, ...);
    else
        MPI Bcast(d_buf, ...);
    end if
    dummy kernel<<<...>>>(d_buf,...);
    cudaMemcpyAsync(..., cudaMemcpyDeviceToHost, cpy_stream);
end for
```