MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

THE OHIO STATE UNIVERSITY

COLLEGE OF ENGINEERING

# High-Performance Adaptive MPI Derived Datatype Communication for Modern Multi-GPU Systems


Ching-Hsiang Chu, Jahanzeb Maqbool Hashmi, Kawthar Shafie Khorassani, Hari Subramoni, **Dhabaleswar K. (DK) Panda**

The Ohio State University

{chu.368, hashmi.29, shafiekhorassani.1}@osu.edu,

{subramon, panda}@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~panda

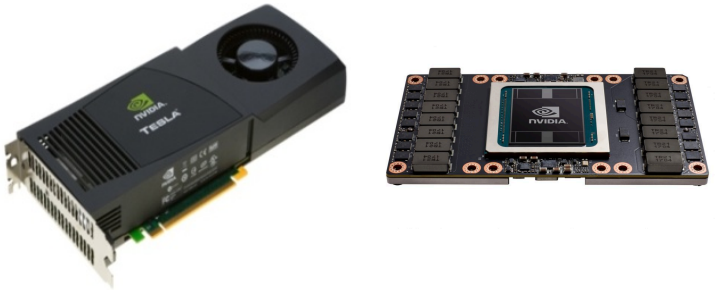
# Outline

- **Introduction**
- Problem Statement
- Proposed Designs
- Performance Evaluation
- Concluding Remarks

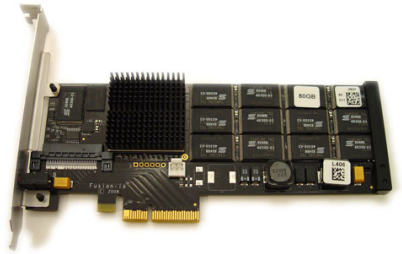
# Drivers of Modern HPC Cluster Architectures


Multi-core Processors

High Performance Interconnects - InfiniBand
<1usec latency, 200Gbps Bandwidth>




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



SSD, NVMe-SSD, NVRAM

- Multi-core/many-core technologies
- Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
- Solid State Drives (SSDs), Non-Volatile Random-Access Memory (NVRAM), NVMe-SSD
- Multiple Accelerators (NVIDIA GPGPUs and Intel Xeon Phi) connected by PCIe/NVLink interconnects
- Available on HPC Clouds, e.g., Amazon EC2, NSF Chameleon, Microsoft Azure, etc.


***Summit***

***Sierra***

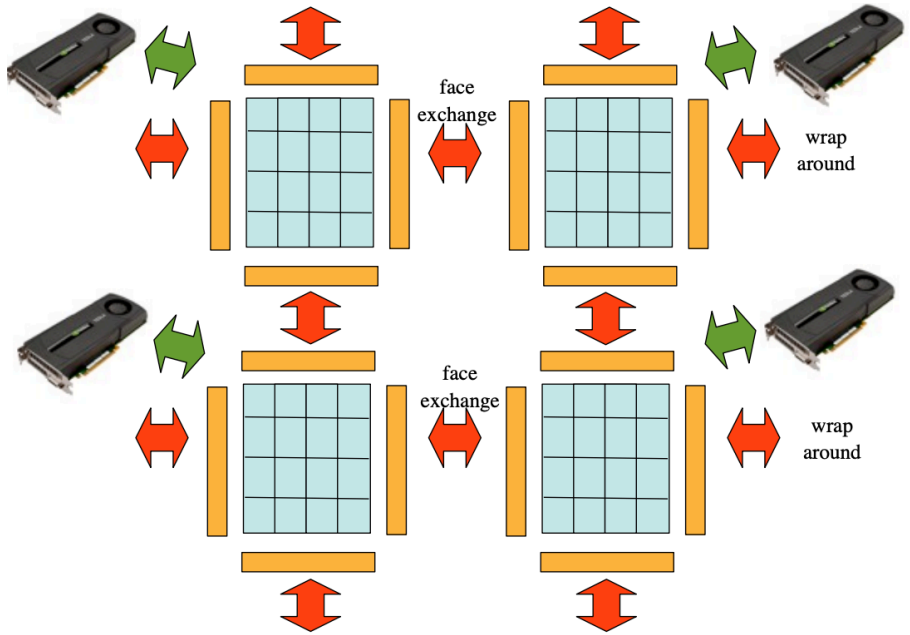
***Sunway TaihuLight***


***K - Computer***

# Non-contiguous Data Transfer for HPC Applications

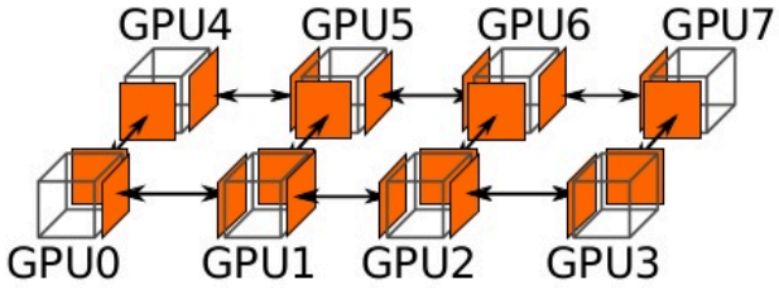
- Wide usages of MPI derived datatype for Non-contiguous Data Transfer
  - Requires Low-latency and high overlap processing

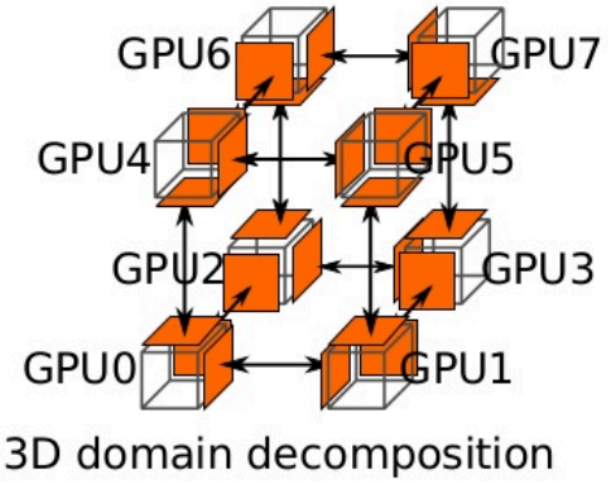
## Quantum Chromodynamics




Mike Clark. "GPU Computing with QUDA, "Developer Technology Group, https://www.olcf.ornl.gov/wp-content/uploads/2013/02/Clark_M_LQCD.pdf

## Weather Simulation: COSMO model

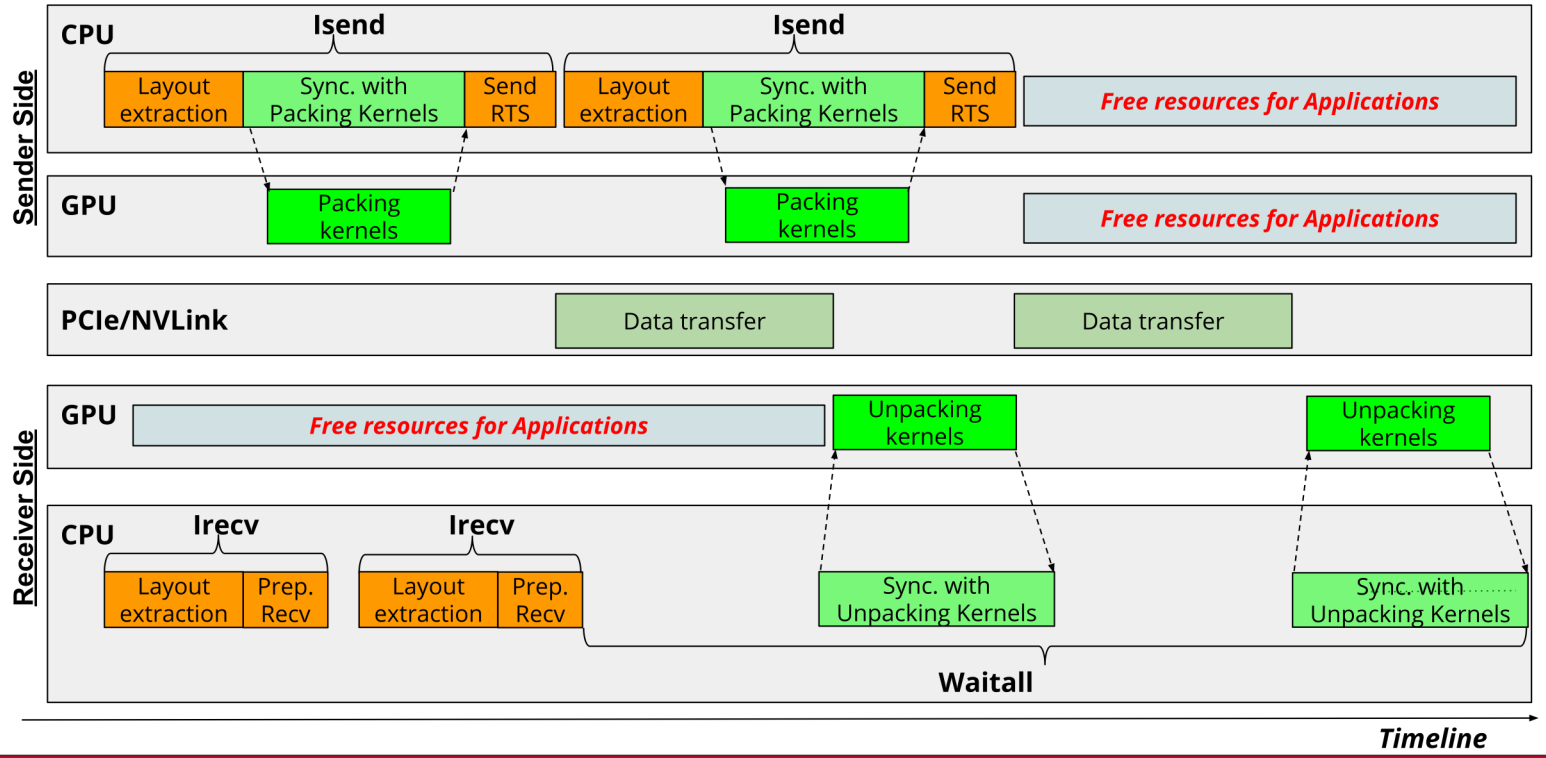



M. Martinasso, G. Kwasniewski, S. R. Alam, Thomas C. Schulthess, and T. Hoefler. "A PCIe congestion-aware performance model for densely populated accelerator servers. " SC 2016

# State-of-the-art MPI Derived Datatype Processing

- GPU kernel-based packing/unpacking [1-3]
  - High-throughput memory access
  - Leverage GPUDirect RDMA capability





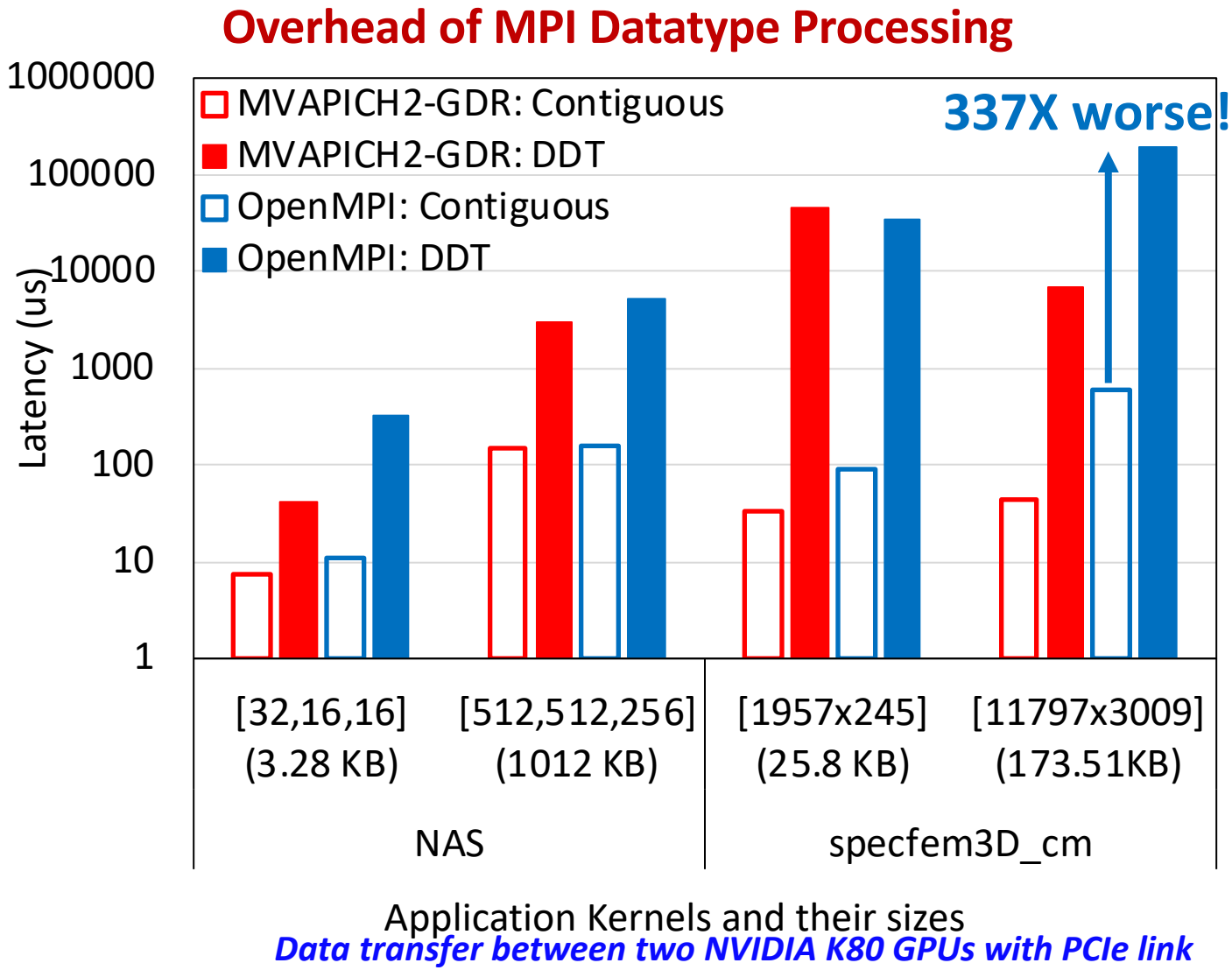
[1] R. Shi, X. Lu, S. Potluri, K. Hamidouche, J. Zhang and D. K. Panda, "HAND: A Hybrid Approach to Accelerate Non-contiguous Data Movement Using MPI Datatypes on GPU Clusters," ICPP 2014.

[2] C. Chu, K. Hamidouche, A. Venkatesh, D. Banerjee , H. Subramoni, and D. K. Panda, Exploiting Maximal Overlap for Non-Contiguous Data Movement Processing on Modern GPU-enabled Systems, IPDPS 2016.

[3] Wei Wu, George Bosilca, Rolf vandeVaart, Sylvain Jeaugey, and Jack Dongarra. "GPU-Aware Non-contiguous Data Movement In Open MPI, " HPDC 2016.

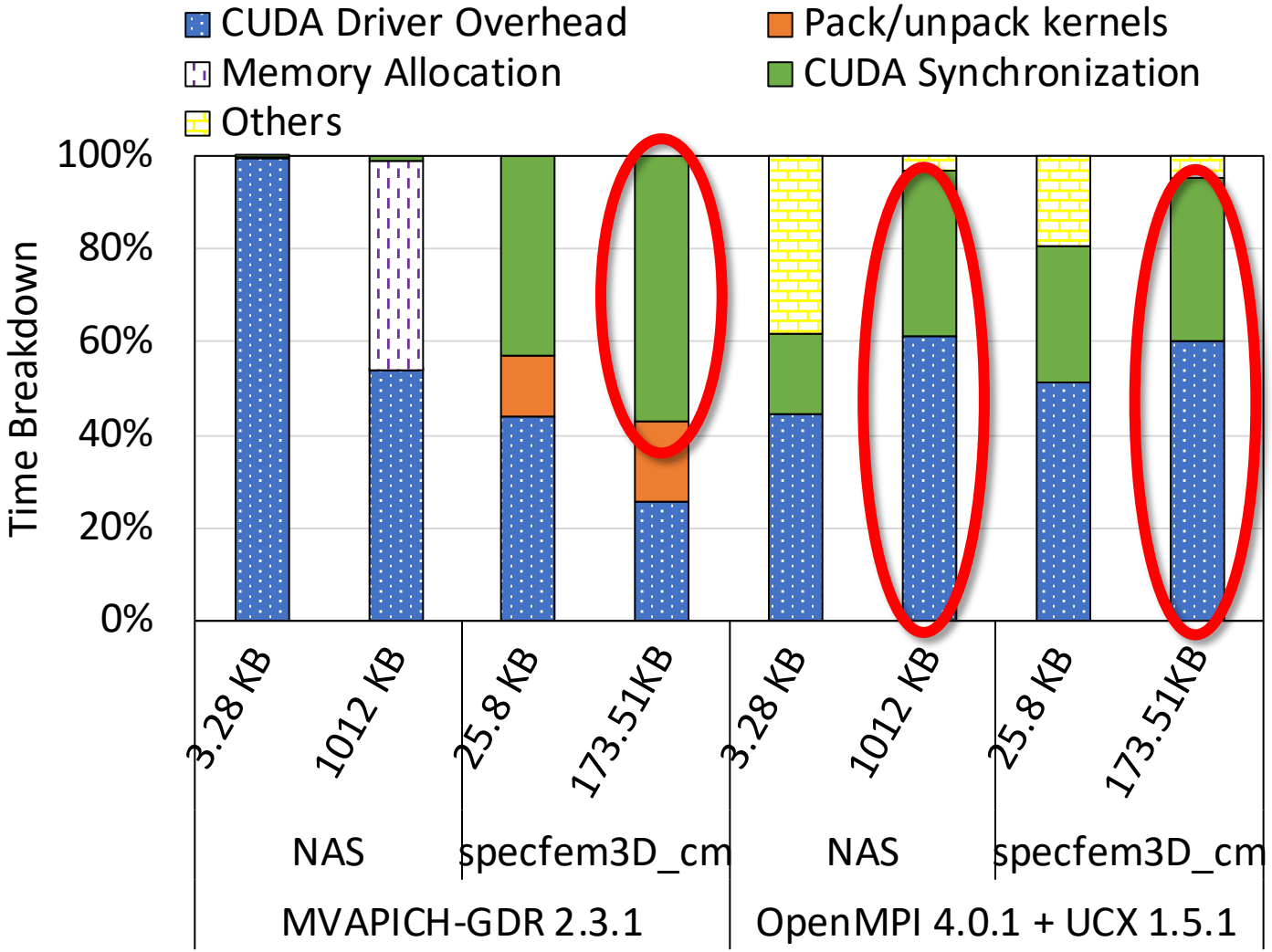
# Expensive Packing/Unpacking Operations in GPU-Aware MPI

- Significant overhead when moving non-contiguous GPU-resident data
  - Wasting cycles
  - Extra data copies
  - **High Latency!!!**



# Analysis of Packing/Unpacking Operations in GPU-Aware MPI

- Primary overhead
  - Packing/Unpacking
  - CPU-GPU synchronization
  - GPU driver overhead
- Can we reduce or **eliminate** the expensive packing/unpacking operations?




*Data transfer between two NVIDIA K80 GPUs with PCIe link*

# Outline

- Introduction
- **Problem Statement**
- Proposed Designs
- Performance Evaluation
- Concluding Remarks

# Problem Statement

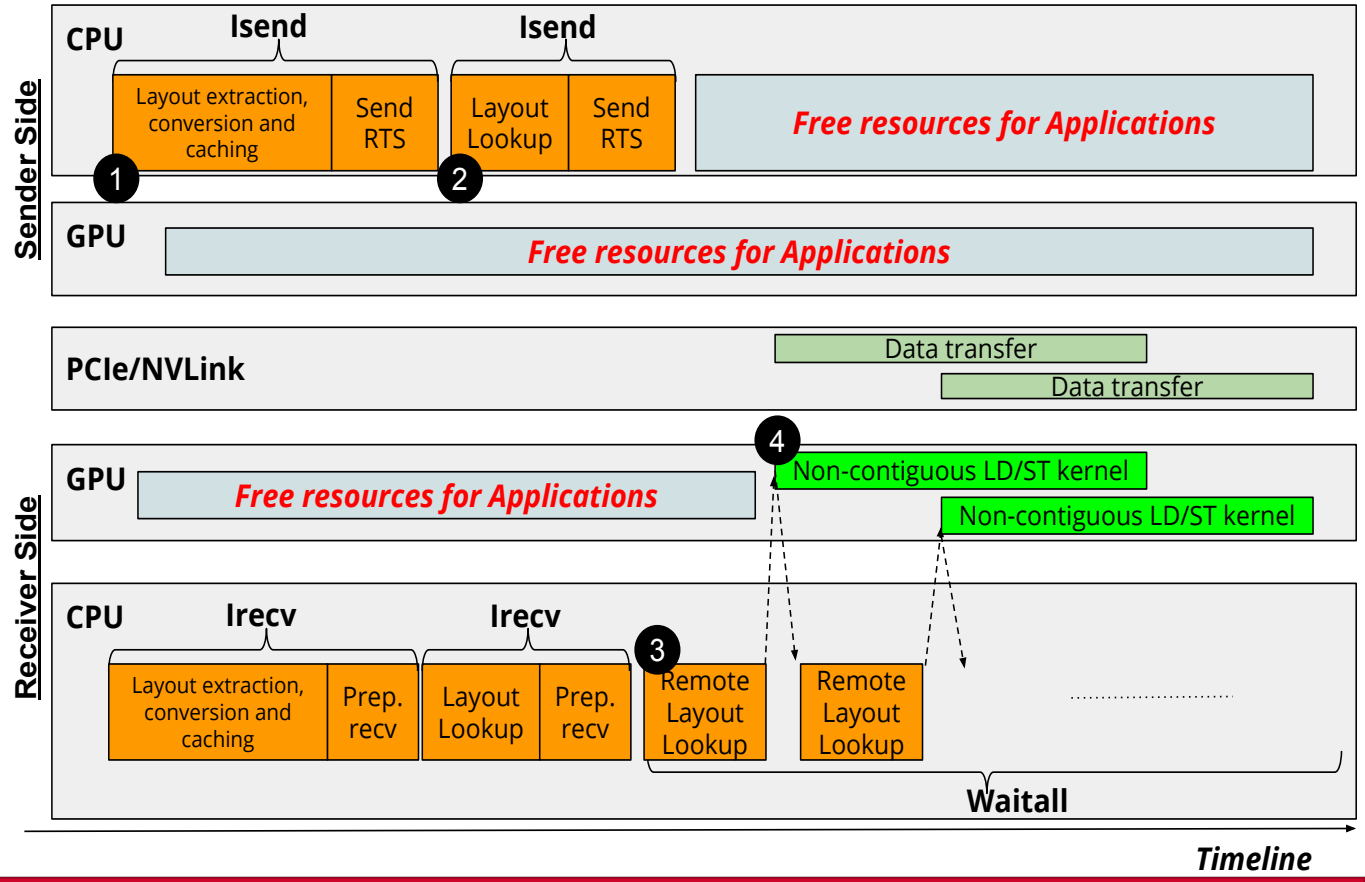
- How can we exploit load-store based remote memory access model over high-performance interconnects like PCIe and NVLink to achieve “packing-free” non-contiguous data transfers for GPU-resident data?
- Can we propose new designs that mitigate the overheads of existing approaches and offer optimal performance for GPU based derived datatype transfers when packing/unpacking approaches are inevitable?
- How to design an adaptive MPI communication runtime that can dynamically employ optimal DDT processing mechanisms for diverse application scenarios?

# Outline

- Introduction
- Problem Statement
- **Proposed Designs**
  - Zero-copy non-contiguous data movement over NVLink/PCIe
  - One-shot packing/unpacking
  - Adaptive MPI derived datatype processing
- Performance Evaluation
- Concluding Remarks

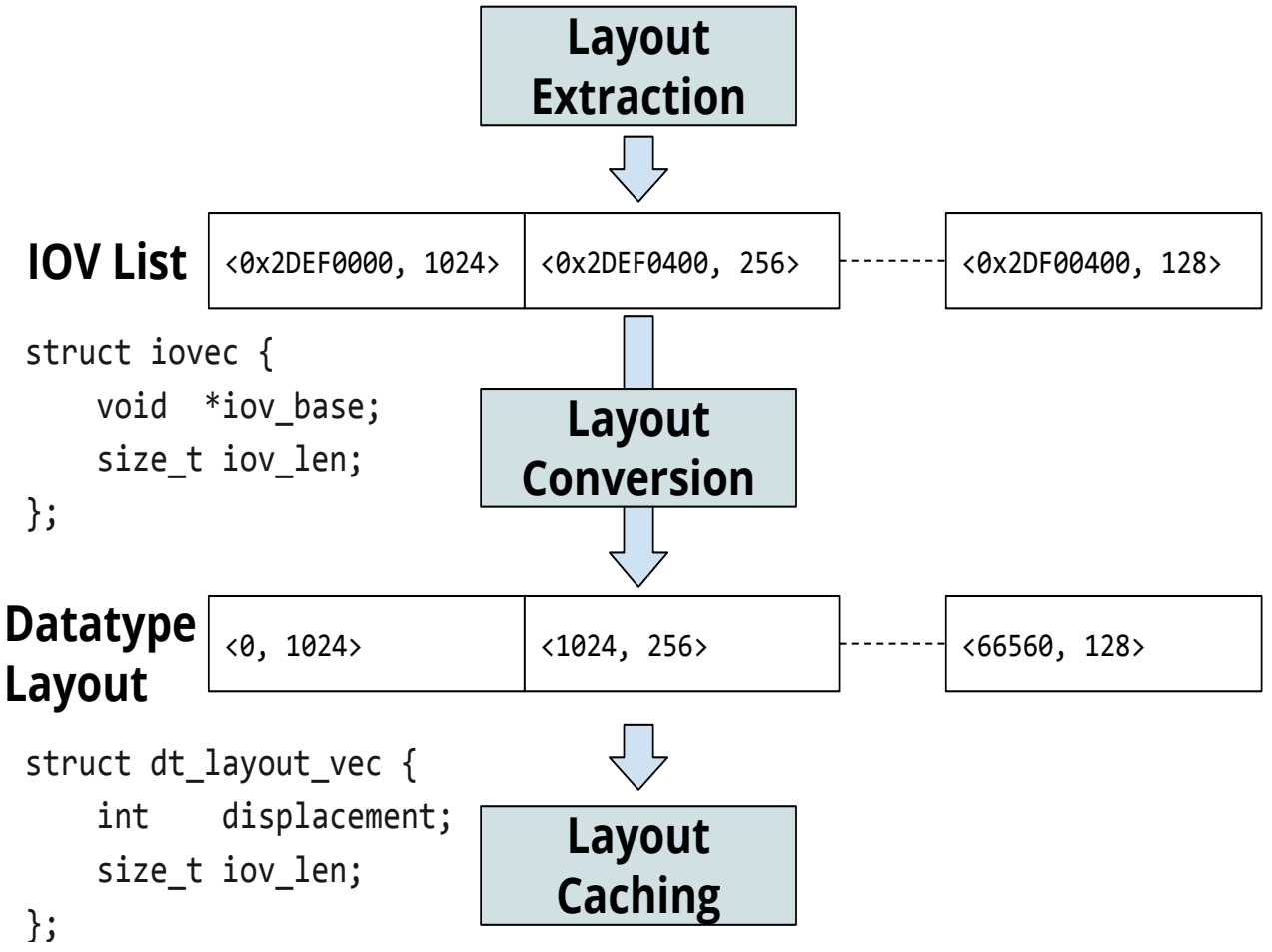
# Overview of Zero-copy Datatype Transfer

- Direct link such as PCIe/NVLink is available between two GPUs
- Efficient datatype layout exchange and cache
- Load-store data movement



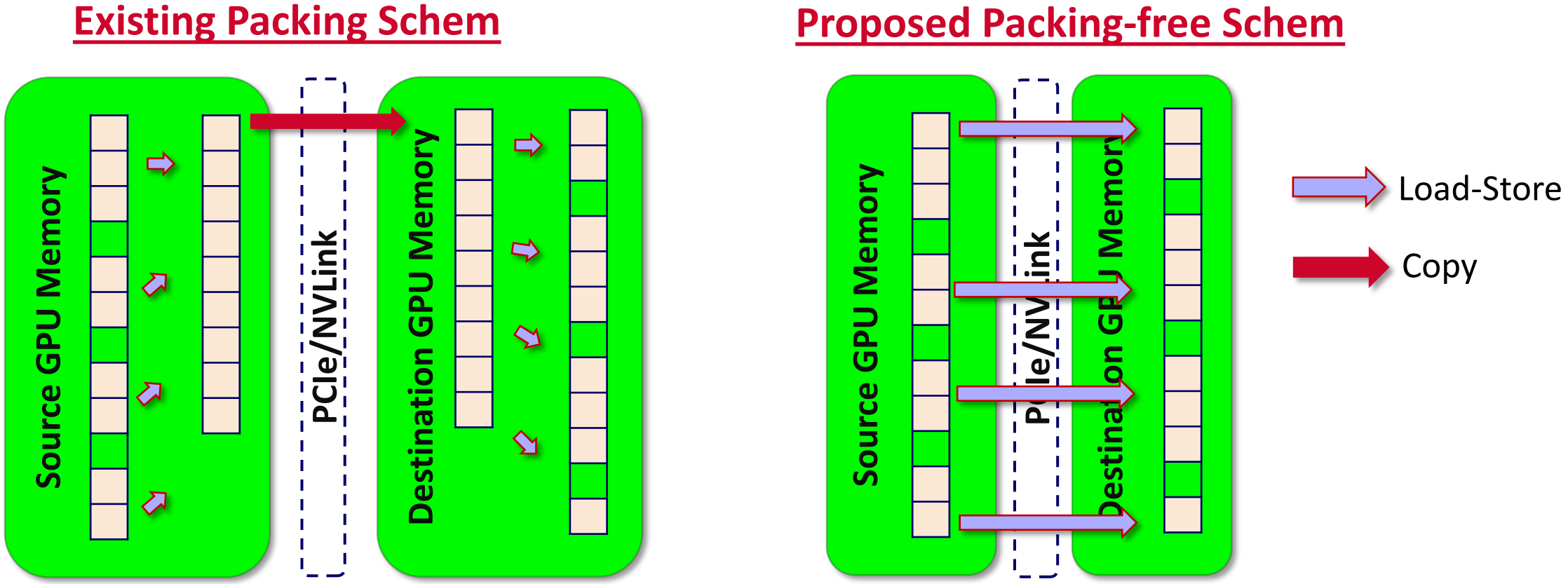
# Zero-copy Datatype Transfer: Enhanced Layout Cache

- Convert IOV list to displacement list
  - Improved reusability
  - One-time effort
- Cache datatype layout on the shared system memory
  - Accessible within the node without extra copies



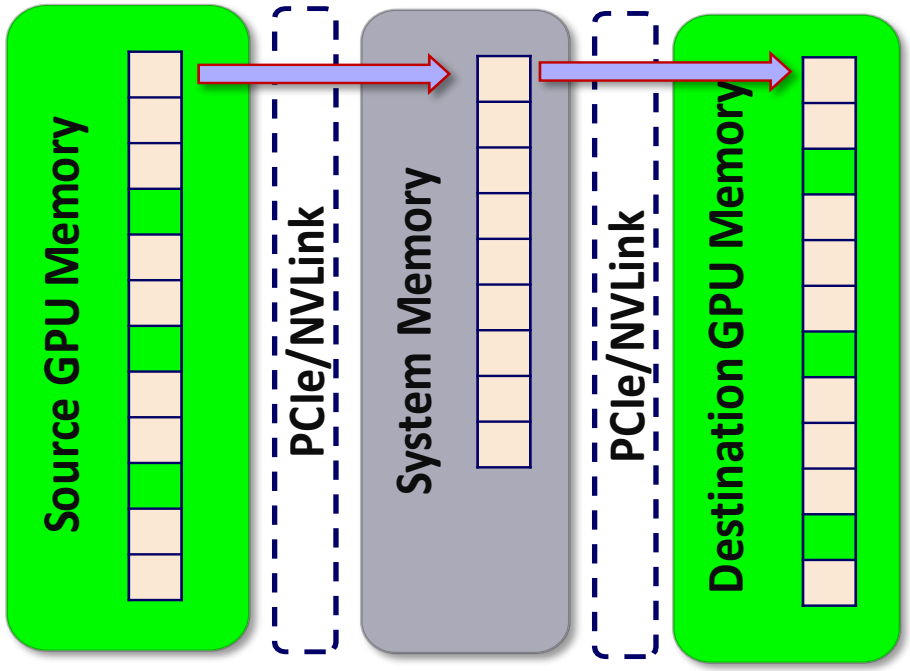
# Zero-copy Datatype Transfer: Copy vs. Load-Store

- Exploiting load-store capability of modern interconnects
  - Eliminate extra data copies and expensive packing/unpacking processing



# One-shot Packing/Unpacking Mechanism

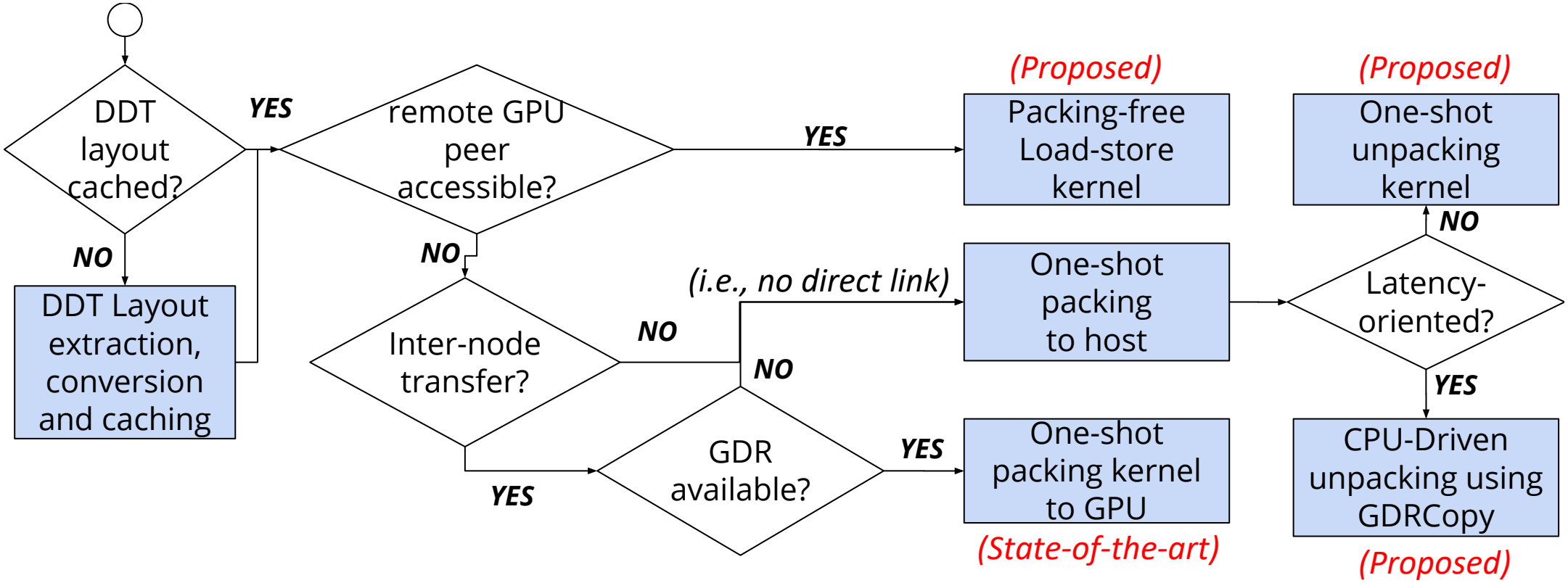
- Packing/unpacking is inevitable if there is no direct link
- Direct packing/unpacking between CPU and GPU memory to avoid extra copies




1. GDRCopy-based
   - CPU-driven low-latency copy-based scheme
2. Kernel-based
   - GPU-driven high-throughput load-store-based scheme

# Adaptive Selection

- Availability of GPUDirect peer access and GPUDirect RDMA
- Latency- or throughput-oriented communication pattern



# Outline

- Introduction
- Problem Statement
- Proposed Designs
- **Performance Evaluation**
- Concluding Remarks

# Experimental Environments

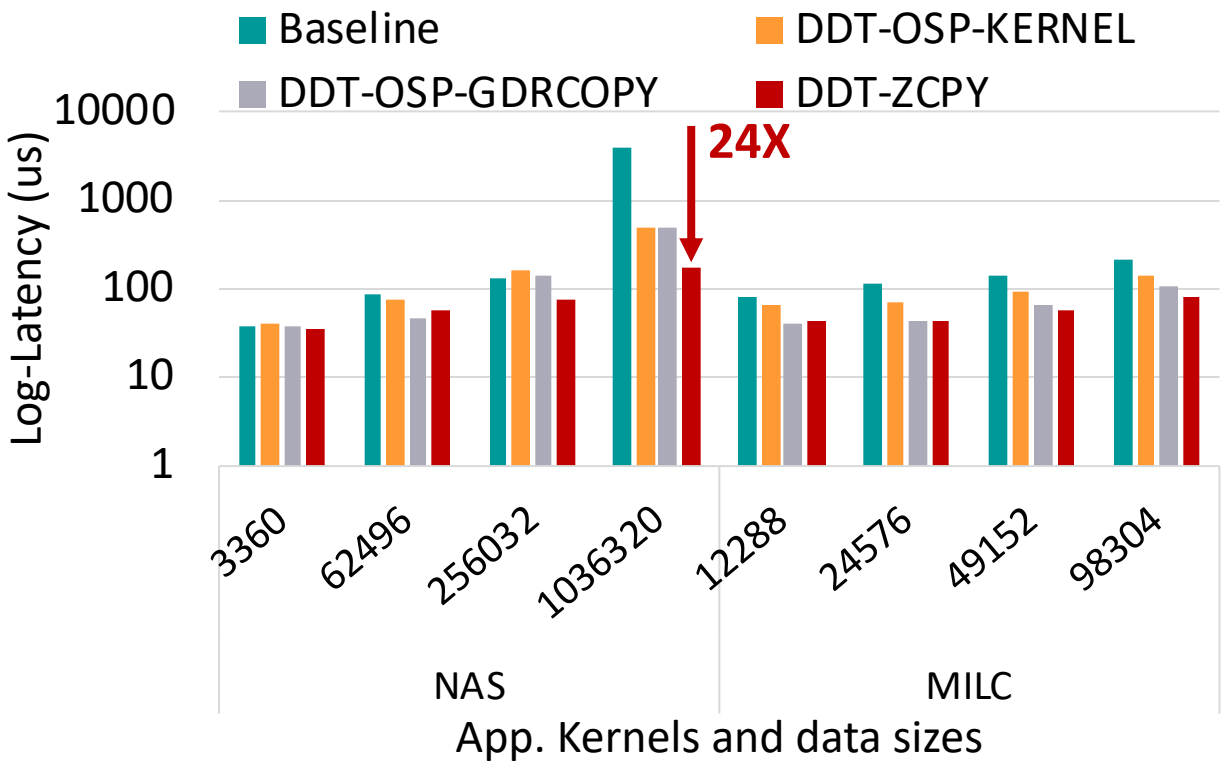
| | Cray CS-Storm | NVIDIA DGX-2 |
|---|---|---|
| CPU Model | Intel Haswell | Intel Skylake |
| System memory | 256 GB | 1.5 TB |
| GPUs | 8 NVIDIA Tesla K80 | 16 NVIDIA Tesla V100 |
| Interconnects | PCIe Gen3<br>Mellanox IB FDR | NVLink/NVSwitch<br>Mellanox IB EDR x 8 (Unused) |
| OS & compiler version | RHEL 7.3 & GCC 4.8.5 | Ubuntu 18.04 & GCC 7.3.0 |
| NVIDIA driver & CUDA versions | 410.79 & 9.2.148 | 410.48 & 9.2.148 |

- Benchmarks: Modified DDTBench to use GPU-resident data
  - NAS_MG, MILC, Specfem3D_cm, and Specfem3D_oc
- Application kernels
  - COSMO model & Jacobi Method
- Baseline: MVAPICH2-GDR 2.3.1

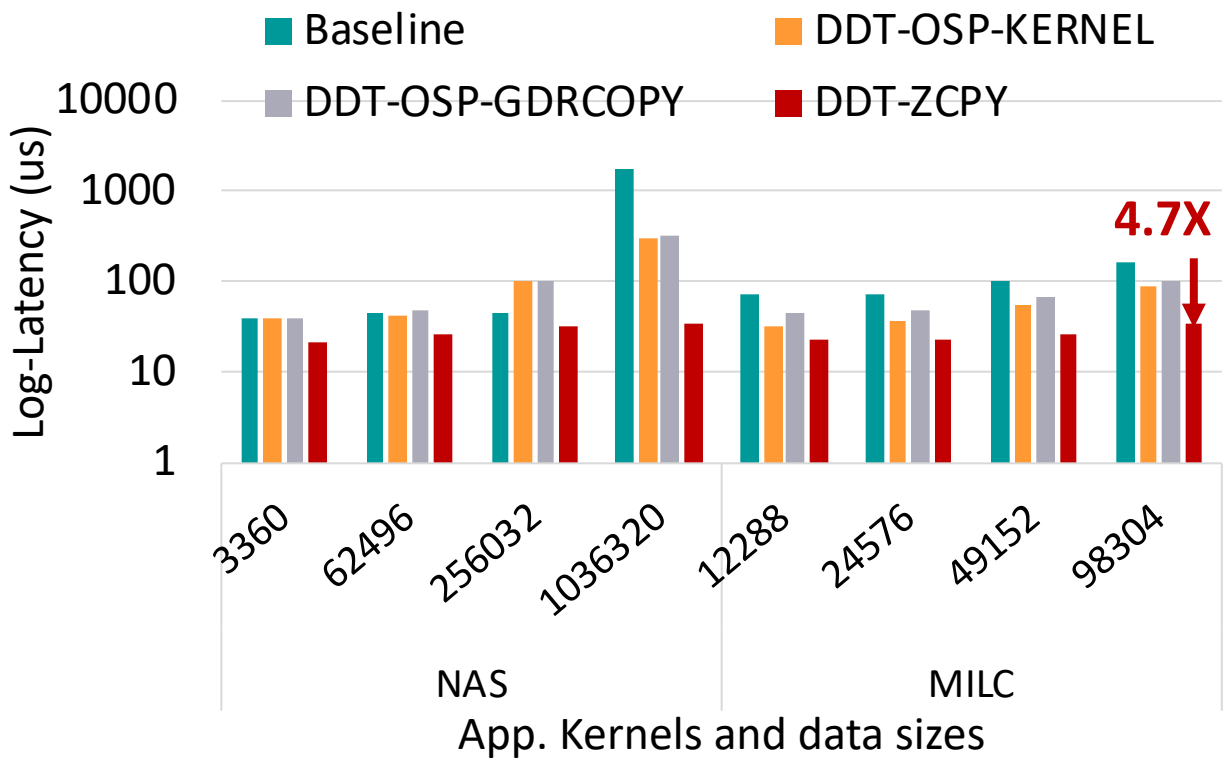
# Overview of the MVAPICH2 Project

- 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-3.1), 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 3,050 organizations in 89 countries**
  - **More than 615,000 (> 0.6 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Jun '19 ranking)
    - 3rd, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center i
    - 5th, 448, 448 cores (Frontera) at TACC
    - 8th, 391,680 cores (ABCI) in Japan
    - 15th, 570,020 cores (Neurion) in South Korea and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  - **http://mvapich.cse.ohio-state.edu** Partner in the TACC Frontera System
- Empowering Top500 systems for over a decade

# Evaluation of Zero-copy Design: Dense Layout




*Platform: Cray CS-Storm; Two GPUs through PCIe Switch*

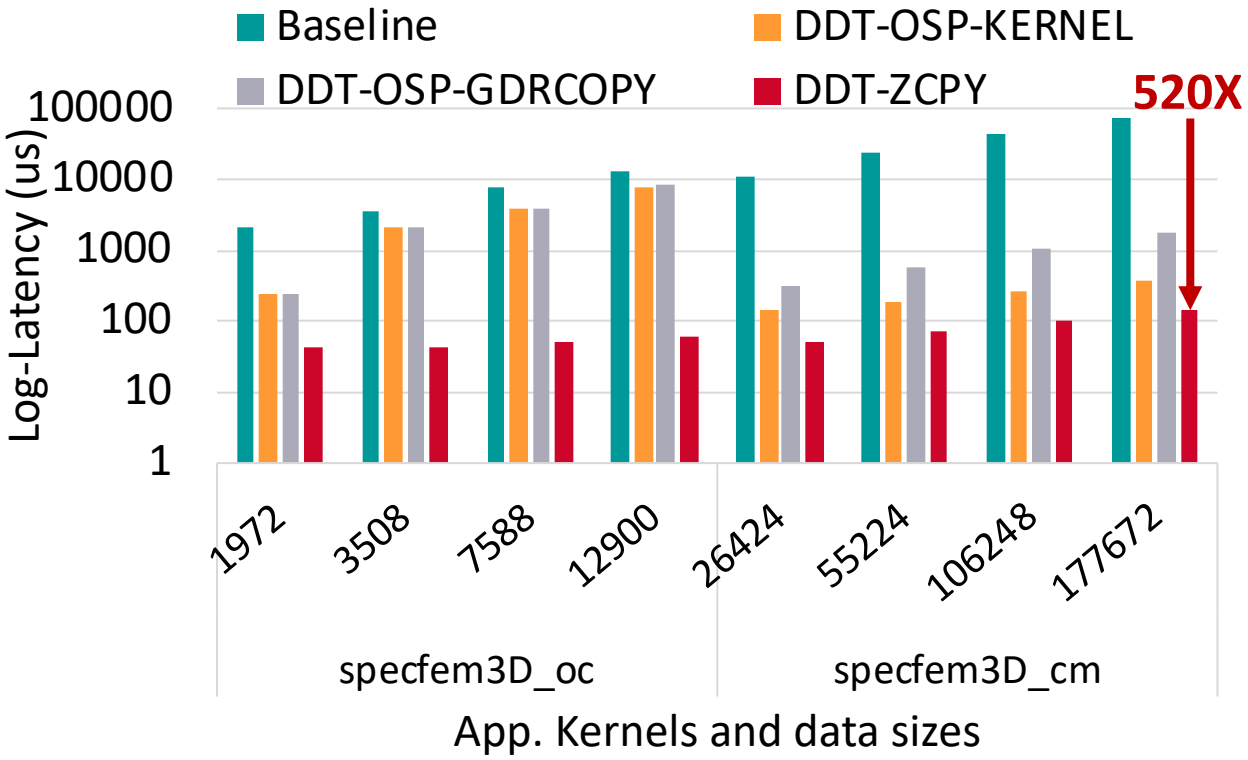



*Platform: NVIDIA DGX-2; Two GPUs through NVLink/NVSwitch*

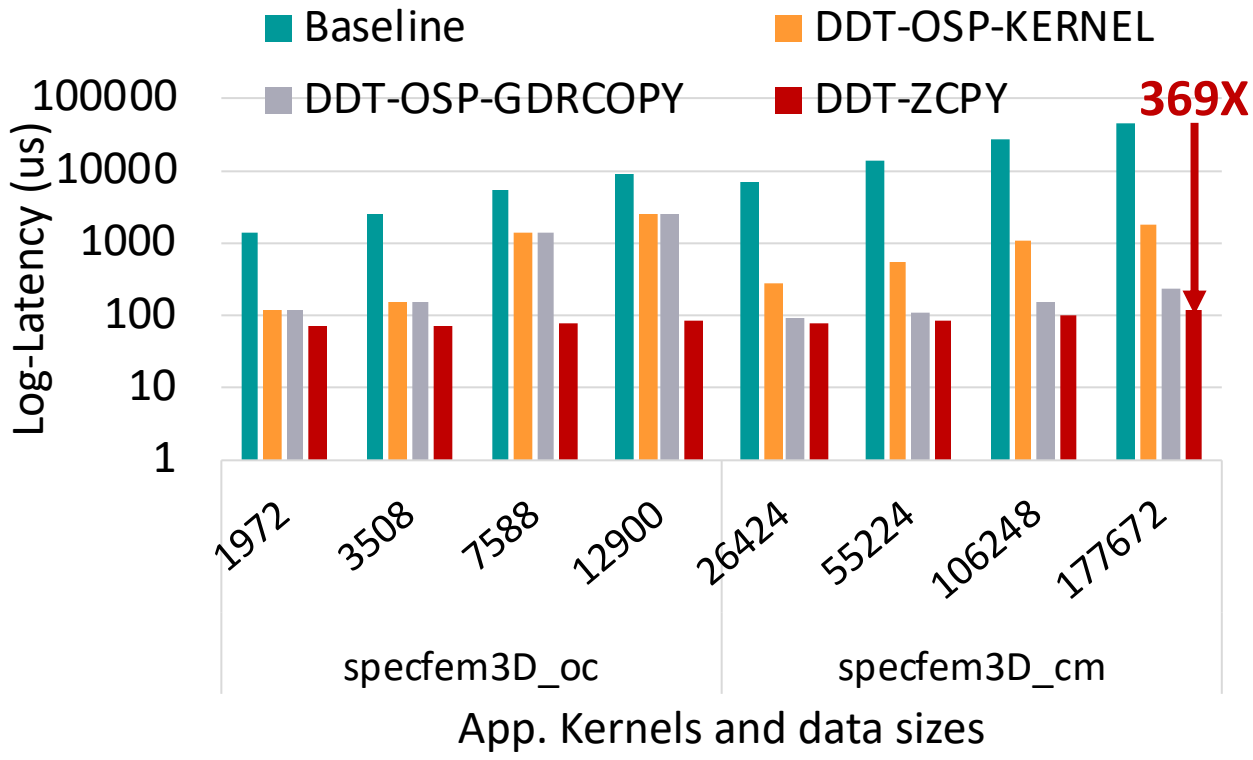
- Zero-copy performs the best in almost all cases!

*Please refer to the paper for more performance comparison!*

# Evaluation of Zero-copy Design: Sparse Layout



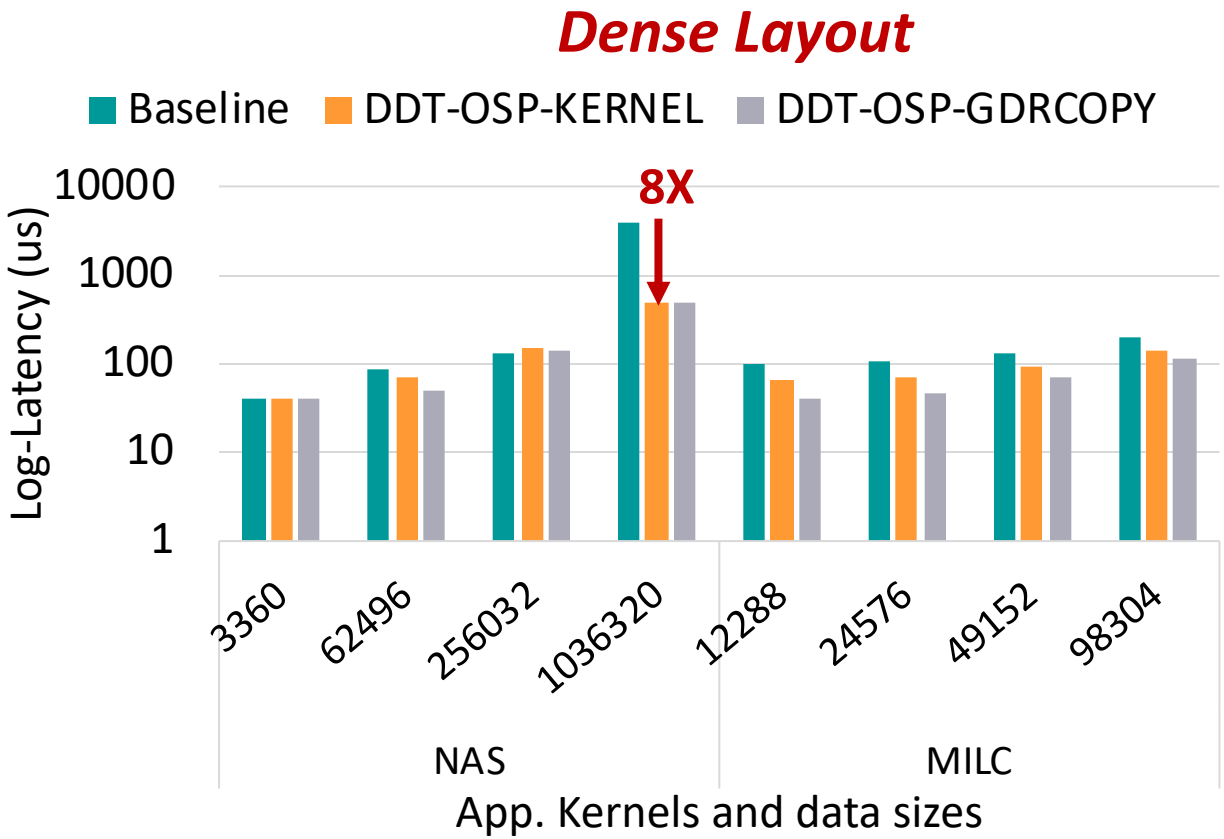

*Platform: Cray CS-Storm; Two GPUs through PCIe Switch*

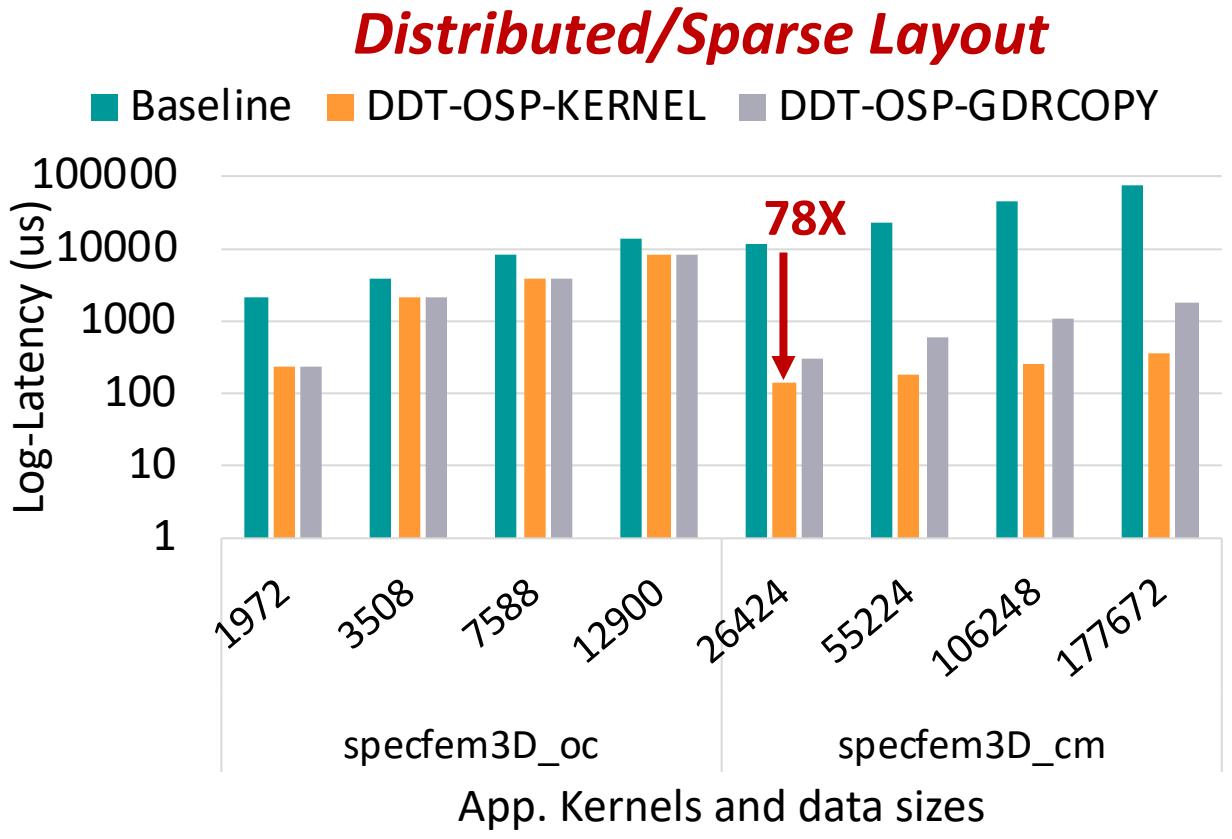



*Platform: NVIDIA DGX-2; Two GPUs through NVLink/NVSwitch*

- Zero-copy performs the best in all cases by avoiding unnecessary data copies and CPU-GPU synchronization

# Evaluation of One-shot Packing Design



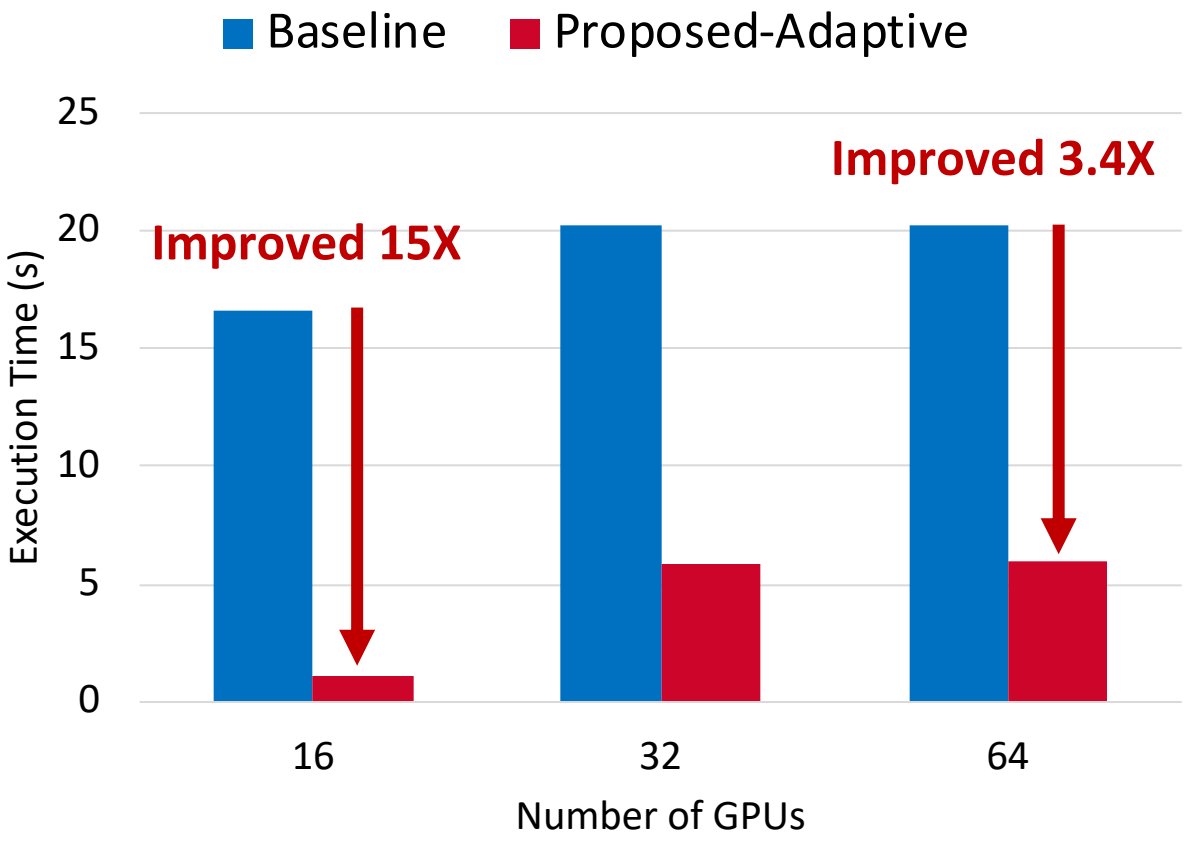

***Platform: Cray CS-Storm; Two GPUs on different sockets without direct link***

- GDRCopy-based scheme performs better for dense layout
- Kernel-based scheme performs better for sparse layout

***Please refer to the paper for more performance comparison!***

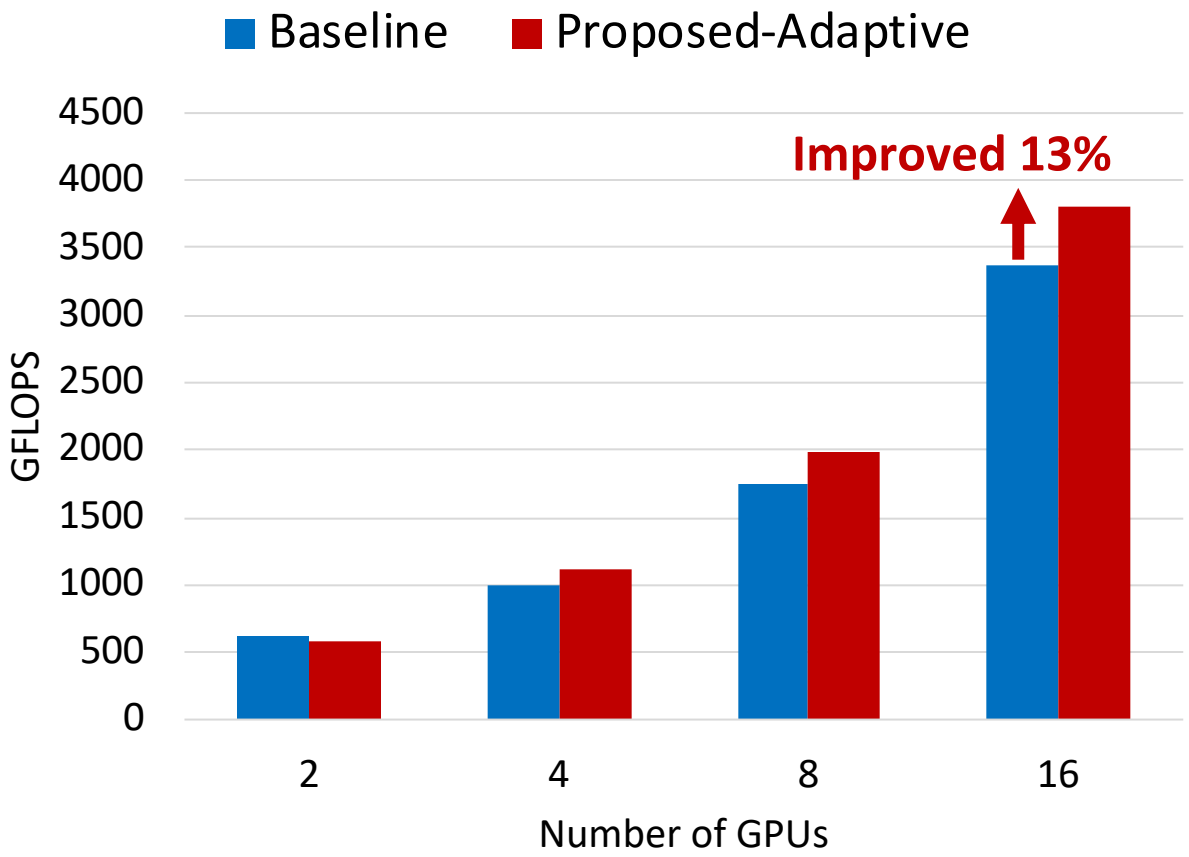
# Evaluation of Applications

- COSMO Model
  (https://github.com/cosunae/HaloExchangeBenchmarks)




*Platform: Cray CS-Storm, 8 NVIDIA K80 GPUs per node*

- Jacobi (2DStencil Computation)




*Platform: NVIDIA DGX-2, 16 NVIDIA V100 GPUs per node*

# Outline

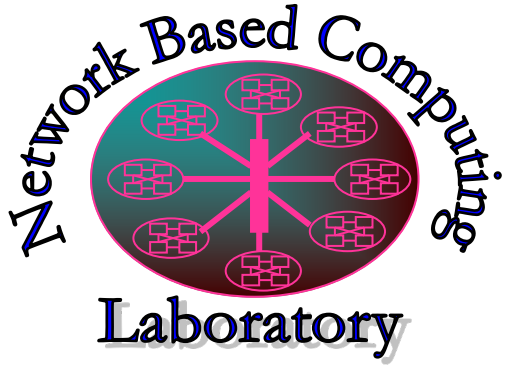
- Introduction
- Problem Statement
- Proposed Designs
- Performance Evaluation
- **Concluding Remarks**

# Concluding Remarks

- Non-contiguous data communication is common in many HPC applications
  - however, it is not optimized in current GPU-Aware MPI implementations
- Proposed designs significantly reduce the packing overhead
  - Zero-copy design eliminates expensive packing/unpacking operations
  - One-shot design avoids extra data copies
  - Adaptive scheme dynamically selects the optimal communication paths
- Publicly available since MVAPICH2-GDR 2.3.2 release
  - http://mvapich.cse.ohio-state.edu/

# Thank You!

panda@cse.ohio-state.edu




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/