



MVA PICH

MPI, PGAS and Hybrid MPI+PGAS Library



**THE OHIO STATE
UNIVERSITY**

ENGILITY

Engineered to Make a Difference

Efficient and Scalable Multi-Source Streaming Broadcast on GPU Clusters for Deep Learning

Ching-Hsiang Chu¹, Xiaoyi Lu¹, Ammar A. Awan¹, Hari Subramoni¹,
Jahanzeb Hashmi¹, Bracy Elton² and Dhabaleswar K. (DK) Panda¹

¹Department of Computer Science and Engineering, The Ohio State University

²Engility Corporation

Outline

- **Introduction**
 - Deep Learning on GPU and InfiniBand (IB) Clusters
 - Multi-source Broadcast-type Operation for Deep Learning
- **Analysis**
- **Proposed Design**
 - Streaming-based Design with IB multicast and NVIDIA GPUDirect features
- **Performance Evaluation**
- **Conclusion and Future Work**

Trends in Modern HPC Architecture



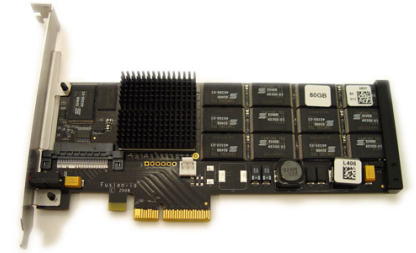
Multi-core Processors



High Performance Interconnects –
InfiniBand (**IB**), Omni-Path
< 1 μ sec latency, 100 Gbps Bandwidth>



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



SSD, NVMe-SSD, NVRAM

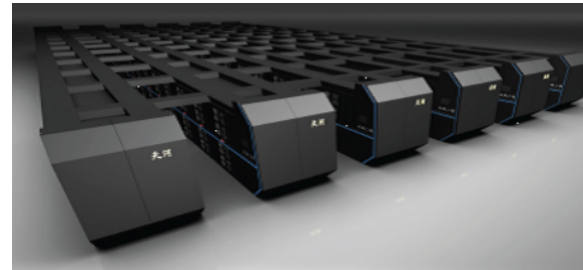
- Multi-core/many-core technologies
- High Performance Interconnects
- Accelerators/Coprocessors are becoming common in high-end systems
- High Performance Storage and Compute devices



Sunway TaihuLight



K - Computer



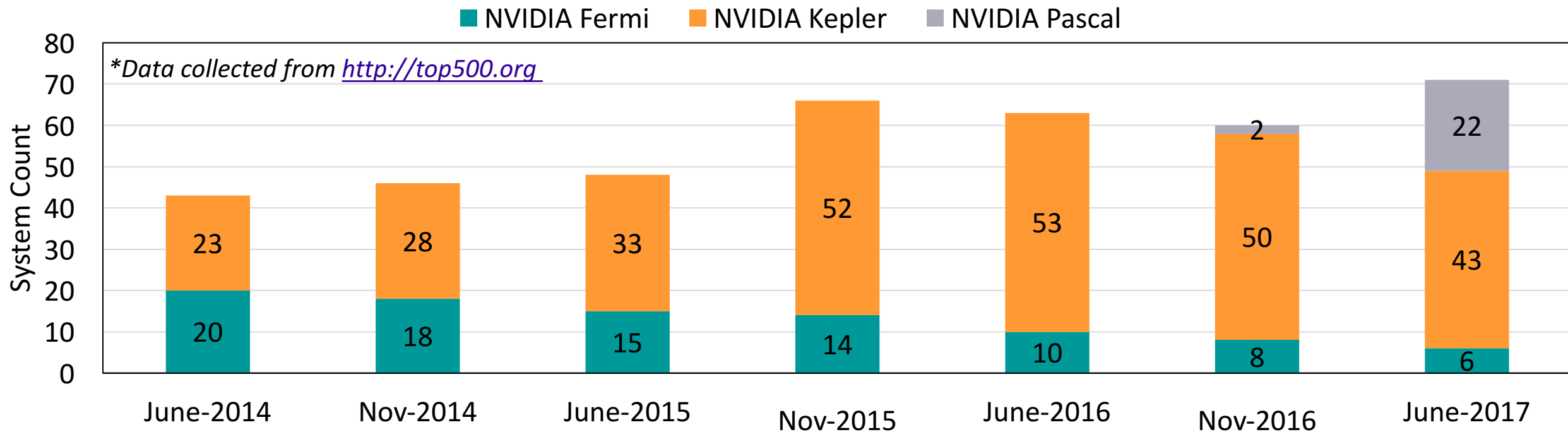
Tianhe – 2



Titan

GPU in HPC Systems

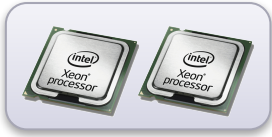
- Growth of GPU clusters in the last 3 years
 - NVIDIA GPUs boost many Top 500 and Green 500 systems
 - “Top 13 systems on the latest **Green500** are all equipped with the P100 hardware”*



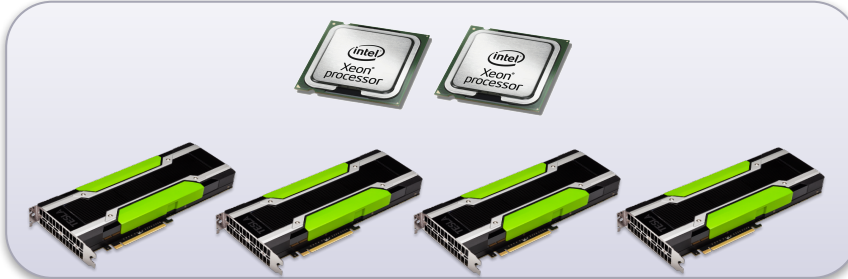
Architectures for Deep Learning (DL)

Past and Current Trend

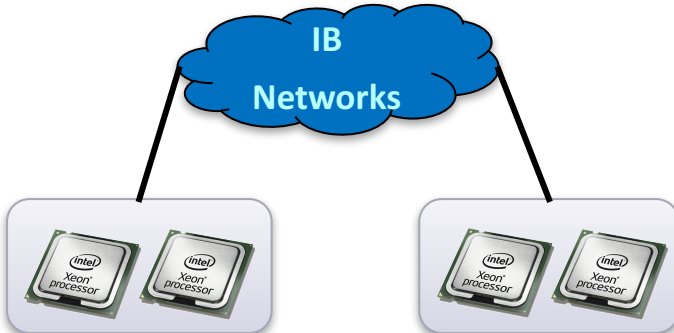
Multi-core CPUs within a node



Multi-core CPUs + Multi-GPU within a node



Multi-core CPUs across nodes

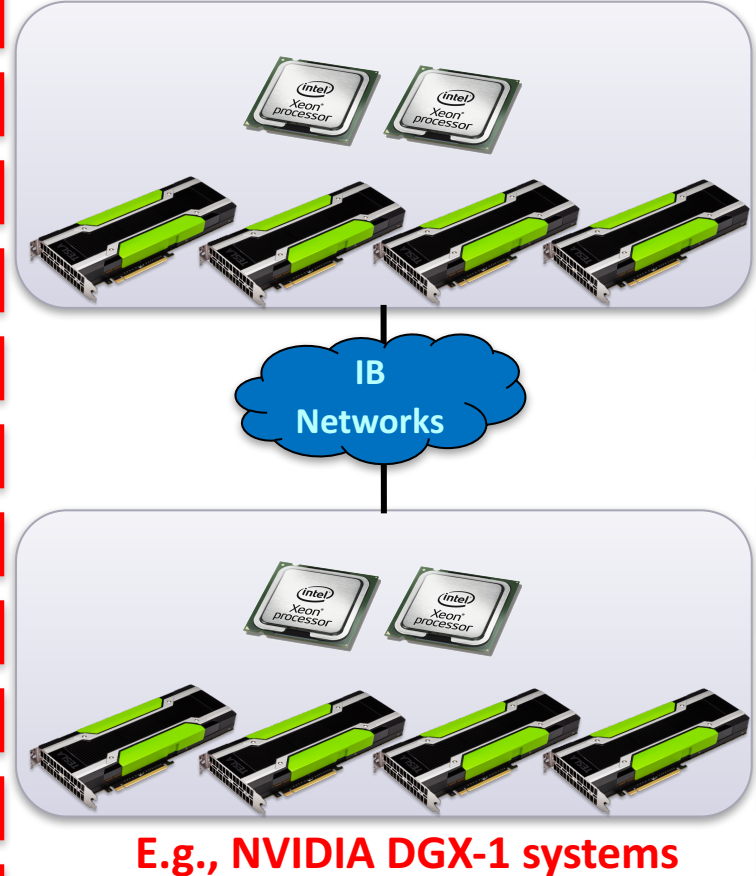


Multi-core CPUs + Single GPU across nodes



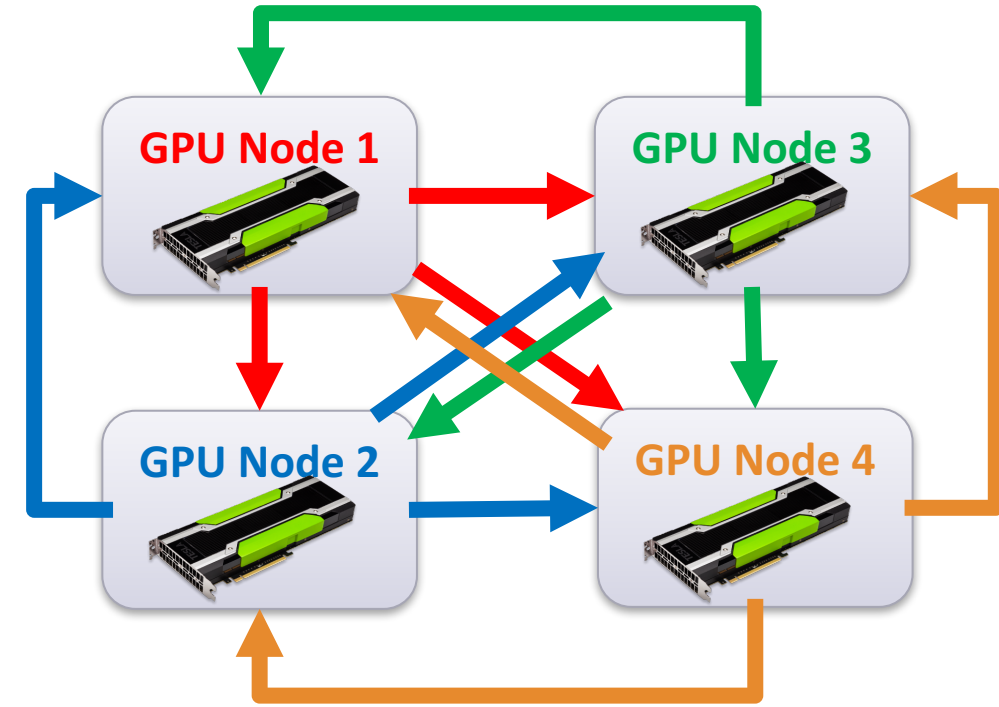
Near-future

Multi-core CPUs + Multi-GPU across nodes



High-performance Deep Learning

- Computation using **GPU**
- Communication using **MPI**
 - Exchanging partial gradients after each minibatch
 - **All-to-all (Multi-Source) communications**
 - E.g., **MPI_Bcast**
- Challenges
 - High computation-communication **overlap**
 - Good **scalability** for upcoming large-scale GPU clusters
 - No application-level modification

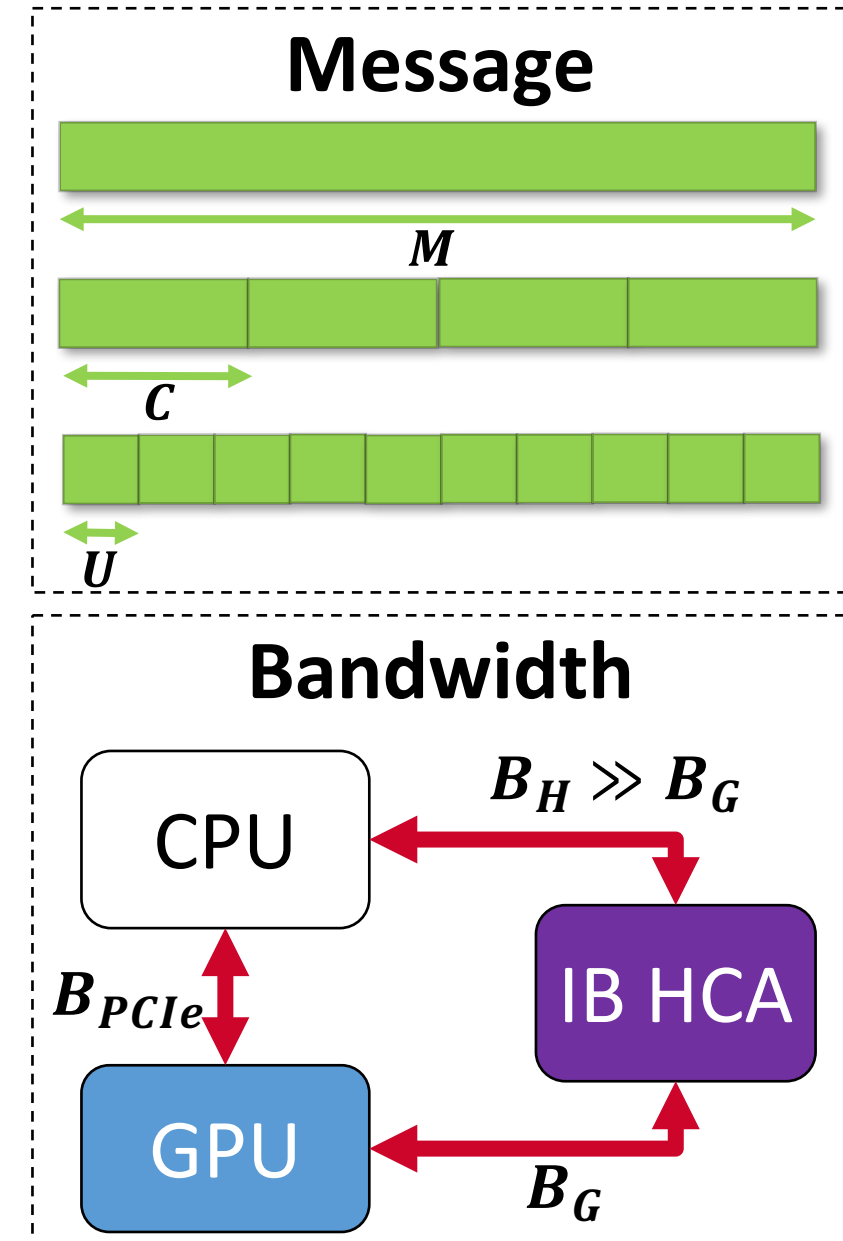


Outline

- Introduction
- Analysis
 - Existing Designs
 - Problem Statement
- Proposed Design
- Performance Evaluation
- Conclusion and Future Work

Evaluation Parameters

Notation	Meaning	Unit
n	Number of processes	N/A
m	Number of broadcast sources	N/A
t_s	Set up time for sending data	sec
$t_o(n)$	Overhead for issuing an IB-MCAST packet	sec
M	Original message size	bytes
C	Size of a data chunk	bytes
U	Maximum Transmission Unit for IB-MCAST, provided by hardware manufacturer	bytes
B_H	Bandwidth of reading Host memory	bytes/sec
B_G	Bandwidth of reading GPU memory (NVIDIA GPUDirect RDMA)	bytes/sec
B_{PCIe}	PCIe Bandwidth between Host and GPU memory	bytes/sec



Ring-based Broadcast

- Direct

$$(n-1) \times \left(t_s + \frac{M}{B_G} \right)$$

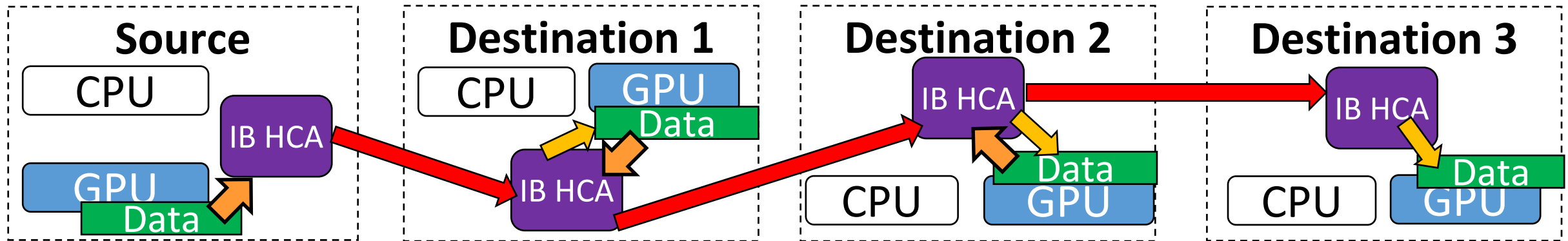
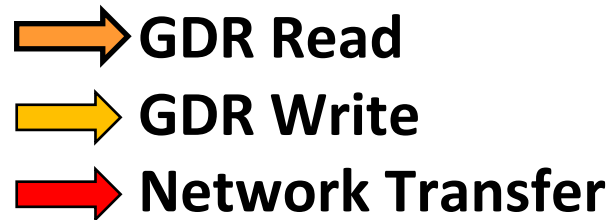
- Pipeline

$$\left[\frac{M}{C} + (n-2) \right] \times \left(t_s + \frac{C}{B_G} \right)$$

- Staging

$$\frac{M}{B_{PCIe}} + (n-1) \times \left(t_s + \frac{M}{B_H} \right)$$

Poor
Scalability



K-nomial-based Broadcast

- Direct

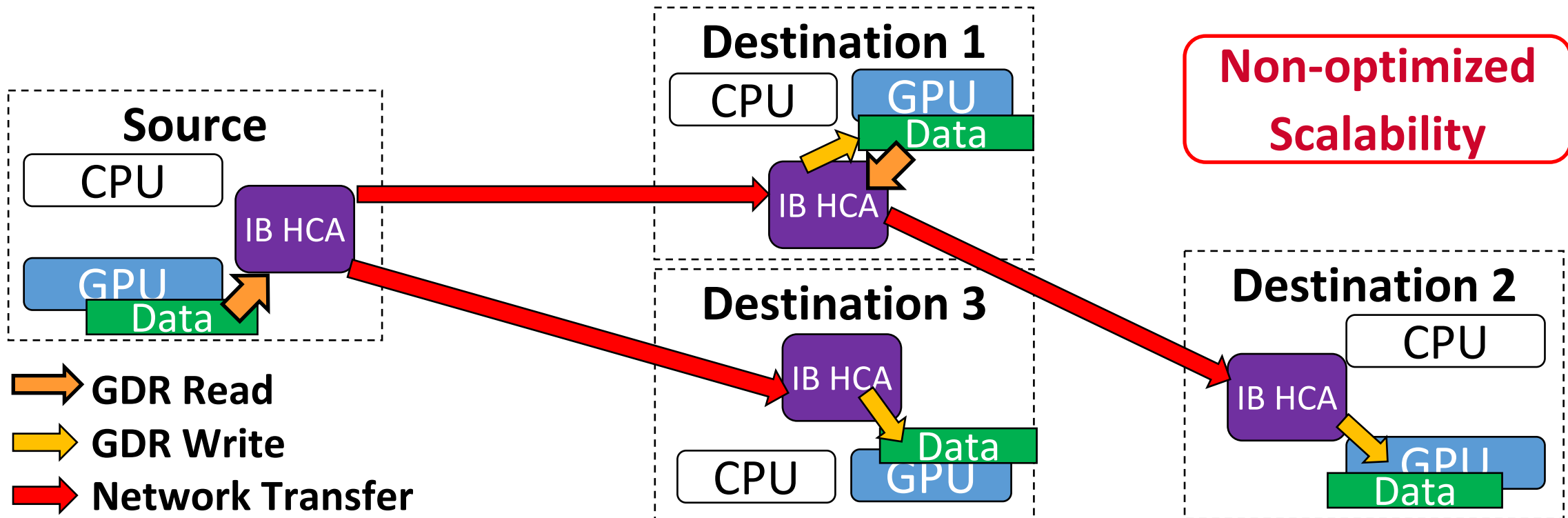
$$[\log_k n] \times \left(t_s + \frac{M}{B_G} \right)$$

- Pipeline

$$\left(\frac{M}{C} \times [\log_k n] \right) \times \left(t_s + \frac{C}{B_G} \right)$$

- Staging

$$\frac{M}{B_{PCIe}} + [\log_k n] \times \left(t_s + \frac{M}{B_H} \right)$$



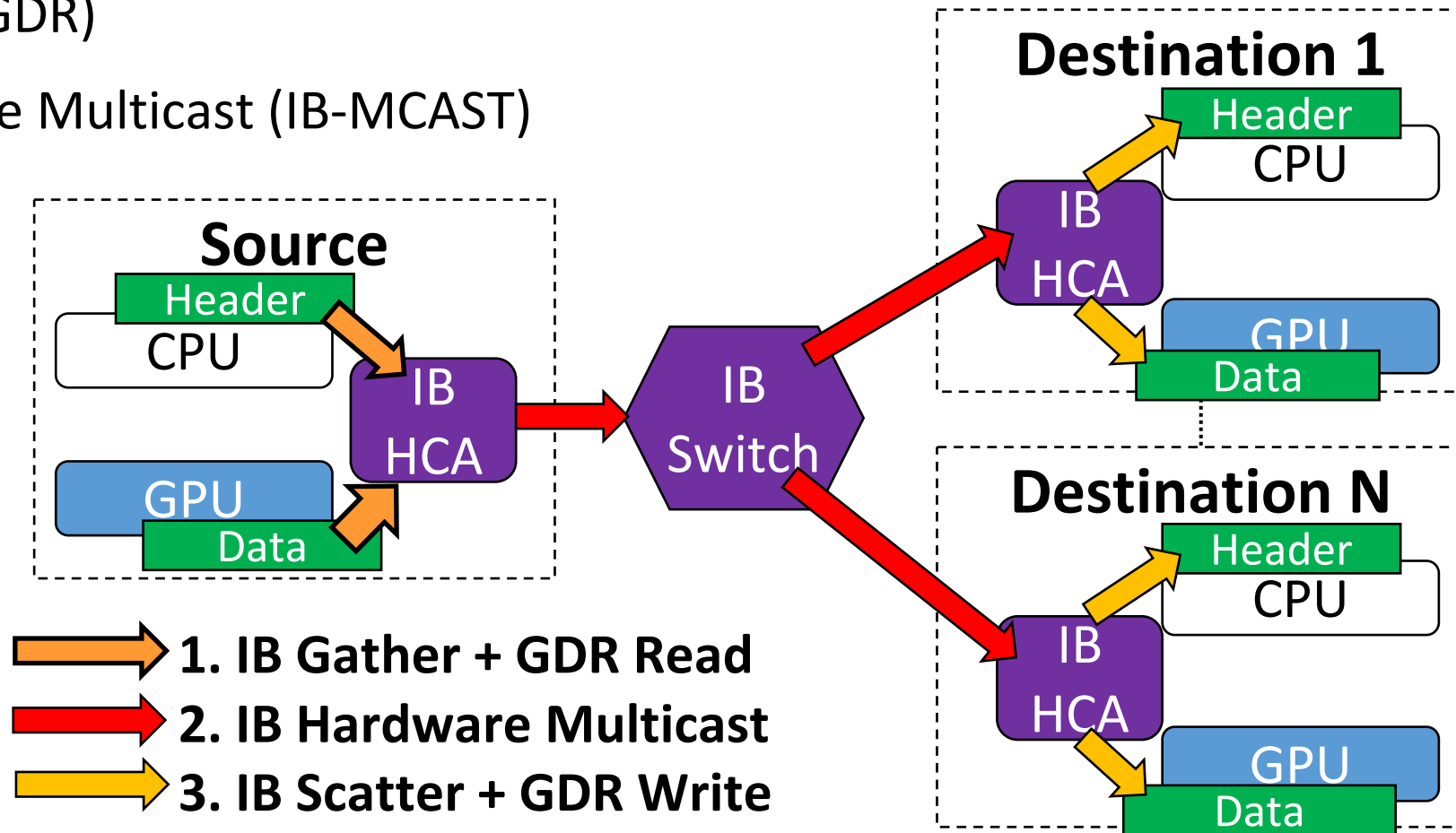
Hardware Multicast-based Broadcast*

- For GPU-resident data, using
 - GPUDirect RDMA (GDR)
 - InfiniBand Hardware Multicast (IB-MCAST)

- **Overhead**

- IB UD limit
- GDR limit

$$\frac{M}{U} \times \left(t_s + t_o(n) + \frac{U}{B_G} \right)$$



*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

- How to determine techniques to leverage **IB-MCAST** and other GPU advanced features **GDR** to design **efficient and scalable broadcast** with large messages on GPU clusters?
- How to achieve **high overlap and scalability** for multi-source broadcast operations?
- How to determine attainable theoretical and practical performance benefits for deep learning applications?

Outline

- Introduction
- Analysis
- **Proposed Design**
 - Streaming-based Design with IB multicast and NVIDIA GPUDirect features
- Performance Evaluation
- Conclusion and Future Work

Overview of Proposed Streaming Design

- **Optimized broadcast send operation**
 - **Streaming** the GPU-resident data through host memory
 - Leveraging InfiniBand hardware multicast
 - **Low-latency**: avoiding GDR Read limit
 - **Overlapping** data transfers within and across nodes
- **Optimized broadcast receive operation**
 - Zero-copy scheme by leveraging GDR feature
 - **Low-latency**: avoiding unnecessary data transfers

Optimized Broadcast Send

- **Preparing Intermediate buffer (*im_buf*)**

- Page-locked (pinned) host buffer

- Fast Device-Host data movement

- Allocated at initialization phase

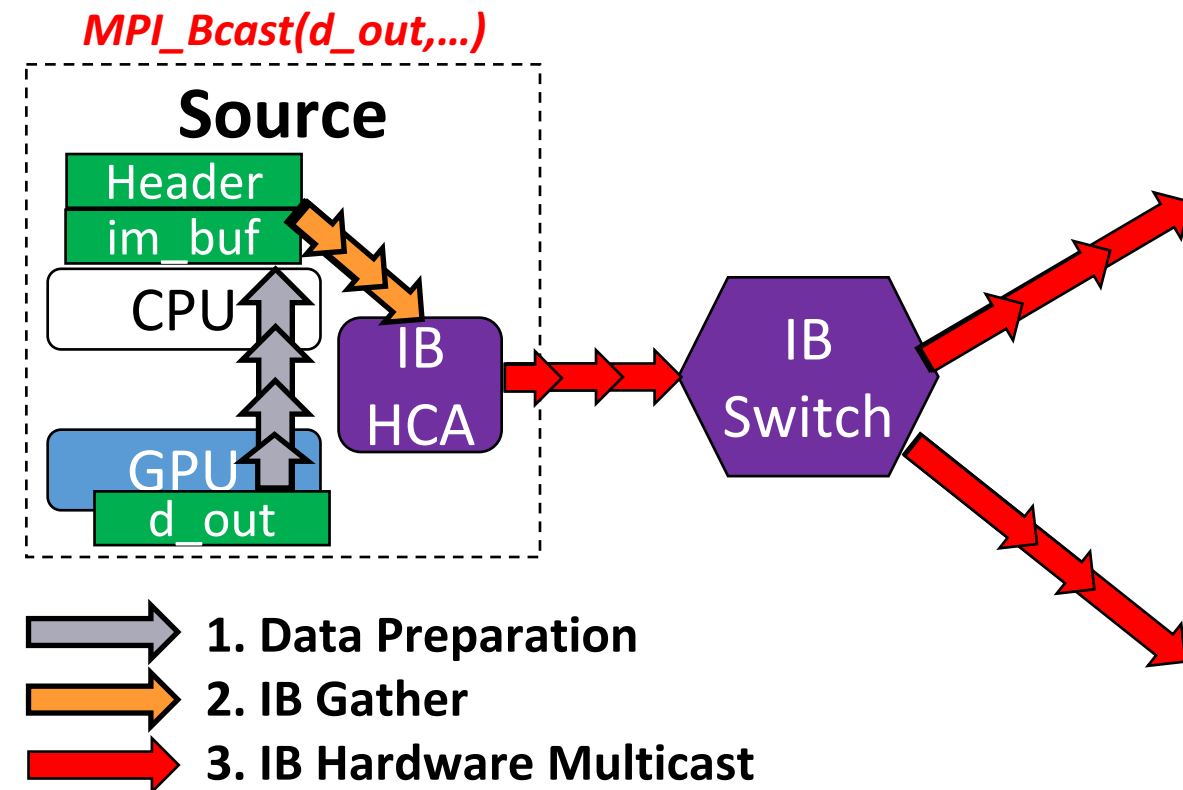
- Low overhead

- **Streaming data through host**

- Fine-tuned chunked data

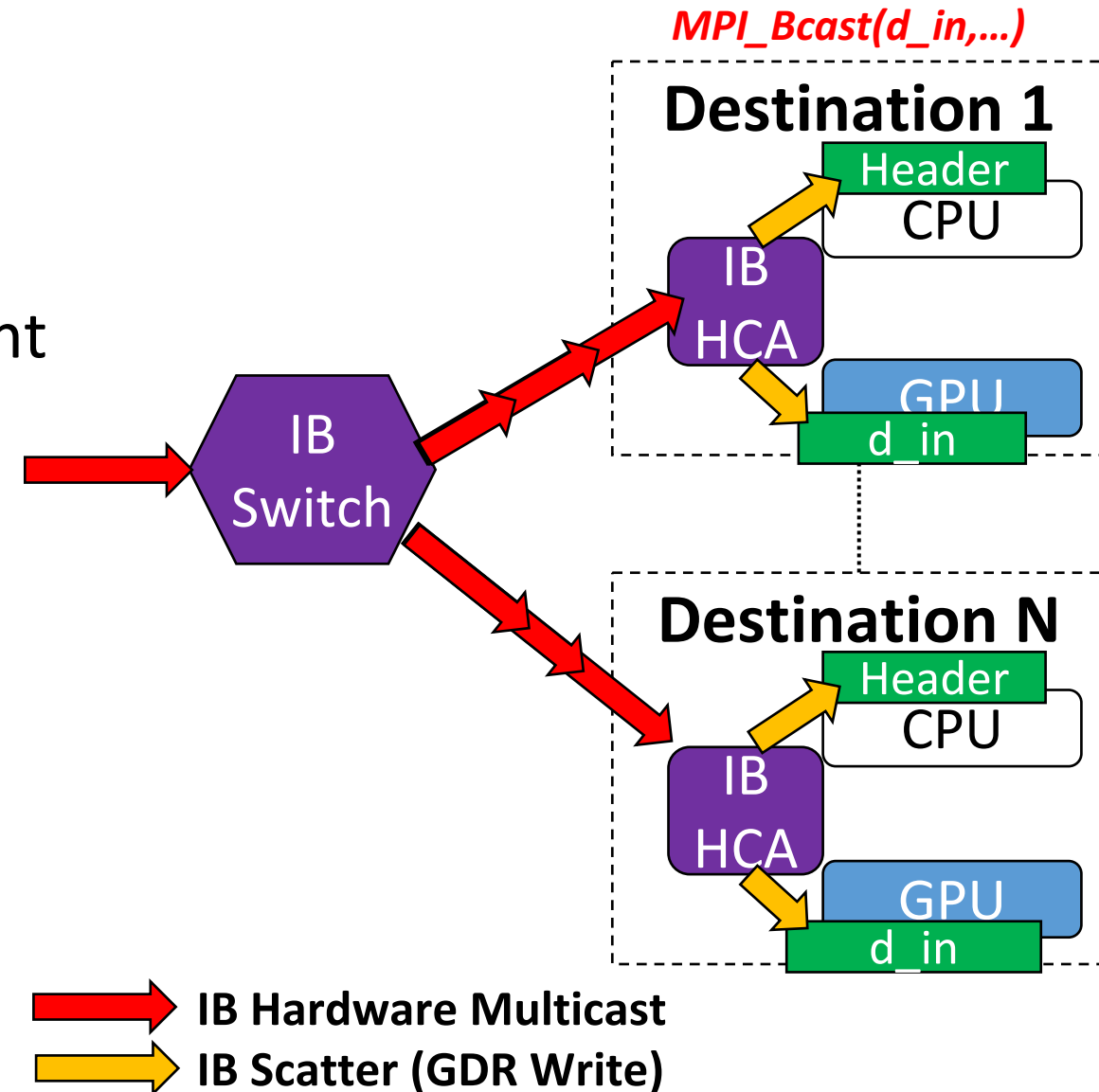
- Asynchronous copy operations

- Three-stage pipeline







Optimized Broadcast Receive

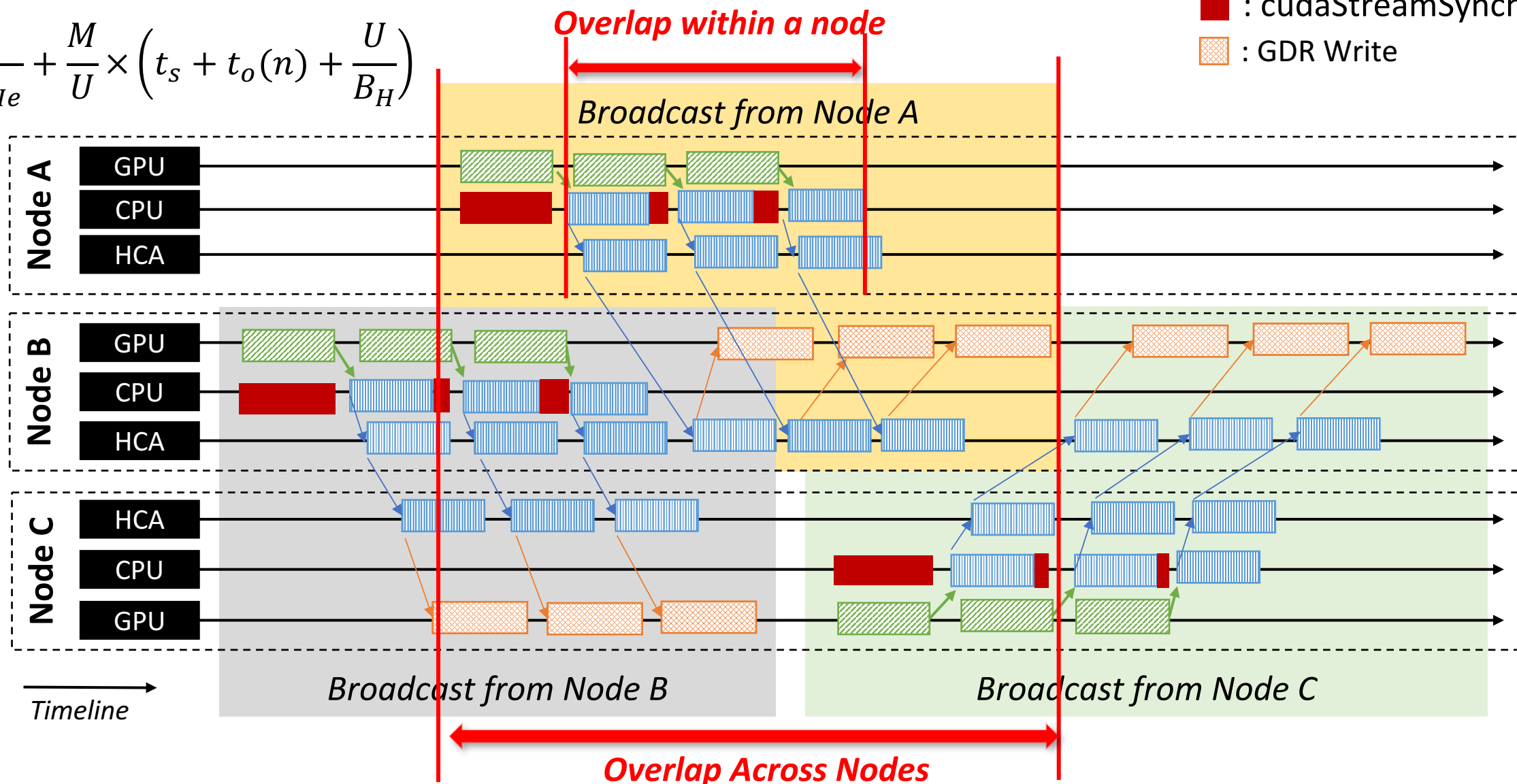
- **Zero-copy broadcast receive**
 - Pre-posted user buffer (d_in)
 - Avoids additional data movement
 - Leverages IB Scatter and GDR features
 - Low-latency
 - Free-up PCIe resources for applications



Overlap Opportunities

$$\frac{C}{B_{PCIe}} + \frac{M}{U} \times \left(t_s + t_o(n) + \frac{U}{B_H} \right)$$

-  : cudaMemcpyAsync
-  : IB Hardware Multicast
-  : cudaStreamSynchronize
-  : GDR Write



Outline

- Introduction
- Analysis
- Proposed Design
- Performance Evaluation
 - OSU Micro-Benchmark (OMB)
 - Deep Learning Framework
- Conclusion and Future Work

Overview of the MVAPICH2 Project

- **High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)**
 - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Started in 2001, First version available in 2002
 - MVAPICH2-X (MPI + PGAS), Available since 2011
 - **Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014**
 - Support for Virtualization (MVAPICH2-Virt), Available since 2015
 - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
 - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
 - **Used by more than 2,775 organizations in 85 countries**
 - **More than 420,000 (> 0.4 million) downloads from the OSU site directly**
 - Empowering many TOP500 clusters (June '17 ranking)
 - **1st, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China**
 - 15th, 241,108-core (Pleiades) at NASA
 - 20th, 462,462-core (Stampede) at TACC
 - 44th, 74,520-core (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 TFlops) ->
 - Sunway TaihuLight (1st in Jun'16, 10M cores, 100 PFlops)



Experimental Environments

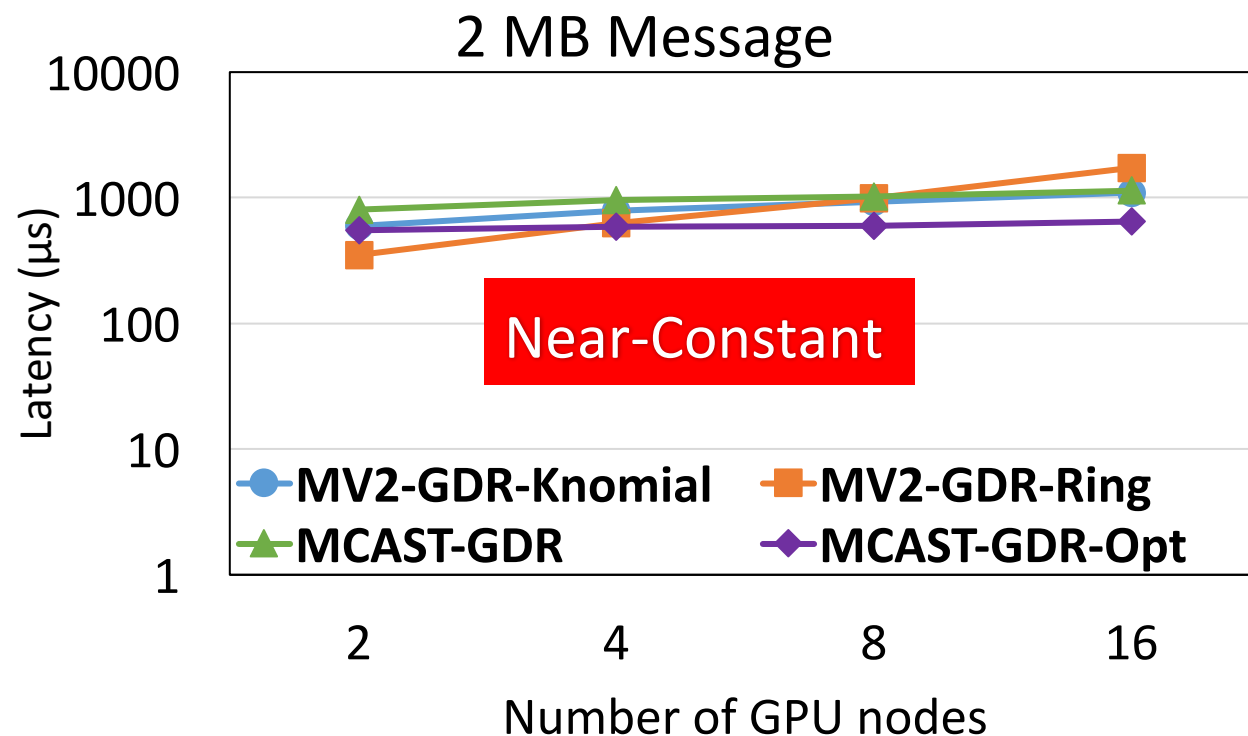
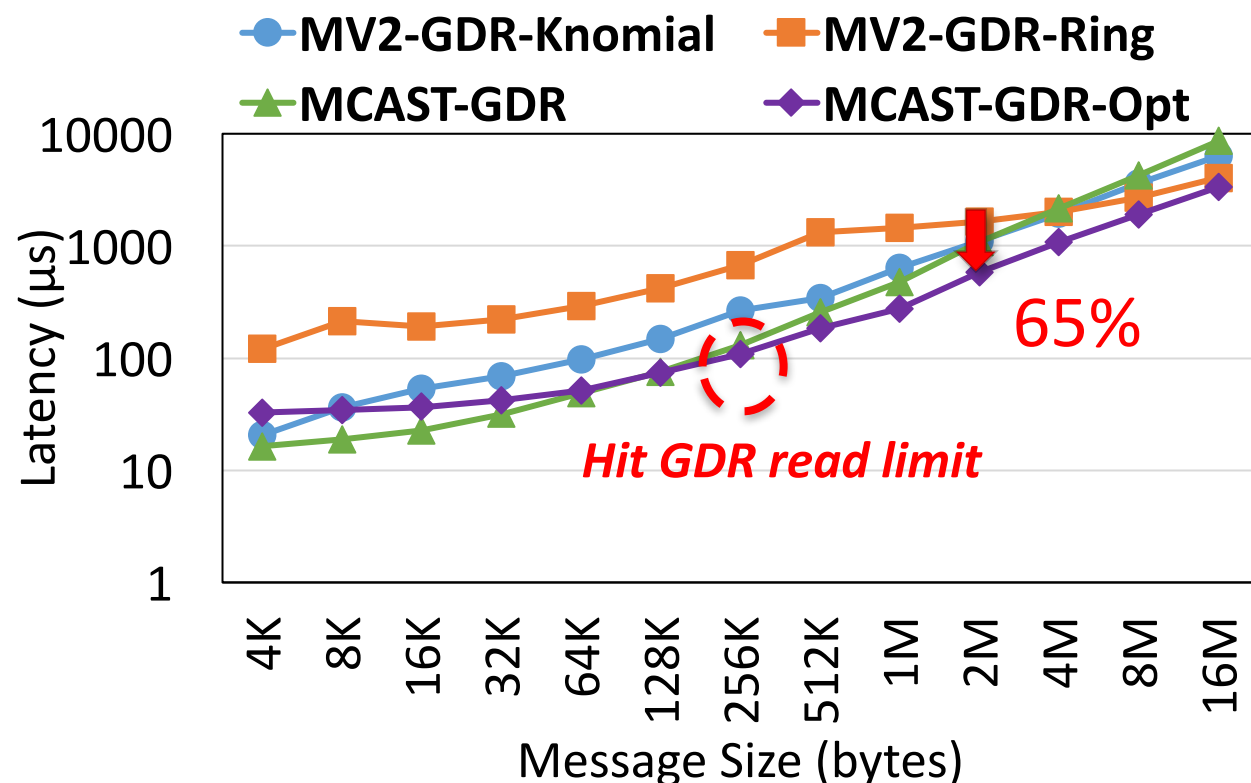
- **RI2 cluster @ The Ohio State University**
 - Two 14-core Intel (Broadwell) Xeon E5-2680 V4 processors
 - 1 NVIDIA K80 GPU per node; Used Up to **16 GPU nodes**
 - One single port InfiniBand EDR HCA
 - Mellanox SB7790 and SB7800 InfiniBand switches
- **Ohio State University (OSU) Micro-Benchmark (OMB)**
<http://mvapich.cse.ohio-state.edu/benchmarks/>
 - osu_bcast - MPI_Bcast Latency Test
- **Deep learning framework: CUDA-Aware Microsoft Cognitive Toolkit (CA-CNTK)***
 - AlexNet and VGG models with ImageNet dataset

*D. S. Banerjee, K. Hamidouche and D. K. Panda, "Re-Designing CNTK Deep Learning Framework on Modern GPU Enabled Clusters," IEEE CloudCom, Luxembourg City, 2016, pp. 144-151.

Evaluation: Benchmark Evaluation

- @ RI2 cluster, 16 GPUs, 1 GPU/node

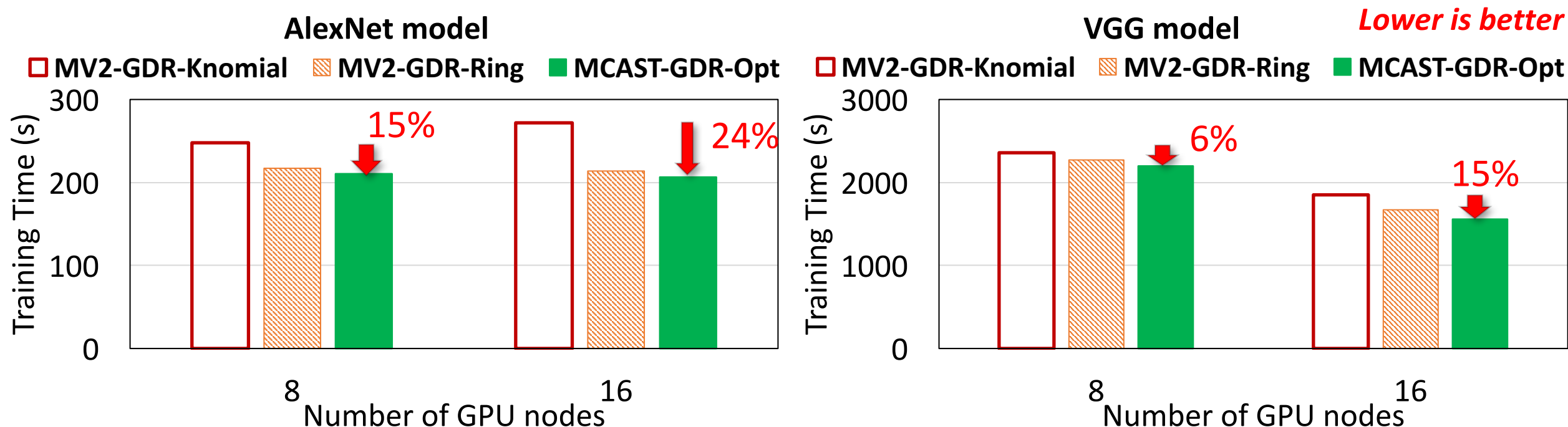
Lower is better



- Provide near-constant latency over the system sizes
- Reduces up to 65% of latency for large messages

Evaluation: Deep Learning Frameworks

- @ RI2 cluster, 16 GPUs, 1 GPU/node:
 - CUDA-Aware Microsoft Cognitive Toolkit (CA-CNTK) **without modification**

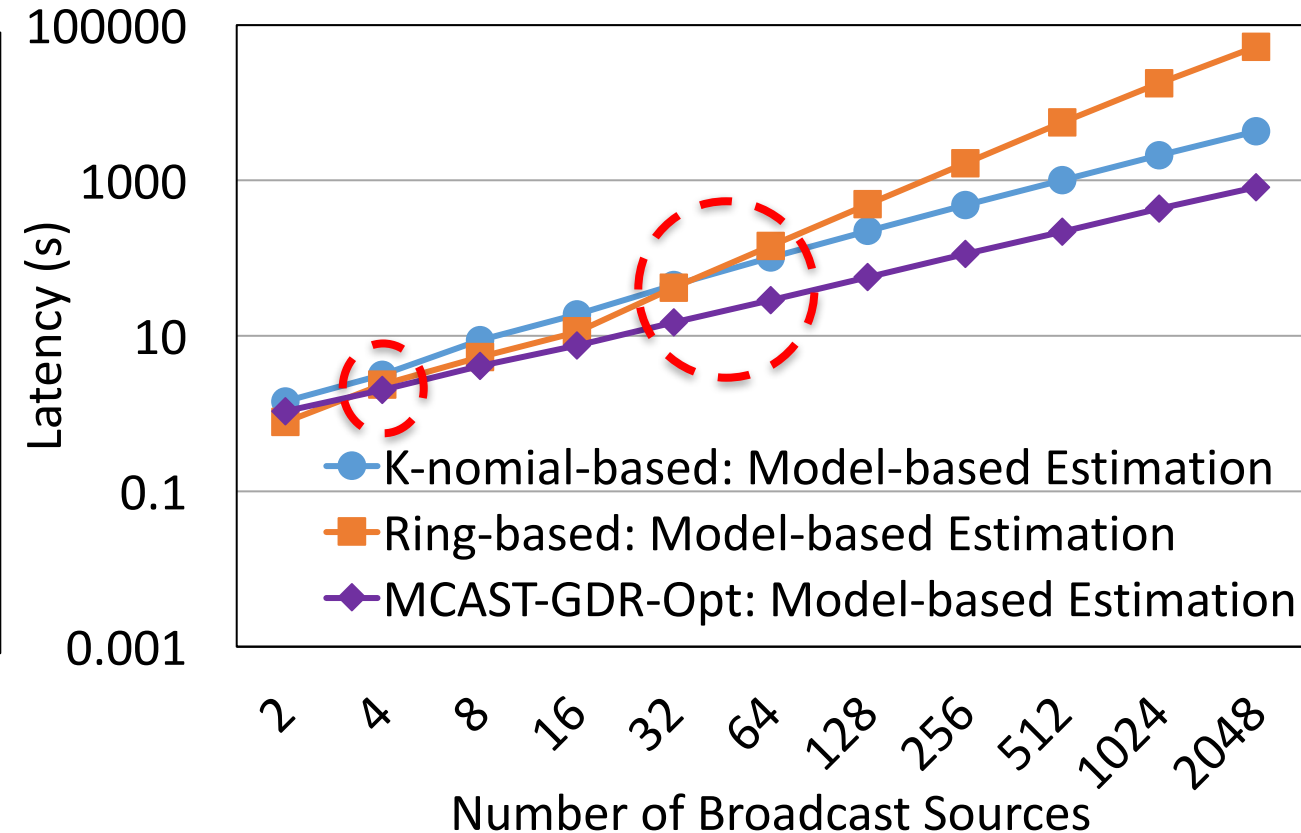
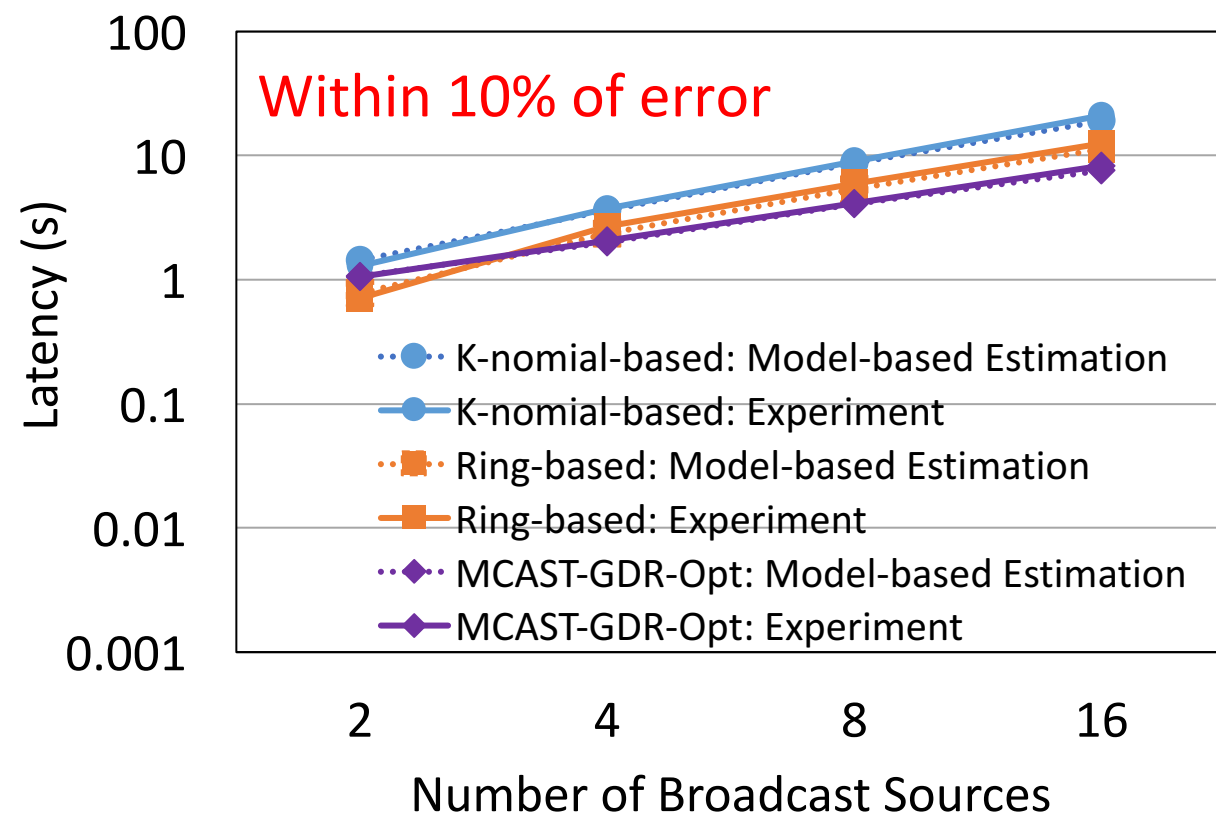


- Reduces up to 24% and 15% of latency for AlexNet and VGG models
- Higher improvement is expected for larger system sizes

Performance Prediction

- Based on the architecture on RI2 cluster

$$M = 2MB; C = 512 KB; U = 4 KB; B_H \approx 100 Gbps; B_{PCIe} = 8 Gbps; t_o(n) \approx \frac{1}{\alpha} \times \ln(n), 15 \leq \alpha \leq 20$$



Outline

- Introduction
- Analysis
- Proposed Design
- Performance Evaluation
- Conclusion and Future Work

Conclusion

- Proposed efficient broadcast schemes to **leverage GDR and MCAST features** for deep learning applications
 - Optimized **streaming design for large messages** transfers
- Provided and evaluated **analytical models** to capture essential performance behavior of alternative broadcast schemes on GPU clusters
- These features are included in the latest release of MVAPICH2-GDR library

Future Work

- **Extend the design for other broadcast-based collective algorithms as well as non-blocking operations**
 - Allreduce, Allgather, ..., and so on
- **Evaluate the proposed design in upcoming larger-scale GPU clusters**



THE OHIO STATE
UNIVERSITY

Thank You!

ENGILITY
Engineered to Make a Difference

**Ching-Hsiang Chu, Xiaoyi Lu, Ammar A. Awan, Hari Subramoni,
Jahanzeb Hashmi, Bracy Elton and Dhabaleswar K. (DK) Panda**

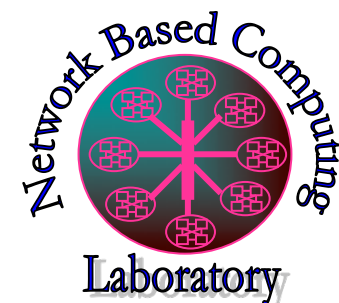
{chu.368, lu.932, awan.10, subramoni.1, hashmi.29}@osu.edu

bracy.elton@engilitycorp.com, panda@cse.ohio-state.edu



MVAPICH

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.

MCAST-based Broadcast

- **NVIDIA GPUDirect^[1]**
 - Remote direct memory access (RDMA) transfers between GPUs and other PCIe devices \Rightarrow **GDR**
 - and more...
- **InfiniBand (IB) hardware multicast (IB MCAST)^[2]**
 - Enables efficient designs of broadcast operations
 - Host-based^[3]
 - GPU-based^[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.