15th ANNUAL WORKSHOP 2019

# Designing High-Performance MPI Collectives in MVAPICH2 for HPC and Deep Learning

**Hari Subramoni**

The Ohio State University
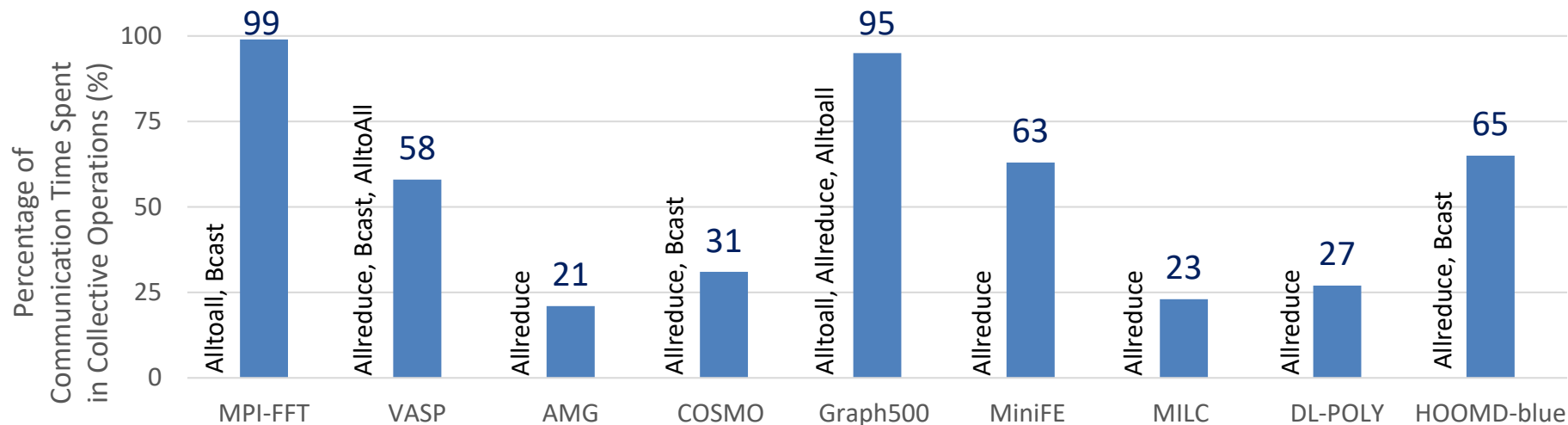
E-mail: subramon@cse.ohio-state.edu

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

**Dhabaleswar K. (DK) Panda**

The Ohio State University

E-mail: panda@cse.ohio-state.edu

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

# WHY COLLECTIVE COMMUNICATION MATTERS FOR HPC AND DL?

Percentage of Communication Time Spent in Collective Operations (%)

| Application | Value | Collectives |
|---|---|---|
| MPI-FFT | 99 | Alltoall, Bcast |
| VASP | 58 | Allreduce, Bcast, AlltoAll |
| AMG | 21 | Allreduce |
| COSMO | 31 | Allreduce, Bcast |
| Graph500 | 95 | Alltoall, Allreduce, Alltoall |
| MiniFE | 63 | Allreduce |
| MILC | 23 | Allreduce |
| DL-POLY | 27 | Allreduce |
| HOOMD-blue | 65 | Allreduce, Bcast |

- Based on HPC Advisory Council (HPCAC) MPI application profiles

- Most application profiles show majority of time spent in collective operations

- Deep Learning applications are also sensitive to performance of collectives (all-reduce and broadcast)

- Optimizing the performance of collective communication is critical for the overall performance of HPC and DL applications
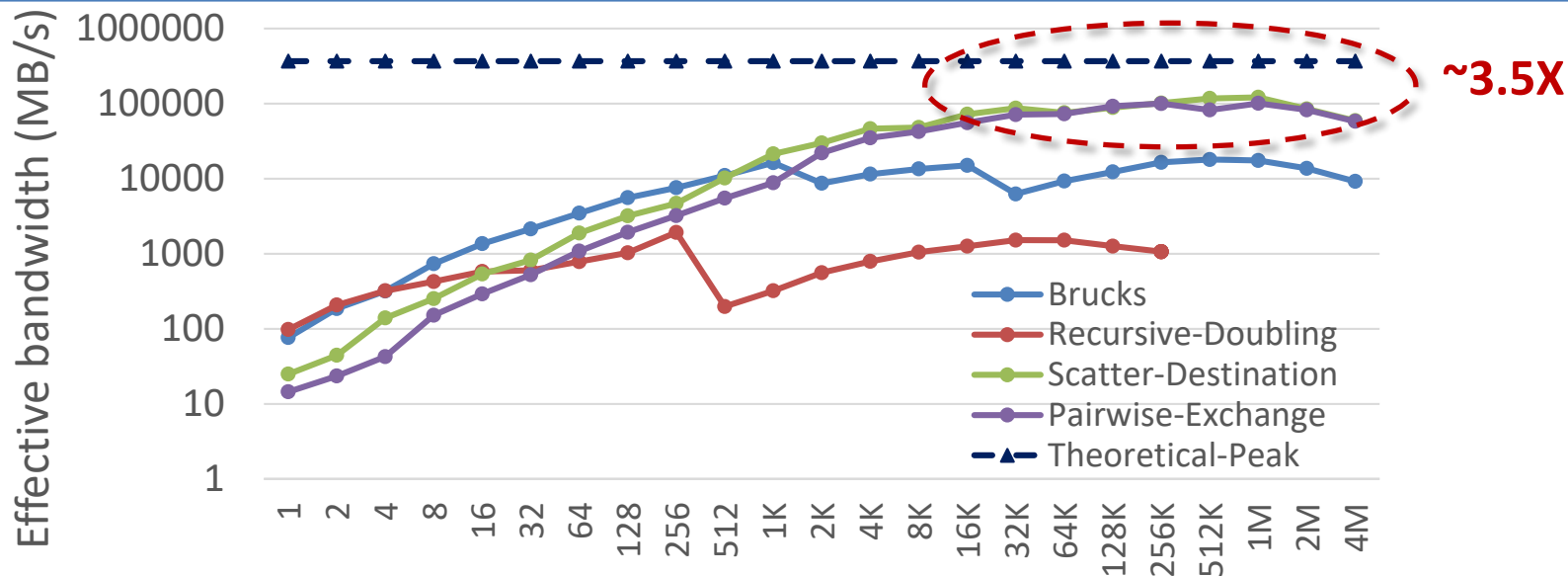
Alltoall Algorithms on single KNL 7250 in Cache-mode using MVAPICH2-2.3rc1

*Why different algorithms of even a dense collective such as Alltoall do not achieve theoretical peak bandwidth offered by the system?*

# 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 2,950 organizations in 86 countries**
  - **More than 522,000 (> 0.5 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '18 ranking)
    - 3rd ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 14th, 556,104 cores (Oakforest-PACS) in Japan
    - 17th, 367,024 cores (Stampede2) at TACC
    - 27th, 241,108-core (Pleiades) at NASA and many others
  - http://mvapich.cse.ohio-state.edu

**Partner in the upcoming TACC Frontera System**

# AGENDA

- Multiple directions being worked out by the MVAPICH2 project

  1. Contention-aware, kernel-assisted designs for large-message intra-node collectives (CMA collectives)
  2. Integrated collective designs with SHARP
  3. Designs for scalable reduction operations while exploiting multi-core
  4. Shared-address space (XPMEM)-based scalable collectives

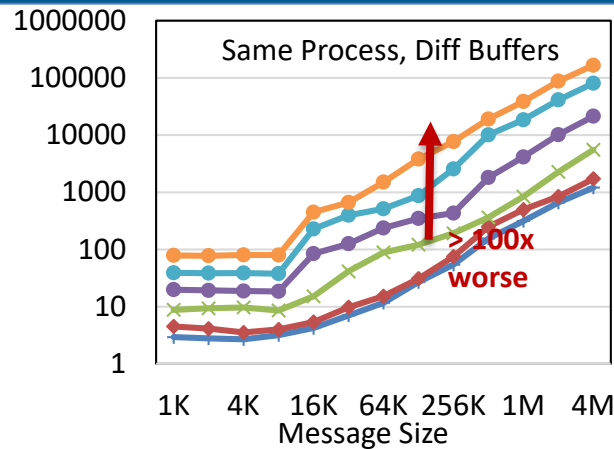- Solutions have been integrated into the MVAPICH2 libraries and publicly available
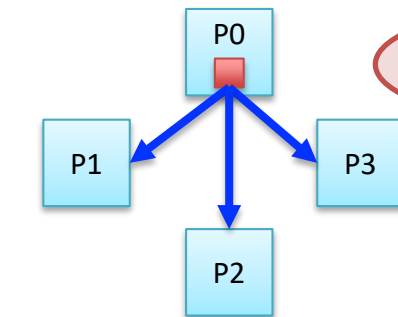
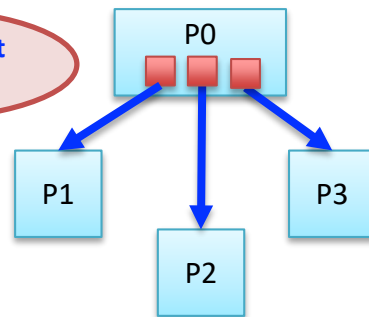# IMPACT OF COLLECTIVE COMMUNICATION PATTERN ON CMA COLLECTIVES
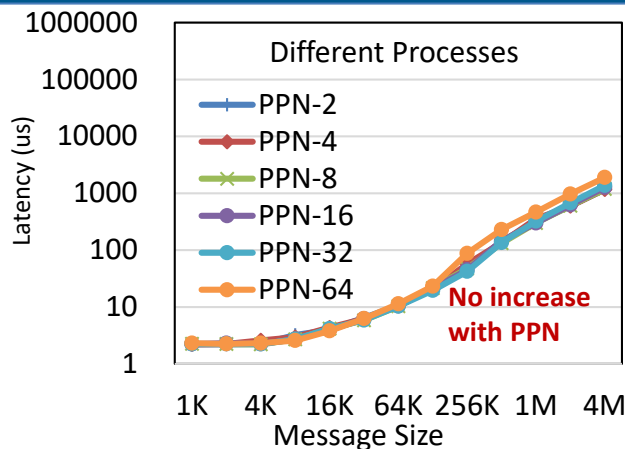


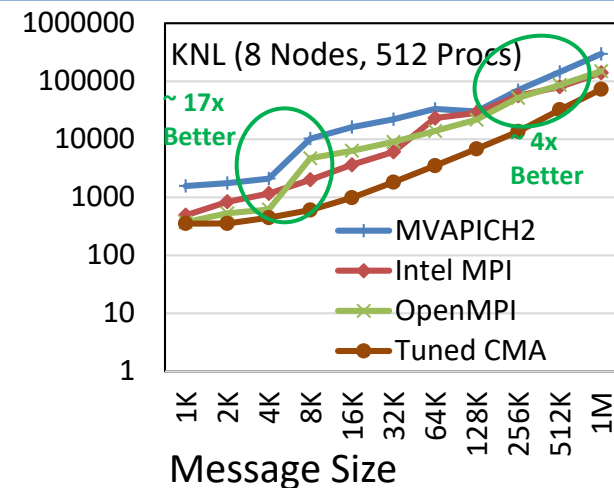**All-to-All – Good Scalability**
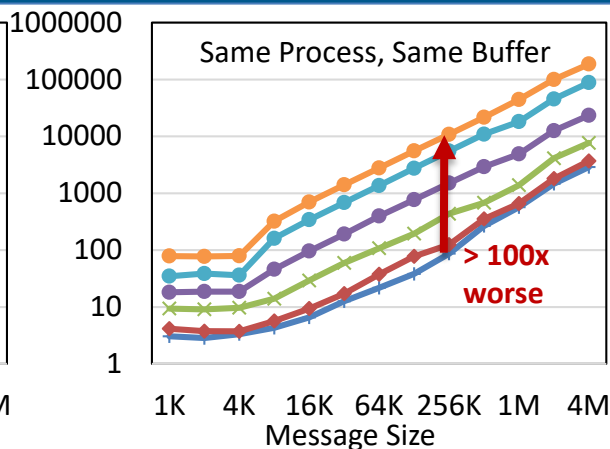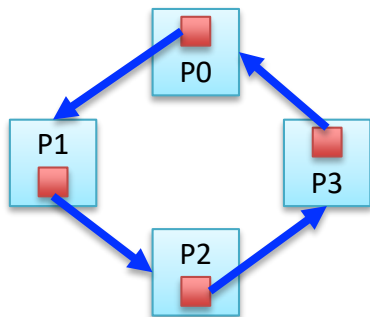
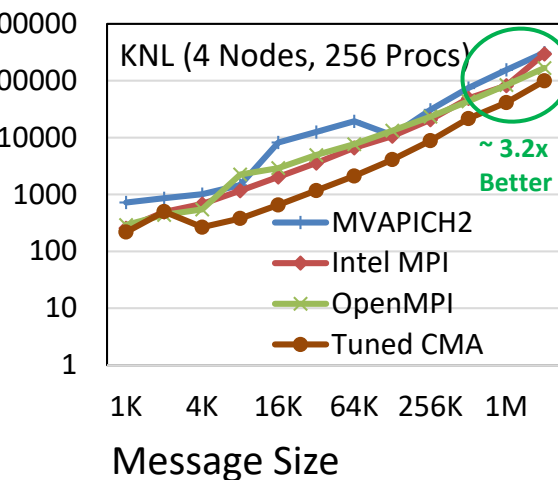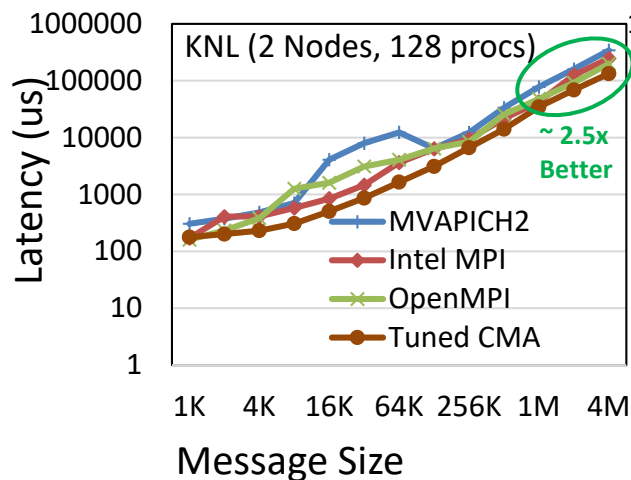**One-to-All - Poor Scalability**

**One-to-All – Poor Scalability**

Contention is at Process level

# MULTI-NODE SCALABILITY USING TWO-LEVEL ALGORITHMS



- Significantly faster intra-node communication
- New two-level collective designs can be composed
- 4x-17x improvement in 8 node Scatter and Gather compared to default MVAPICH2

*S. Chakraborty, H. Subramoni, and D. K. Panda,* Contention Aware Kernel-Assisted MPI Collectives for Multi/Many-core Systems, *IEEE Cluster '17*, *BEST Paper Finalist*
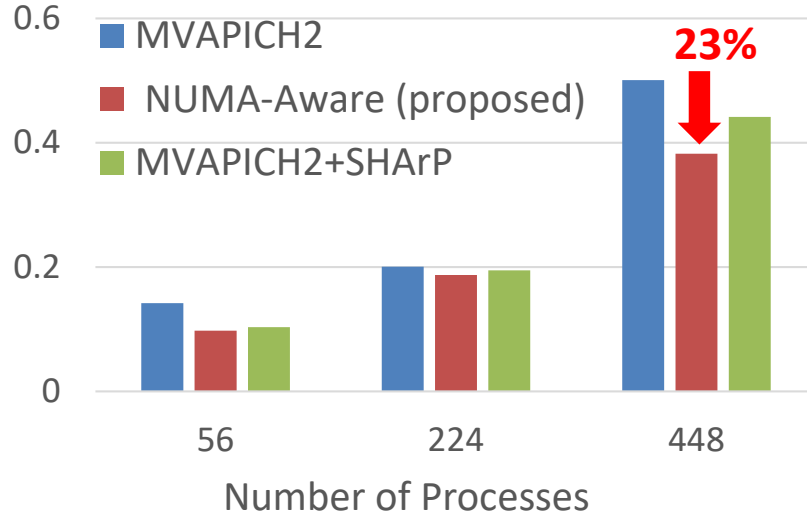
**Available since MVAPICH2-X 2.3b**

# PERFORMANCE OF NUMA-AWARE SHARP DESIGN ON XEON + IB CLUSTER
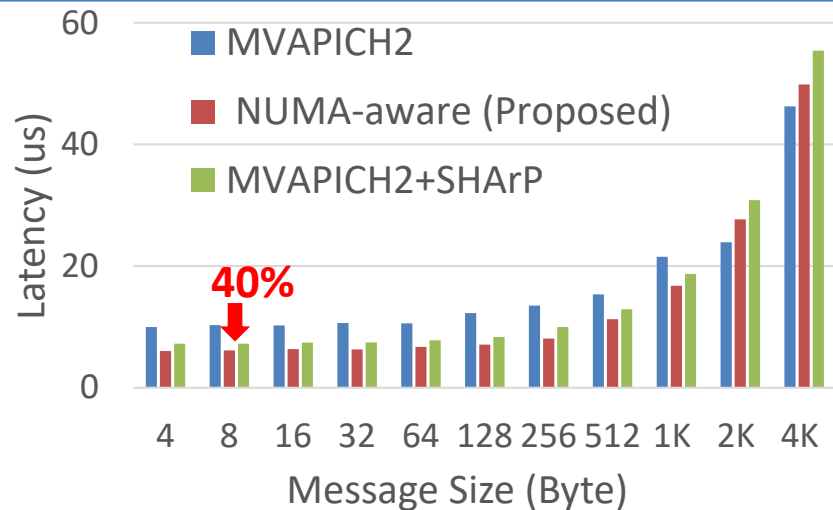


**OSU Micro Benchmark (16 Nodes, 28 PPN)**

**HPCG  (16 nodes, 28 PPN)**

- As the message size decreases, the benefits of using Socket-based design increases

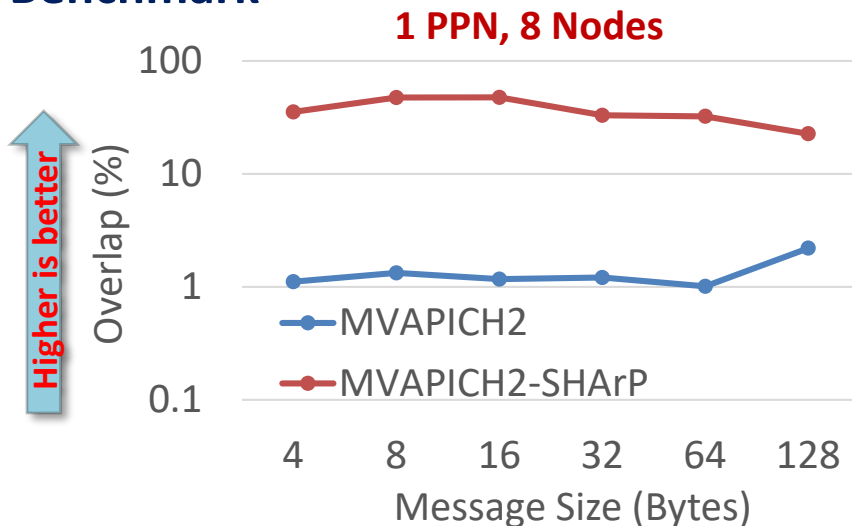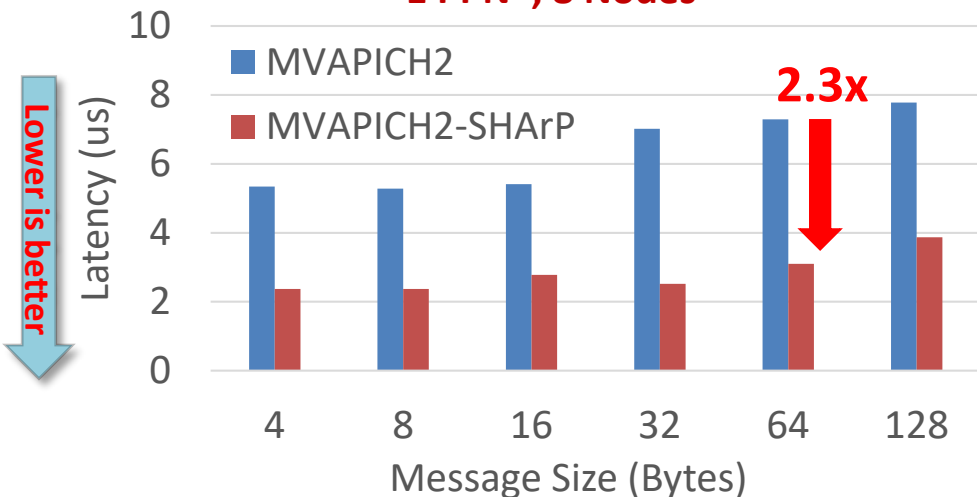- NUMA-aware design can reduce the latency by up to 23% for DDOT phase of HPCG

M. Bayatpour, S. Chakraborty, H. Subramoni, X. Lu, and D. K. Panda, Scalable Reduction Collectives with Data Partitioning-based Multi-Leader Design, SuperComputing '17.

**Available since MVAPICH2-X 2.3b**

# SHARP BASED NON-BLOCKING ALLREDUCE IN MVAPICH2

**MPI_Iallreduce Benchmark**



- Complete offload of Allreduce collective operation to *"Switch"*
  - Higher overlap of communication and computation

*\*PPN: Processes Per Node*

*Available since MVAPICH2 2.3a*

**MVAPICH2 Collectives - OFAWS 2019**

# MVAPICH2-GDR VS. NCCL2 – ALLREDUCE ON DGX-2 (PRELIMINARY RESULTS)

- **Optimized designs in upcoming MVAPICH2-GDR offer better/comparable performance for most cases**

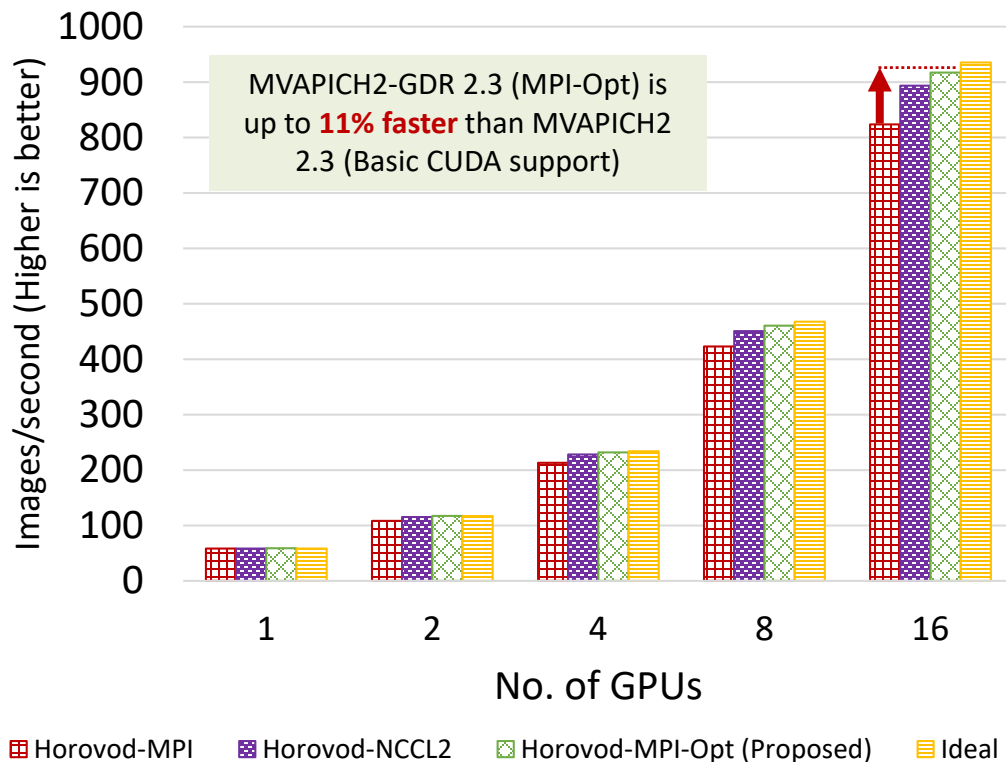- **MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) on 1 DGX-2 node (16 Volta GPUs)**
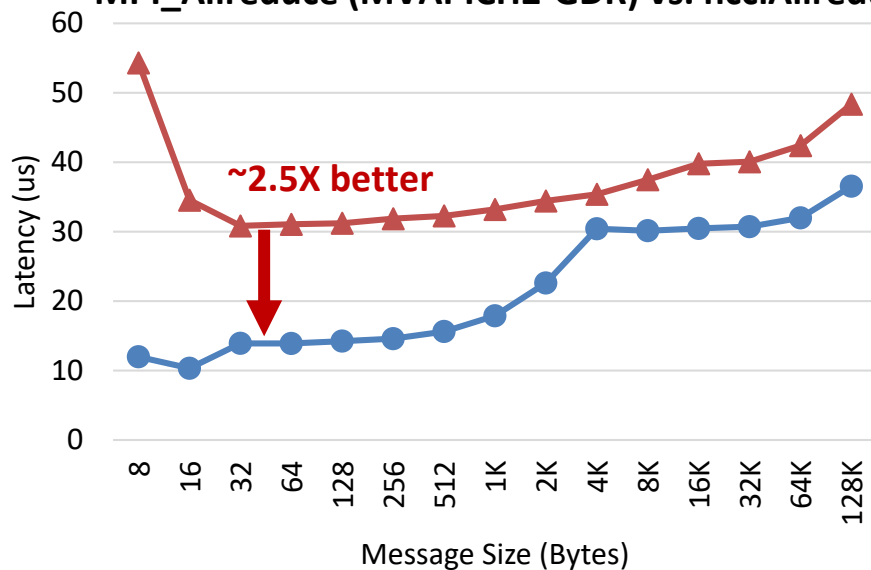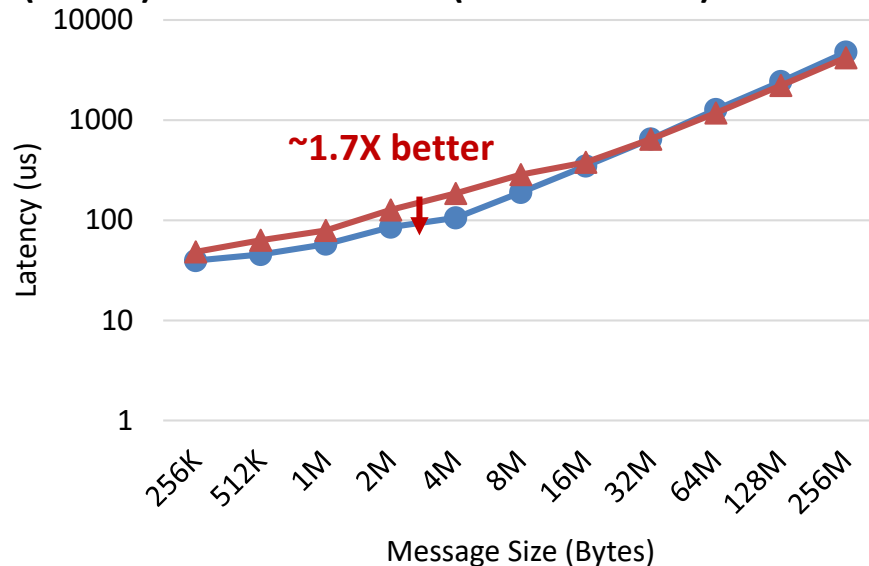


**Platform: Nvidia DGX-2 system (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2**

MVAPICH2 Collectives - OFAWS 2019

- MVAPICH2-GDR offers excellent performance via advanced designs for MPI_Allreduce.

- Up to 11% better performance on the RI2 cluster (16 GPUs)

- Near-ideal – 98% scaling efficiency

A. A. Awan et al., "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation", Under Review, https://arxiv.org/abs/1810.11112

MVAPICH2-GDR 2.3 (MPI-Opt) is up to **11% faster** than MVAPICH2 2.3 (Basic CUDA support)

Images/second (Higher is better)

No. of GPUs

⊞ Horovod-MPI    ⊞ Horovod-NCCL2    ⊞ Horovod-MPI-Opt (Proposed)    ☐ Ideal
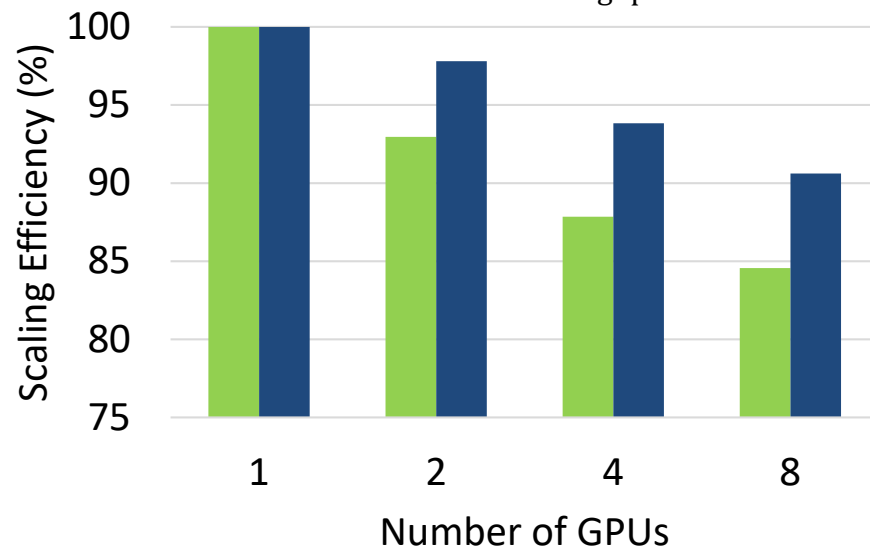
# MVAPICH2-GDR VS. NCCL2 – RESNET-50 TRAINING

- **ResNet-50 Training using TensorFlow benchmark on 1 DGX-2 node (8 Volta GPUs)**
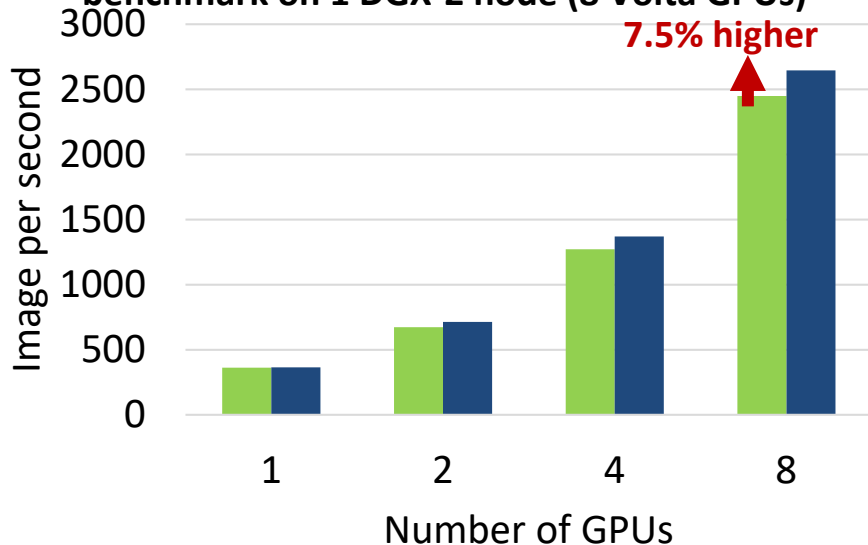


$$\text{Scaling Efficiency} = \frac{\text{Actual throughput}}{\text{Ideal throughput at scale}} \times 100\%$$

*Platform: Nvidia DGX-2 system (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2*

# SHARED ADDRESS SPACE (XPMEM-BASED) COLLECTIVES

- **Offload Reduction computation and communication to peer MPI ranks**
  - Every Peer has direct "load/store" access to other peer's buffers
  - Multiple pseudo roots independently carry-out reductions for intra-and inter-node
  - Directly put reduced data into root's receive buffer
- *True "Zero-copy"* **design for Allreduce and Reduce**
  - No copies require during the entire duration of Reduction operation
  - Scalable to multiple nodes
- *Zero contention* **overheads as memory copies happen in "*user-space*"**

**Available since MVAPICH2-X 2.3rc1**

*J. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, and D. Panda, Designing Efficient Shared Address Space Reduction Collectives for Multi-/Many-cores, International Parallel & Distributed Processing Symposium (IPDPS '18), May 2018.*
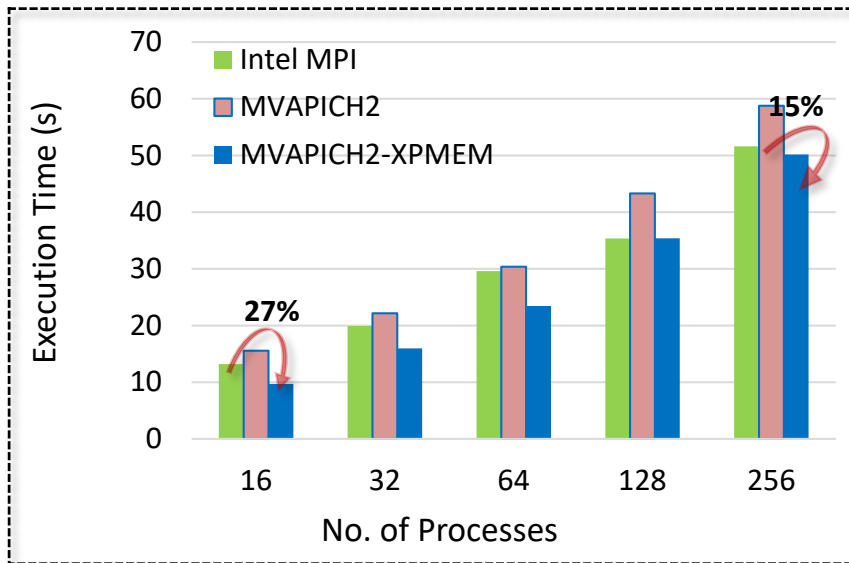
# APPLICATION-LEVEL BENEFITS OF XPMEM-BASED COLLECTIVES



CNTK AlexNet Training
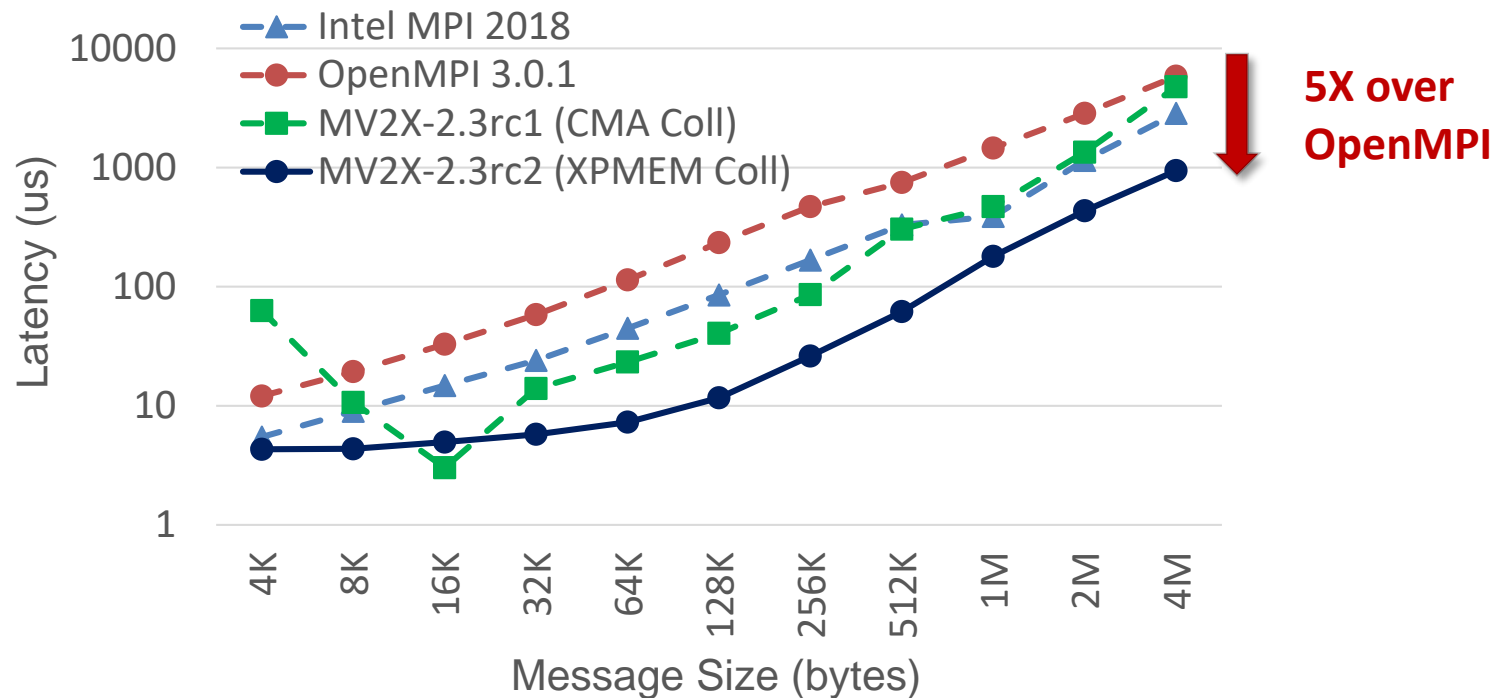(Broadwell, B.S=default, iteration=50, ppn=28)
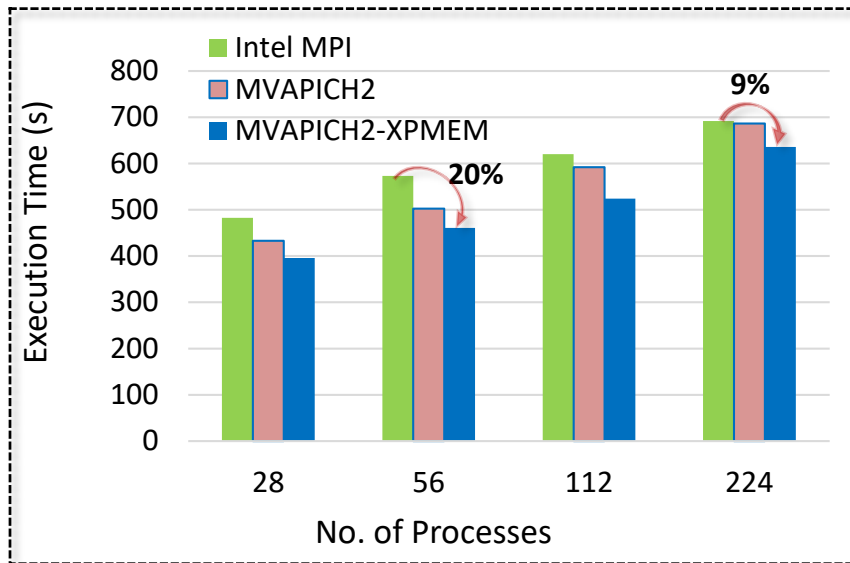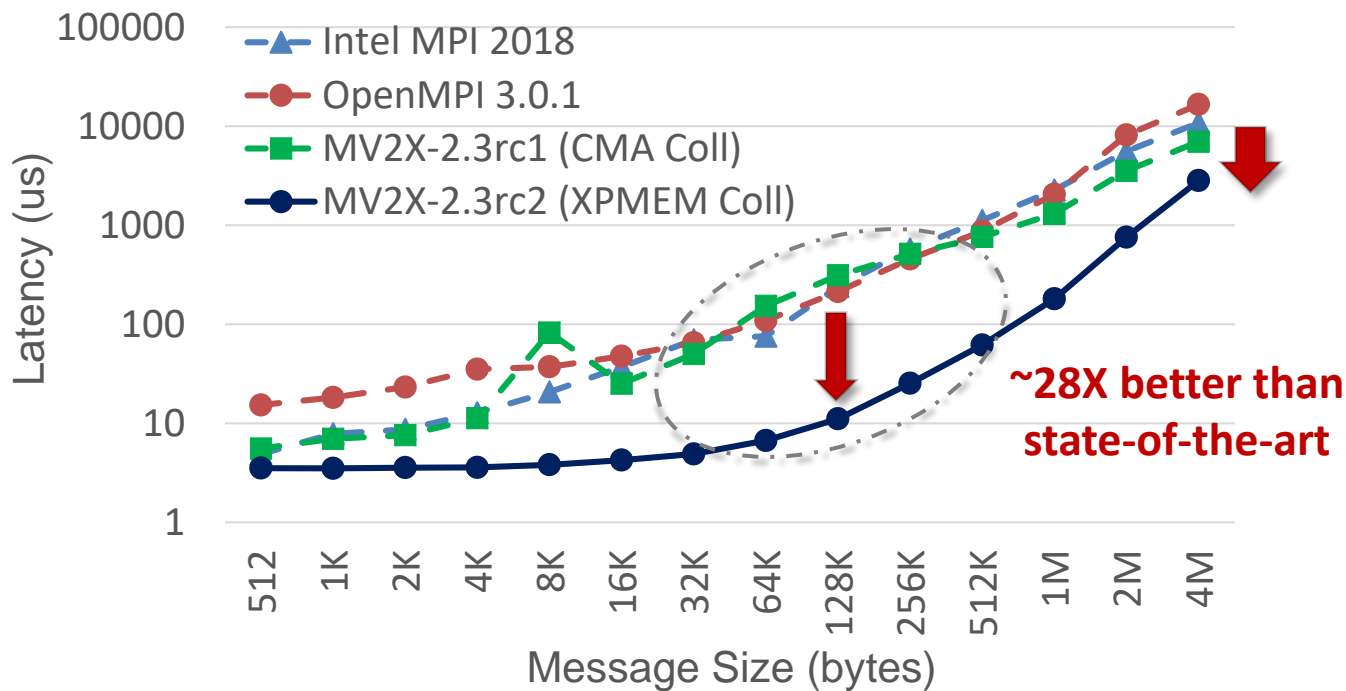
MiniAMR (Broadwell, ppn=16)

- Up to **20%** benefits over IMPI for CNTK DNN training using AllReduce
- Up to **27%** benefits over IMPI and up to **15%** improvement over MVAPICH2 for MiniAMR application kernel

MVAPICH2 Collectives - OFAWS 2019

# BENEFITS OF XPMEM BASED MPI_BCAST



Legend:
- Intel MPI 2018
- OpenMPI 3.0.1
- MV2X-2.3rc1 (CMA Coll)
- MV2X-2.3rc2 (XPMEM Coll)

5X over OpenMPI

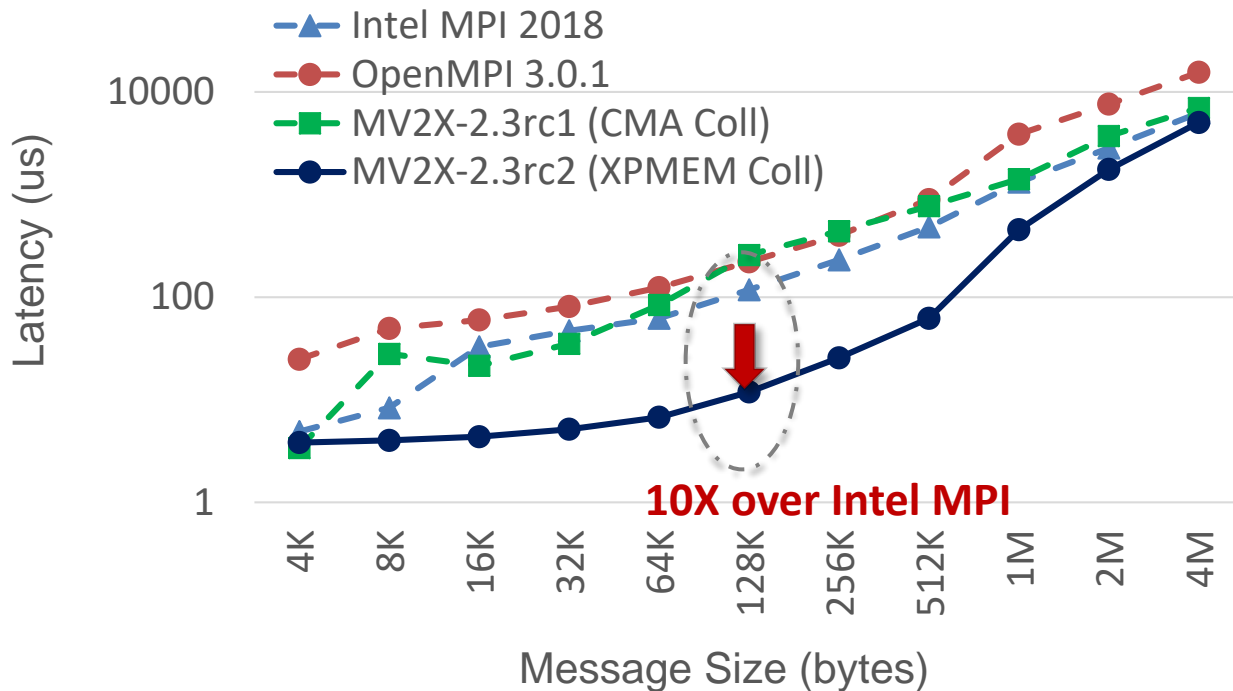Latency (us) vs Message Size (bytes): 4K, 8K, 16K, 32K, 64K, 128K, 256K, 512K, 1M, 2M, 4M

- **28 MPI Processes** on single dual-socket Broadwell E5-2680v4, 2x14 core processor

# BENEFITS OF XPMEM BASED MPI_SCATTER



Chart legend:
- Intel MPI 2018
- OpenMPI 3.0.1
- MV2X-2.3rc1 (CMA Coll)
- MV2X-2.3rc2 (XPMEM Coll)

Y-axis: Latency (us) — 1, 10, 100, 1000, 10000, 100000

X-axis: Message Size (bytes) — 512, 1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K, 256K, 512K, 1M, 2M, 4M

**~28X better than state-of-the-art**

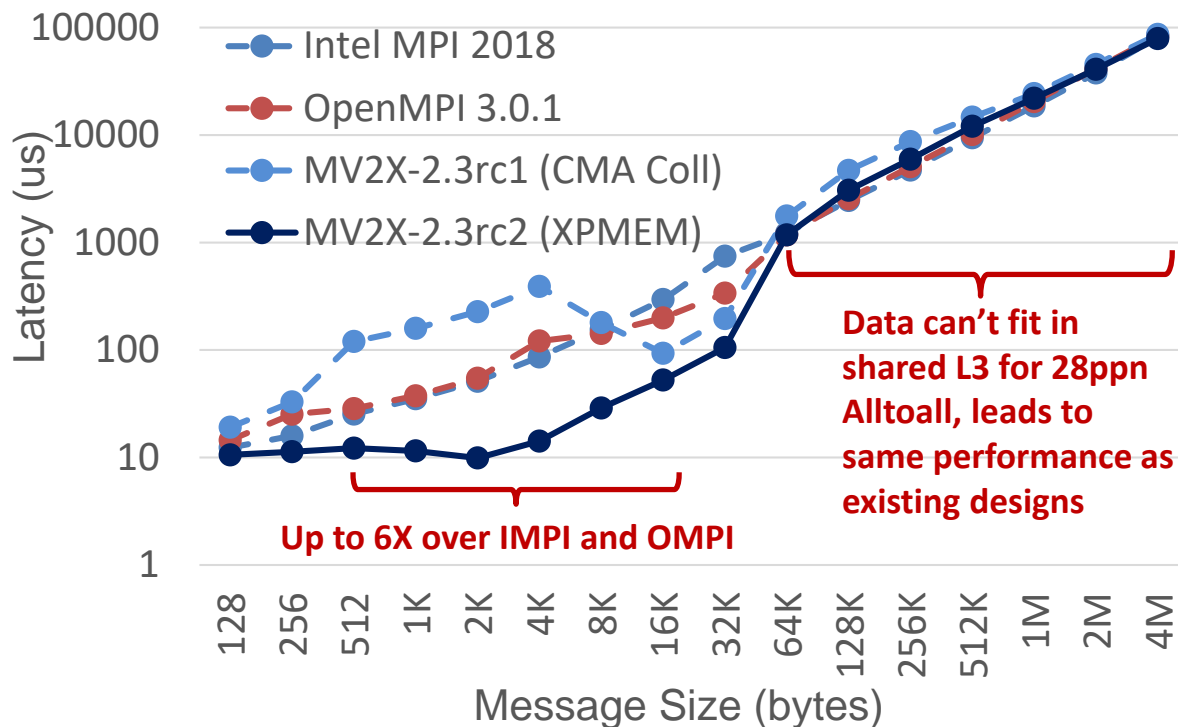- High cache-locality and **contention-free access** compared to CMA

- High cache-locality for medium messages and contention-free access

# BENEFITS OF XPMEM BASED MPI_ALLTOALL



- **28 MPI Processes** on single dual-socket Broadwell E5-2680v4, 2x14 core processor

# CONCLUDING REMARKS

- **Many-core nodes will be the foundation blocks for emerging Exascale systems**

- **Communication mechanisms and runtimes need to be <u>re-designed</u> to take advantage of the <u>high concurrency</u> offered by manycores**

- **Presented a set of <u>novel designs</u> for <u>collective communication</u> primitives in MPI that <u>address several challenges for </u>modern clusters**

- **Demonstrated the <u>performance benefits</u> of our proposed designs under a variety of <u>multi-/many-cores and high-speed networks and a range of HPC and DL applications</u>**

- **These designs are already available in MVAPICH2 libraries**

# THANK YOU!

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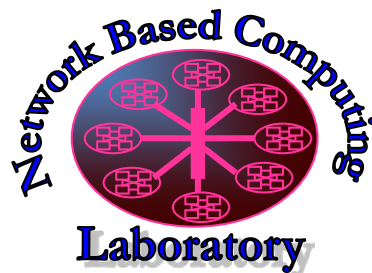


Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/





The High-Performance MPI/PGAS Project
http://mvapich.cse.ohio-state.edu/

The High-Performance Deep Learning Project
http://hidl.cse.ohio-state.edu/