MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

Exploiting Latest Networking and Accelerator Technologies for MPI, Streaming, and Deep Learning: An MVAPICH2-Based Approach

Talk at NRL booth (SC ‘17)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

Drivers of Modern HPC Cluster Architectures

Multi-core Processors

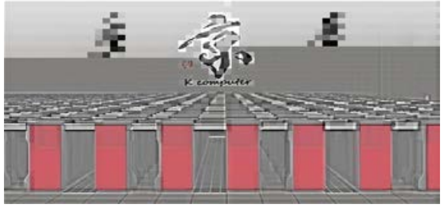
High Performance Interconnects - InfiniBand
<1usec latency, 100Gbps Bandwidth>

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
- Accelerators (NVIDIA GPGPUs and Intel Xeon Phi)
- Available on HPC Clouds, e.g., Amazon EC2, NSF Chameleon, Microsoft Azure, etc.

Sunway TaihuLight



K - Computer

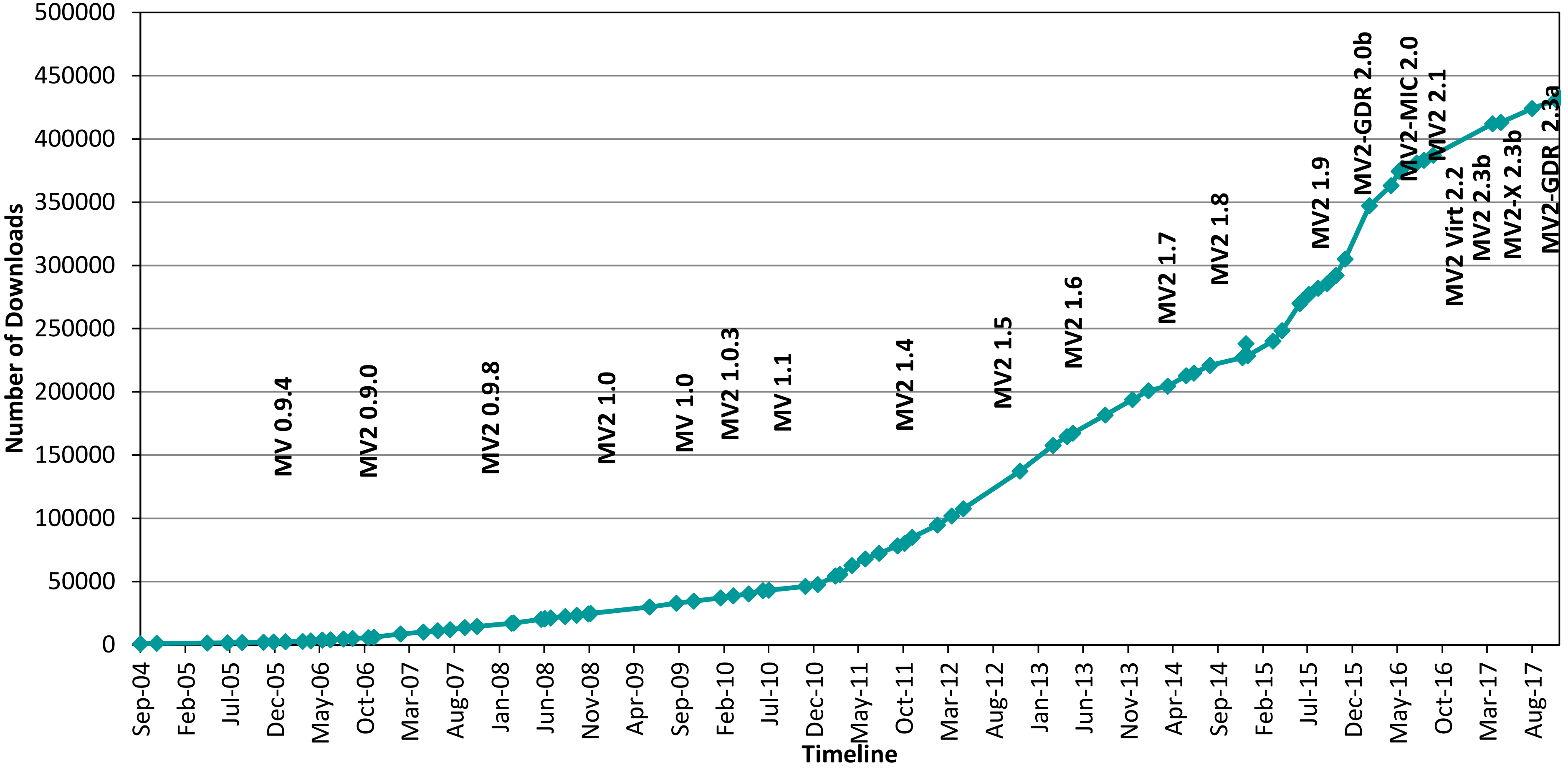
Tianhe – 2

Titan

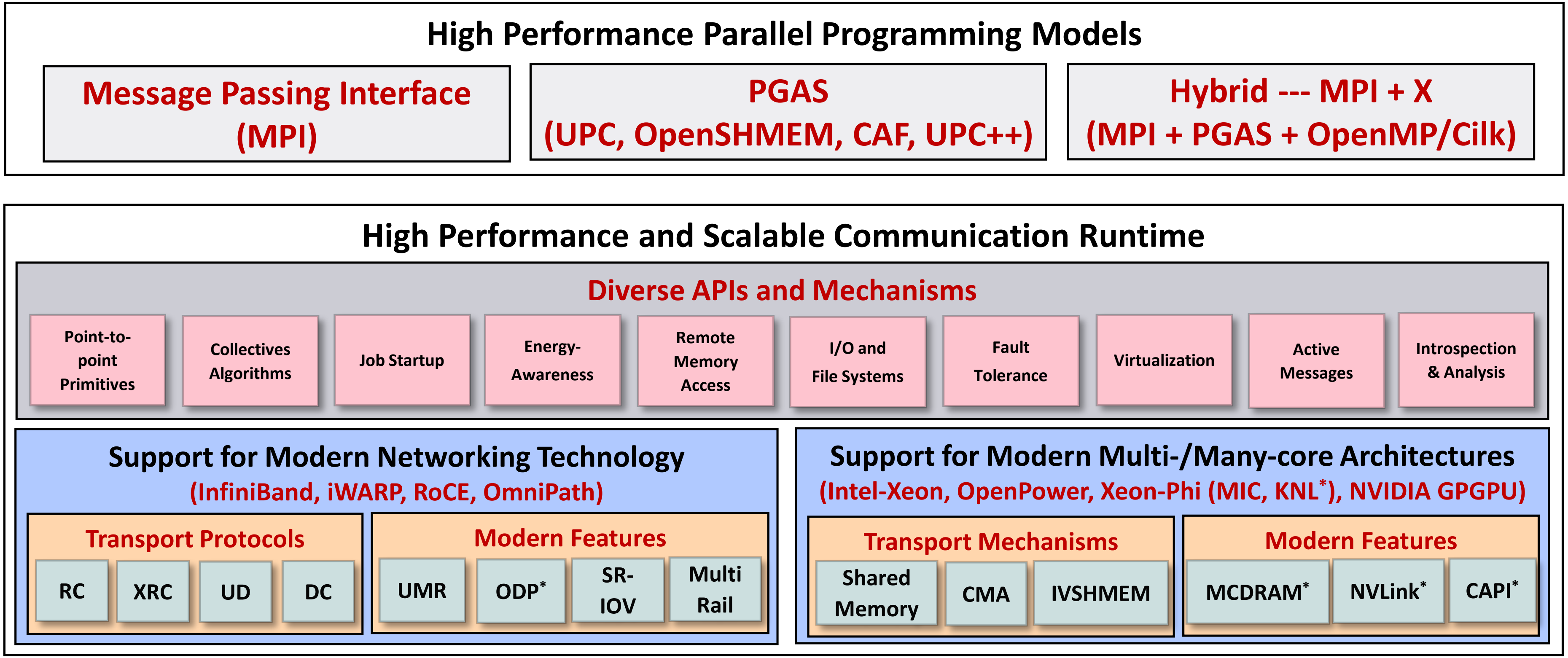
Overview of the MVAPICH2 Project

- 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 2,825 organizations in 85 countries**
  - **More than 433,000 (> 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'17, 10M cores, 100 PFlops)

MVAPICH2 Release Timeline and Downloads



MVAPICH2 Architecture



High Performance Parallel Programming Models

- Message Passing Interface (MPI)
- PGAS (UPC, OpenSHMEM, CAF, UPC++)
- Hybrid --- MPI + X (MPI + PGAS + OpenMP/Cilk)

High Performance and Scalable Communication Runtime

Diverse APIs and Mechanisms

- Point-to-point Primitives
- Collectives Algorithms
- Job Startup
- Energy-Awareness
- Remote Memory Access
- I/O and File Systems
- Fault Tolerance
- Virtualization
- Active Messages
- Introspection & Analysis

Support for Modern Networking Technology

(InfiniBand, iWARP, RoCE, OmniPath)

- Transport Protocols
  - RC
  - XRC
  - UD
  - DC
- Modern Features
  - UMR
  - ODP*
  - SR-IOV
  - Multi Rail

Support for Modern Multi-/Many-core Architectures

(Intel-Xeon, OpenPower, Xeon-Phi (MIC, KNL*), NVIDIA GPGPU)

- Transport Mechanisms
  - Shared Memory
  - CMA
  - IVSHMEM
- Modern Features
  - MCDRAM*
  - NVLink*
  - CAPI*

* Upcoming

MVAPICH2 Software Family

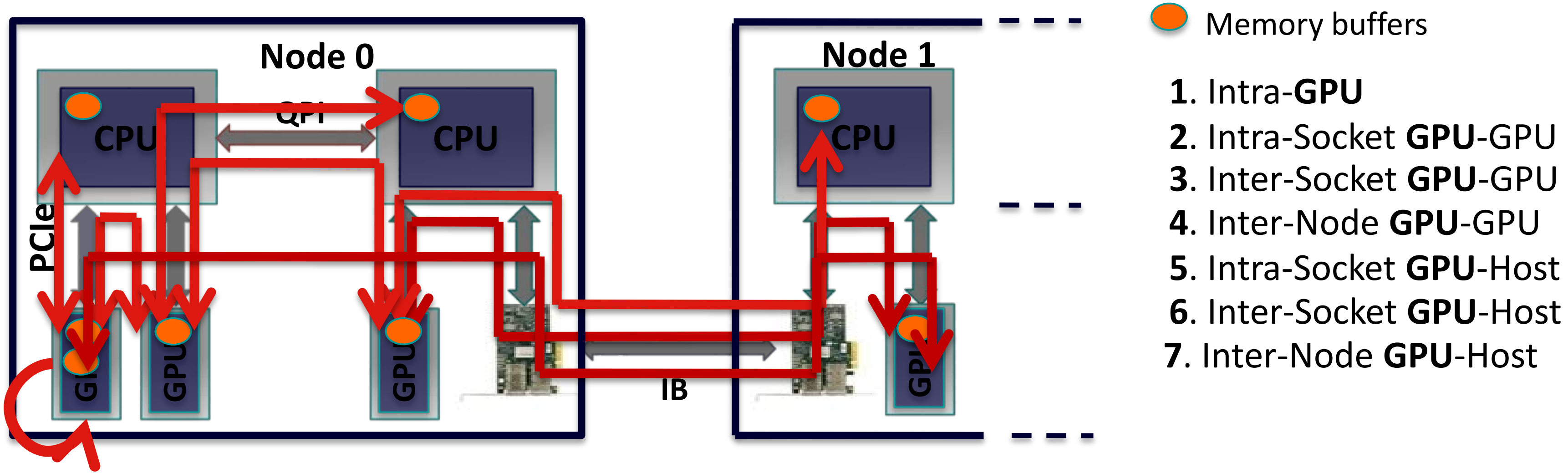
High-Performance Parallel Programming Libraries	
MVAPICH2	Support for InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE
MVAPICH2-X	Advanced MPI features, OSU INAM, PGAS (OpenSHMEM, UPC, UPC++, and CAF), and MPI+PGAS programming models with unified communication runtime
MVAPICH2-GDR	Optimized MPI for clusters with NVIDIA GPUs
MVAPICH2-Virt	High-performance and scalable MPI for hypervisor and container based HPC cloud
MVAPICH2-EA	Energy aware and High-performance MPI
MVAPICH2-MIC	Optimized MPI for clusters with Intel KNC
Microbenchmarks	
OMB	Microbenchmarks suite to evaluate MPI and PGAS (OpenSHMEM, UPC, and UPC++) libraries for CPUs and GPUs
Tools	
OSU INAM	Network monitoring, profiling, and analysis for clusters with MPI and scheduler integration
OEMT	Utility to measure the energy consumption of MPI applications

Outline

- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What's new with MVAPICH2-GDR
    - Maximal overlap in MPI Datatype Processing
    - Efficient Support for Managed Memory
    - Support for OpenPower and NVLink
    - Initial support for GPUDirect Async feature
- Streaming Support with IB Multicast and GDR
- High-Performance Deep Learning with MVAPICH2-GDR
- Conclusions

Optimizing MPI Data Movement on GPU Clusters

- Connected as PCIe devices – Flexibility but Complexity



1. Intra-**GPU**
2. Intra-Socket **GPU**-GPU
3. Inter-Socket **GPU**-GPU
4. Inter-Node **GPU**-GPU
5. Intra-Socket **GPU**-Host
6. Inter-Socket **GPU**-Host
7. Inter-Node **GPU**-Host

8. Inter-Node **GPU**-GPU with IB adapter on remote socket

and more . . .

- For each path different schemes: Shared_mem, IPC, GPUDirect RDMA, pipeline …
- Critical for runtimes to optimize data movement while hiding the complexity

GPU-Aware (CUDA-Aware) MPI Library: MVAPICH2-GPU

- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

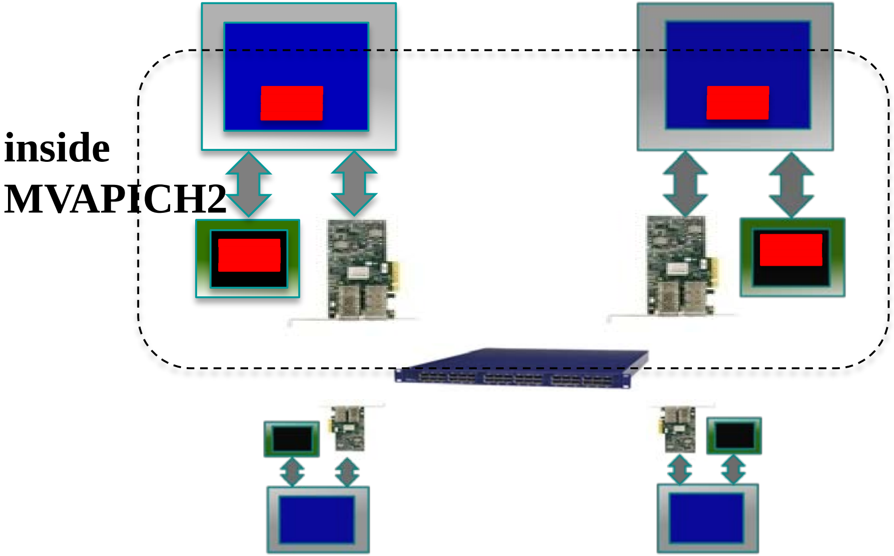
At Sender:

```
MPI_Send(s_devbuf, size, …);
```

At Receiver:

```
MPI_Recv(r_devbuf, size, …);
```

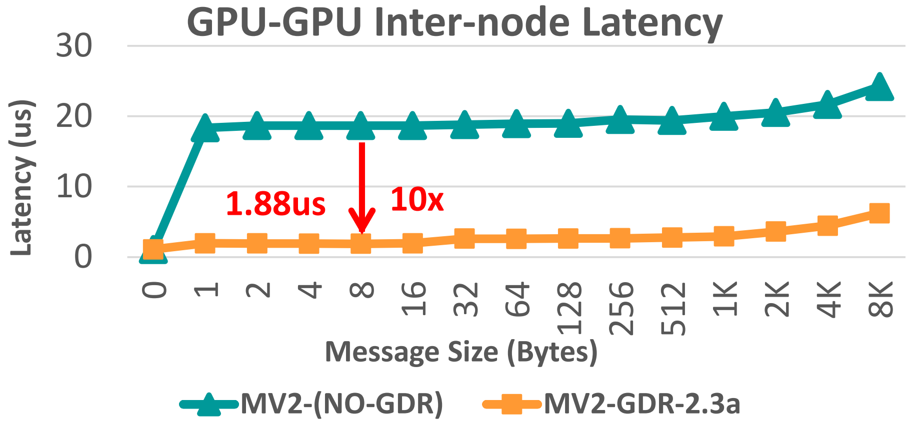
High Performance and High Productivity

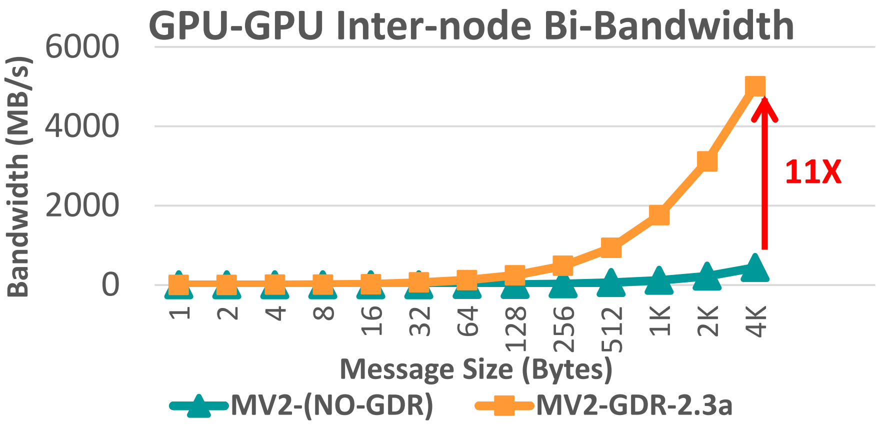


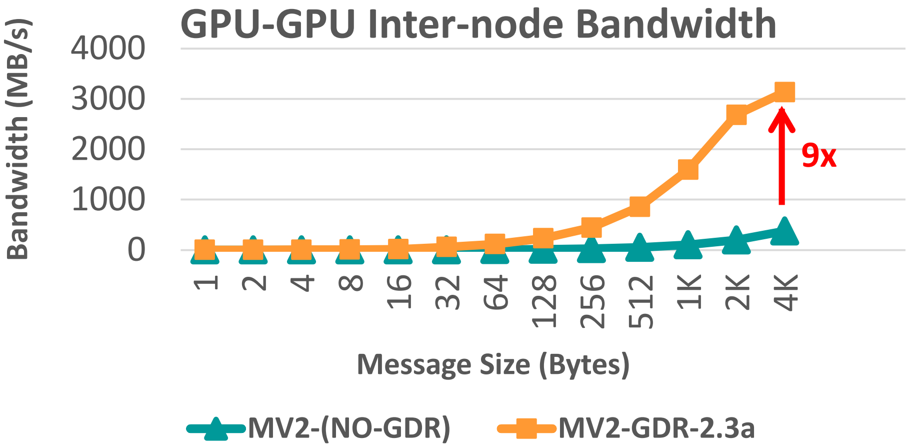
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3 Releases

- Support for MPI communication from NVIDIA GPU device memory
- High performance RDMA-based inter-node point-to-point communication (GPU-GPU, GPU-Host and Host-GPU)
- High performance intra-node point-to-point communication for multi-GPU adapters/node (GPU-GPU, GPU-Host and Host-GPU)
- Taking advantage of CUDA IPC (available since CUDA 4.1) in intra-node communication for multiple GPU adapters/node
- Optimized and tuned collectives for GPU device buffers
- MPI datatype support for point-to-point and collective communication from GPU device buffers

Optimized MVAPICH2-GDR Design

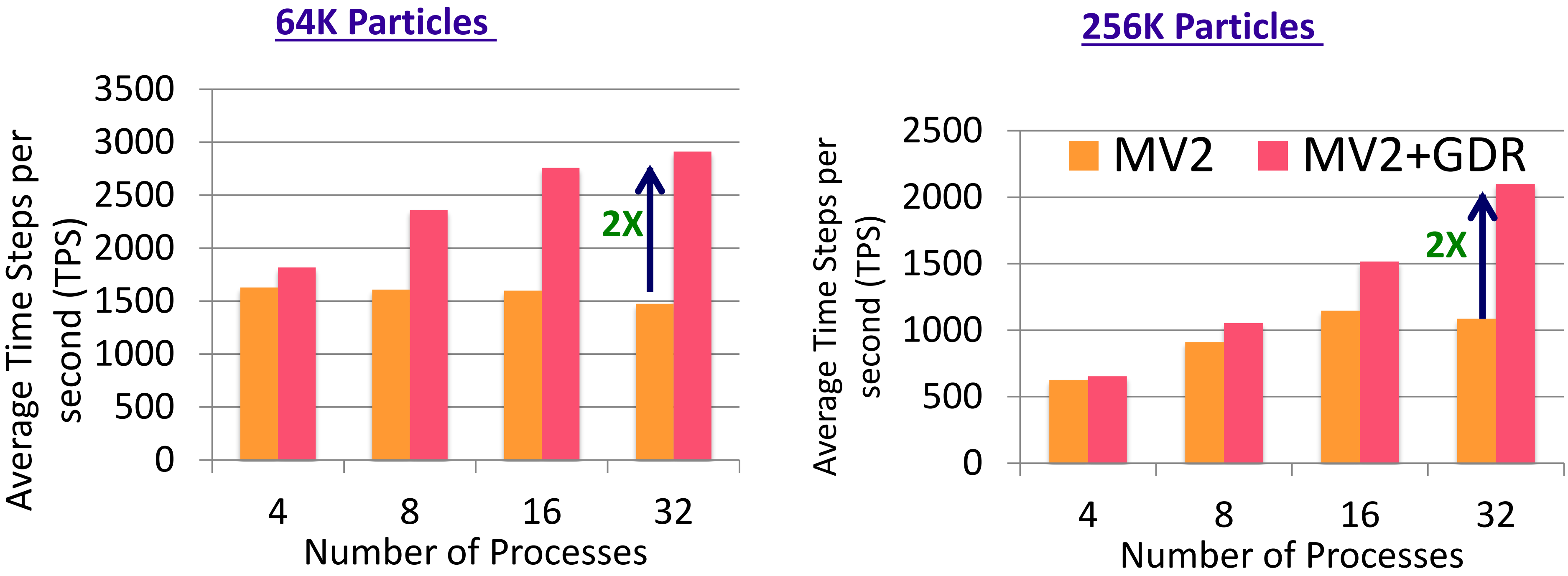






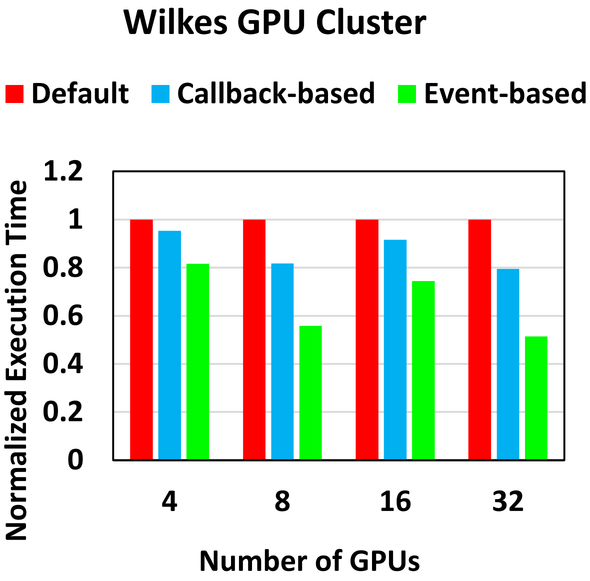
MVAPICH2-GDR-2.3a
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

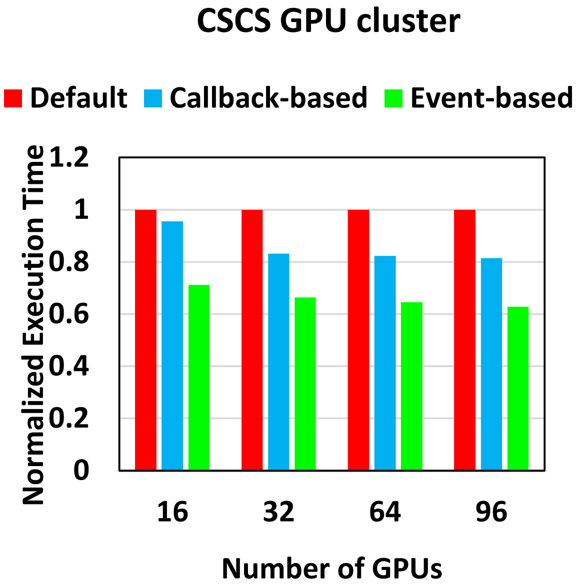
Application-Level Evaluation (HOOMD-blue)

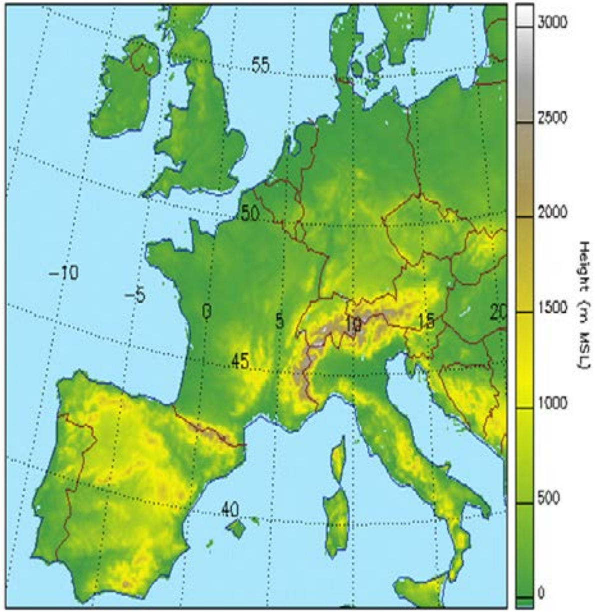


- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomdBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

Application-Level Evaluation (Cosmo) and Weather Forecasting in Switzerland







Cosmo model: http://www2.cosmo-model.org/content /tasks/operational/meteoSwiss/

- 2X improvement on 32 GPUs nodes
- 30% improvement on 96 GPU nodes (8 GPUs/node)

On-going collaboration with CSCS and MeteoSwiss (Switzerland) in co-designing MV2-GDR and Cosmo Application

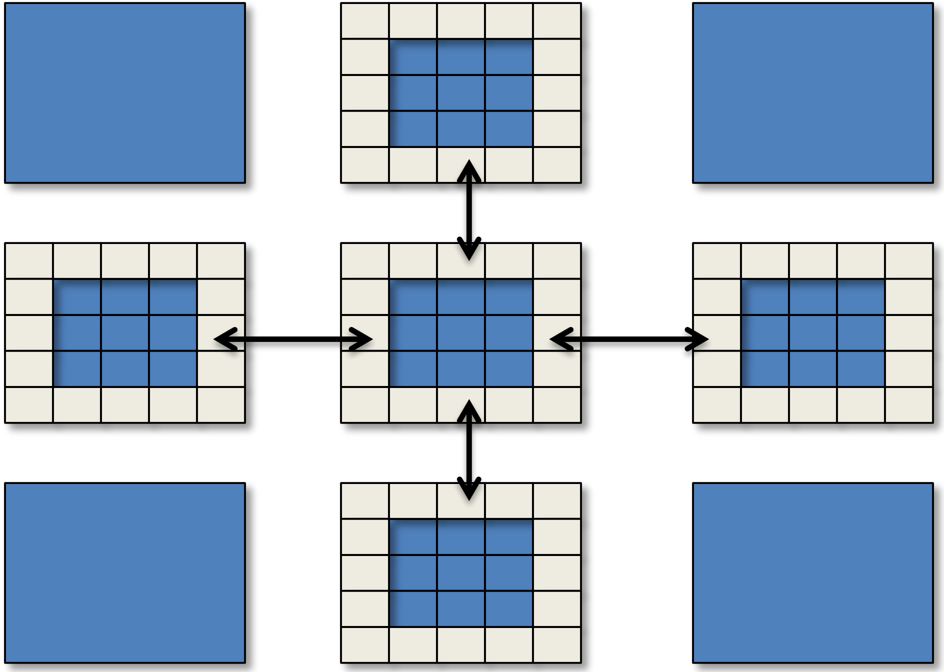
C. Chu, K. Hamidouche, A. Venkatesh, D. Banerjee , H. Subramoni, and D. K. Panda, Exploiting Maximal Overlap for Non-Contiguous Data Movement Processing on Modern GPU-enabled Systems, IPDPS'16

Outline

- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What’s new with MVAPICH2-GDR
  - Maximal overlap in MPI Datatype Processing
  - Efficient Support for Managed Memory
  - Support for OpenPower and NVLink
  - Initial support for GPUDirect Async feature
- Streaming Support with IB Multicast and GDR
- High-Performance Deep Learning with MVAPICH2-GDR
- Conclusions

Non-contiguous Data Exchange

Halo data exchange



- Multi-dimensional data
  - Row based organization
  - Contiguous on one dimension
  - Non-contiguous on other dimensions
- Halo data exchange
  - Duplicate the boundary
  - Exchange the boundary in each iteration

MPI Datatype Processing (Computation Optimization)

- Comprehensive support
  - Targeted kernels for regular datatypes - vector, subarray, indexed_block
  - Generic kernels for all other irregular datatypes
- Separate non-blocking stream for kernels launched by MPI library
  - Avoids stream conflicts with application kernels
- Flexible set of parameters for users to tune kernels
  - Vector
    - MV2_CUDA_KERNEL_VECTOR_TIDBLK_SIZE
    - MV2_CUDA_KERNEL_VECTOR_YSIZE
  - Subarray
    - MV2_CUDA_KERNEL_SUBARR_TIDBLK_SIZE
    - MV2_CUDA_KERNEL_SUBARR_XDIM
    - MV2_CUDA_KERNEL_SUBARR_YDIM
    - MV2_CUDA_KERNEL_SUBARR_ZDIM
  - Indexed_block
    - MV2_CUDA_KERNEL_IDXBLK_XDIM

MPI Datatype Processing (Communication Optimization)

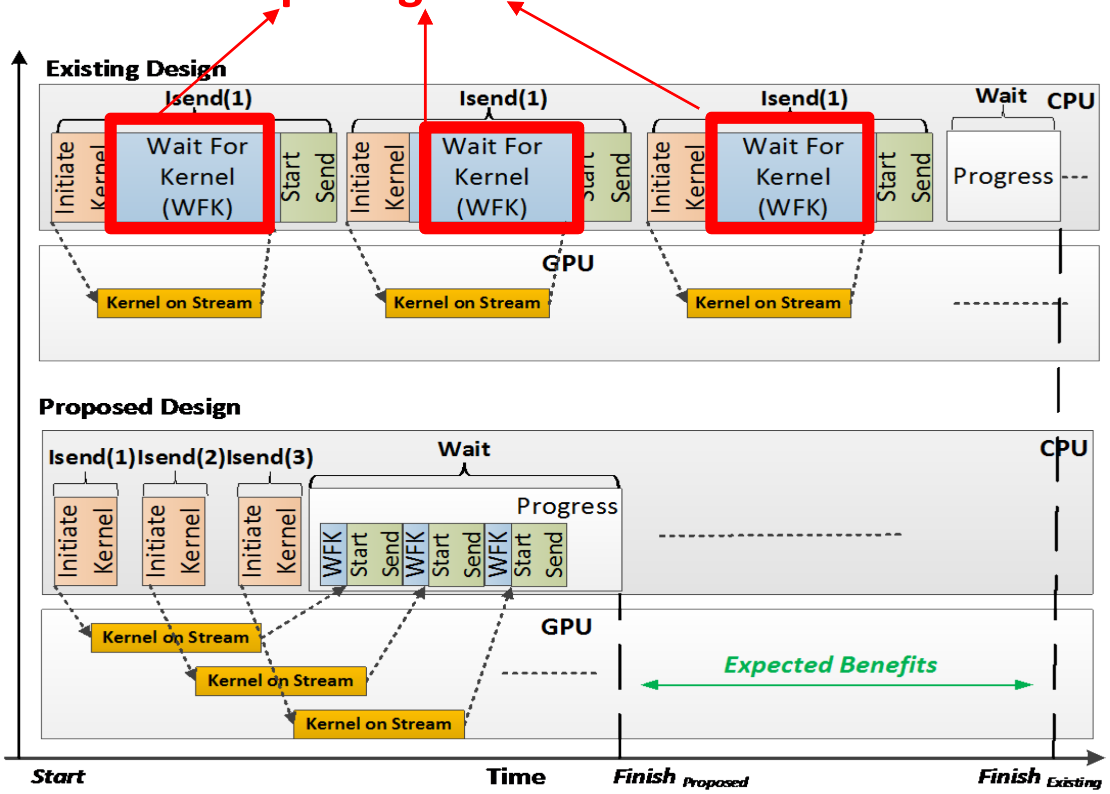
Common Scenario

MPI_Isend (A,.. Datatype,…)
MPI_Isend (B,.. Datatype,…)
MPI_Isend (C,.. Datatype,…)
MPI_Isend (D,.. Datatype,…)
…

MPI_Waitall (…);

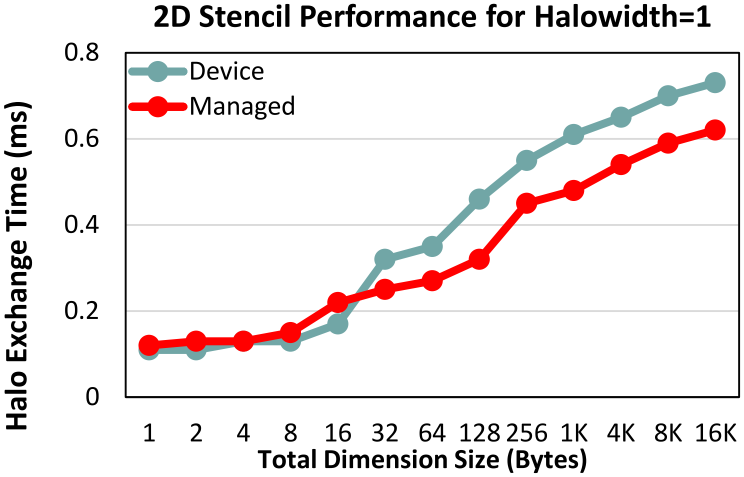
*A, B…contain non-contiguous MPI Datatype

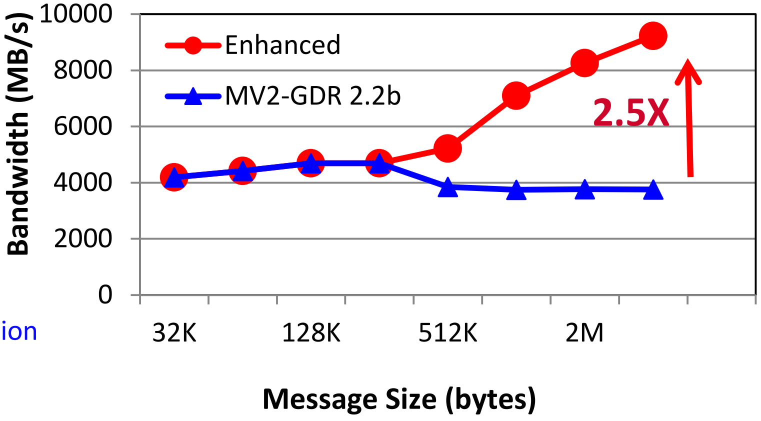
Waste of computing resources on CPU and GPU



Enhanced Support for GPU Managed Memory

- CUDA Managed => no memory pin down
  - No IPC support for intranode communication
  - No GDR support for Internode communication
- Significant productivity benefits due to abstraction of explicit allocation and *cudaMemcpy()*
- Initial and basic support in MVAPICH2-GDR
  - For both intra- and inter-nodes use "pipeline through" host memory
- Enhance intranode managed memory to use IPC
  - Double buffering pair-wise IPC-based scheme
  - Brings IPC performance to Managed memory
  - High performance and high productivity
  - 2.5 X improvement in bandwidth
- OMB extended to evaluate the performance of point-to-point and collective communications using managed buffers
- Available since MVAPICH2-GDR 2.2



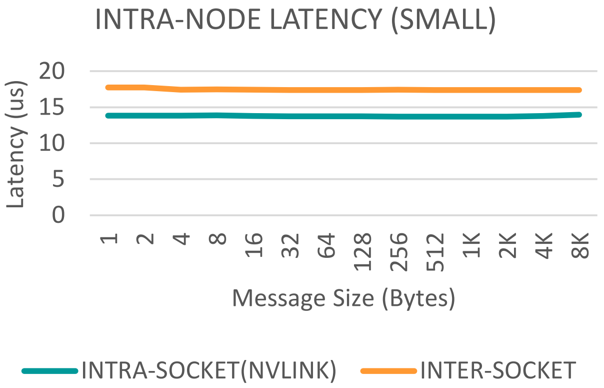


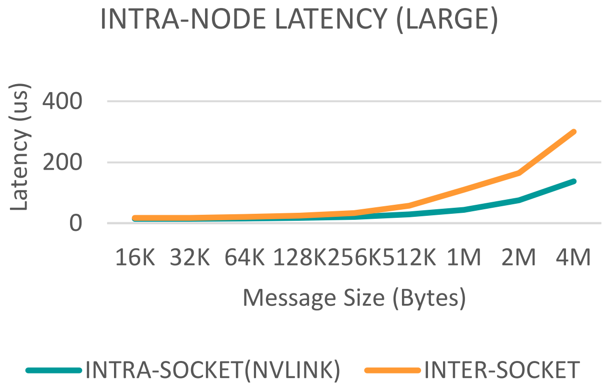
D. S. Banerjee, K Hamidouche, and D. K Panda, Designing High Performance Communication Runtime for GPUManaged Memory: Early Experiences, GPGPU-9 Workshop, held in conjunction with PPoPP '16

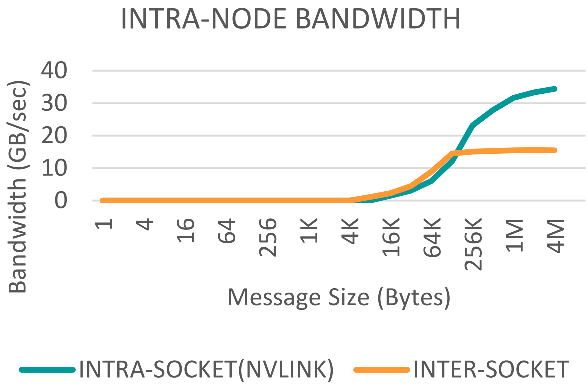
Outline

- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What’s new with MVAPICH2-GDR
    - Maximal overlap in MPI Datatype Processing
    - Efficient Support for Managed Memory
    - Support for OpenPower and NVLink
    - Initial support for GPUDirect Async feature
- Streaming Support with IB Multicast and GDR
- High-Performance Deep Learning with MVAPICH2-GDR
- Conclusions

MVAPICH2-GDR: Performance on OpenPOWER (NVLink + Pascal)

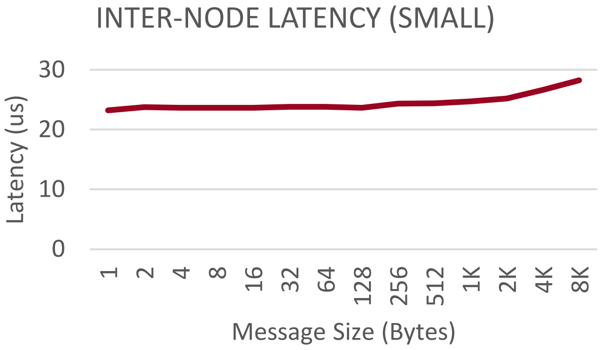

INTRA-NODE LATENCY (SMALL)

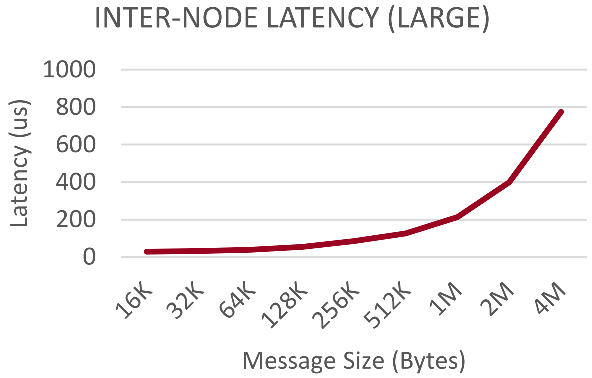

INTRA-NODE LATENCY (LARGE)

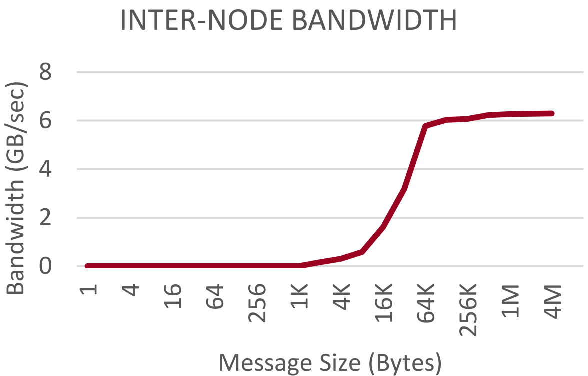

INTRA-NODE BANDWIDTH

Intra-node Latency: 13.8 us (without GPUDirectRDMA)

Intra-node Bandwidth: 33.2 GB/sec (NVLINK)


INTER-NODE LATENCY (SMALL)


INTER-NODE LATENCY (LARGE)


INTER-NODE BANDWIDTH

Inter-node Latency: 23 us (without GPUDirectRDMA)

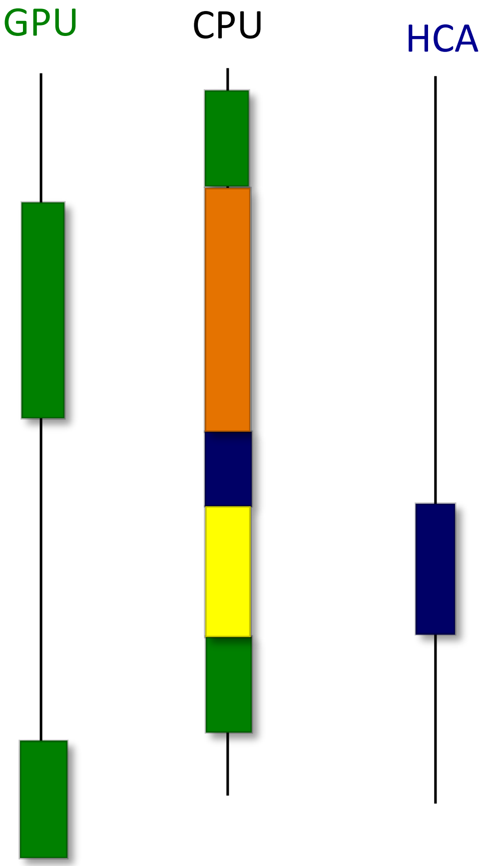
Available in MVAPICH2-GDR 2.3a

Inter-node Bandwidth: 6 GB/sec (FDR)

Platform: OpenPOWER (ppc64le) nodes equipped with a dual-socket CPU, 4 Pascal P100-SXM GPUs, and 4X-FDR InfiniBand Inter-connect

Overview of GPUDirect aSync (GDS) Feature: Current MPI+CUDA interaction

```
CUDA_Kernel_a<<<>>>(A…., stream1)
cudaStreamSynchronize(stream1)
MPI_ISend (A,…., req1)
MPI_Wait (req1)
CUDA_Kernel_b<<<>>>(B…., stream1)
```



100% CPU control

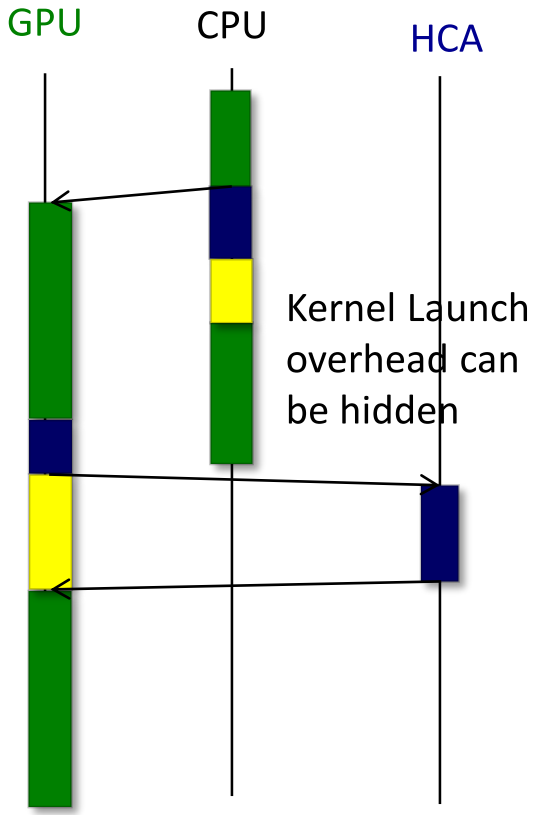
- Limits the throughput of a GPU
- Limits the asynchronous progress
- Wastes CPU cycles

MVAPICH2-GDS: Decouple GPU Control Flow from CPU

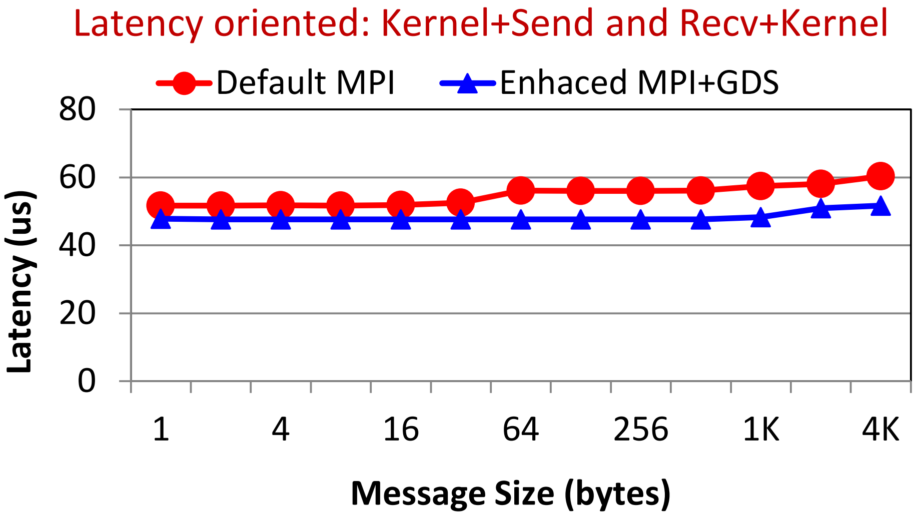
CUDA_Kernel_a<<<>>>(A…., stream1)
MPI_ISend (A,…., req1, stream1)
MPI_Wait (req1, stream1) (non-blocking from CPU)
CUDA_Kernel_b<<<>>>(B…., stream1)

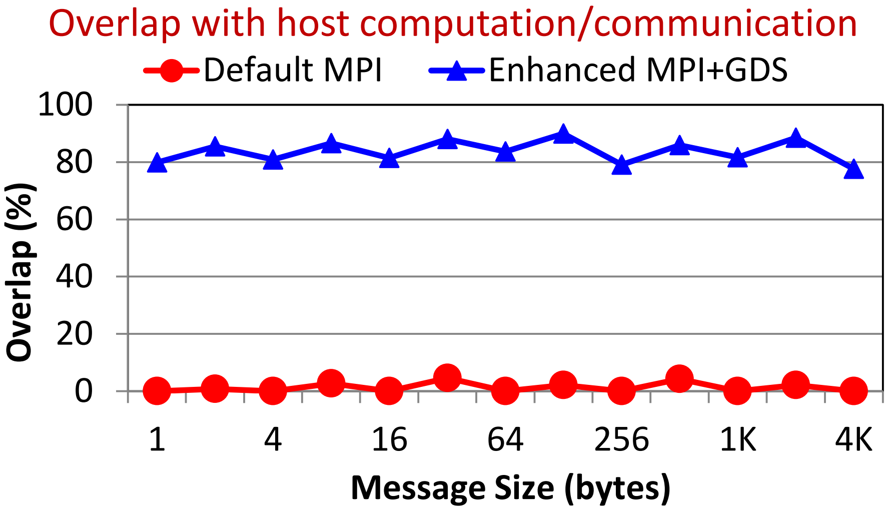
CPU offloads the compute, communication and synchronization tasks to GPU

- CPU is out of the critical path
- Tight interaction between GPU and HCA
- Hides the overhead of kernel launch
- Requires MPI semantics extensions
  - All operations are asynchronous from CPU
  - Extends MPI semantics with Stream-based semantics



MVAPICH2-GDS: Preliminary Results





- Latency Oriented: Able to hide the kernel launch overhead
  - 8-15% improvement compared to default behavior
- Overlap: Asynchronously to offload queue the Communication and computation tasks
  - 89% overlap with host computation at 128-Byte message size

Intel Sandy Bridge, NVIDIA Tesla K40c and Mellanox FDR HCA
CUDA 8.0, OFED 3.4, Each kernel is ~50us

Will be available in a public release soon

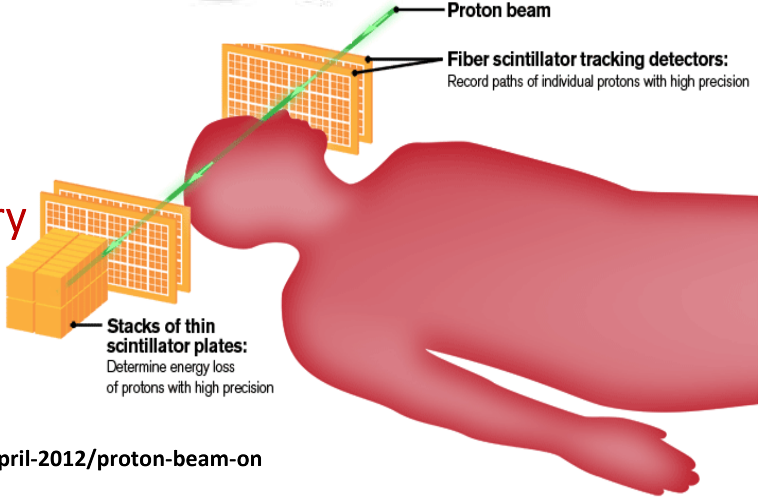
Outline

- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What’s new with MVAPICH2-GDR
  - Maximal overlap in MPI Datatype Processing
  - Efficient Support for Managed Memory
  - Support for OpenPower and NVLink
  - Initial support for GPUDirect Async feature
- Streaming Support with IB Multicast and GDR
- High-Performance Deep Learning with MVAPICH2-GDR
- Conclusions

Streaming Applications

  - Examples - surveillance, habitat monitoring, proton computed tomography (pCT), etc..
- Require efficient transport of data from/to distributed sources/sinks
- Sensitive to latency and throughput metrics

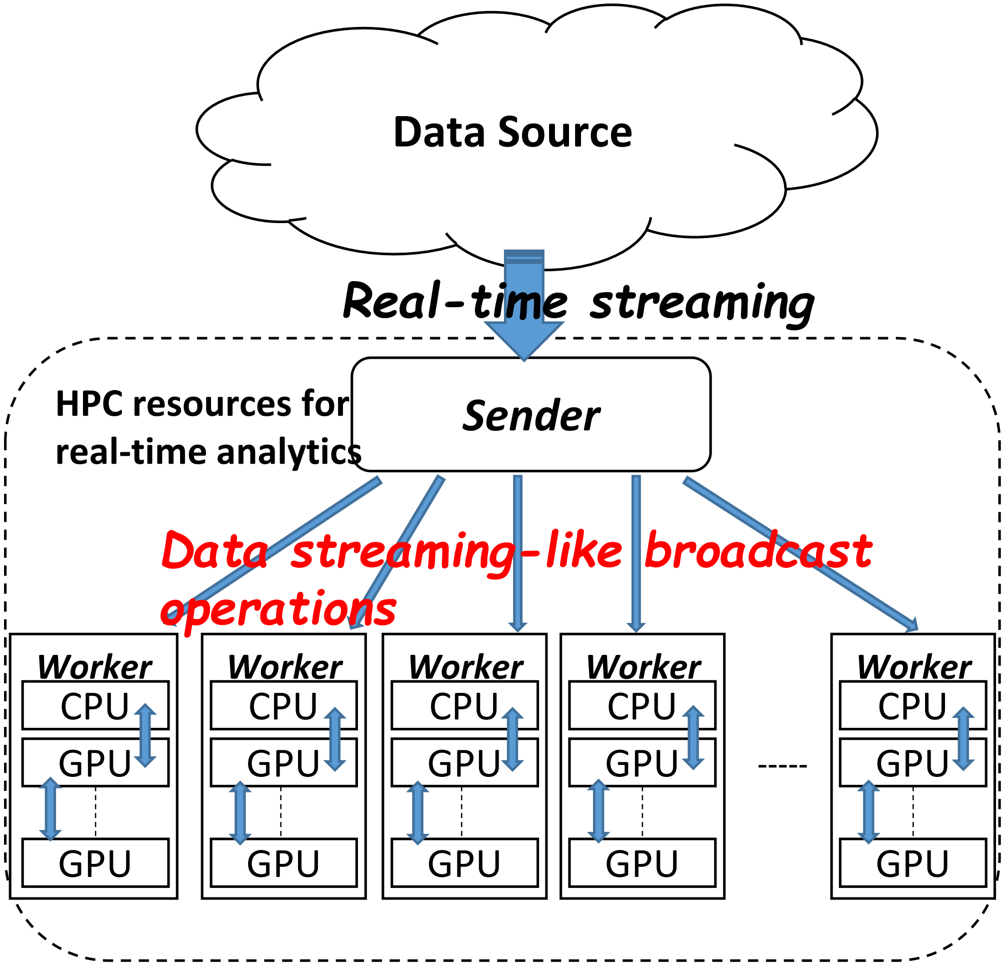
- Require HPC resources to efficiently carry out compute-intensive tasks



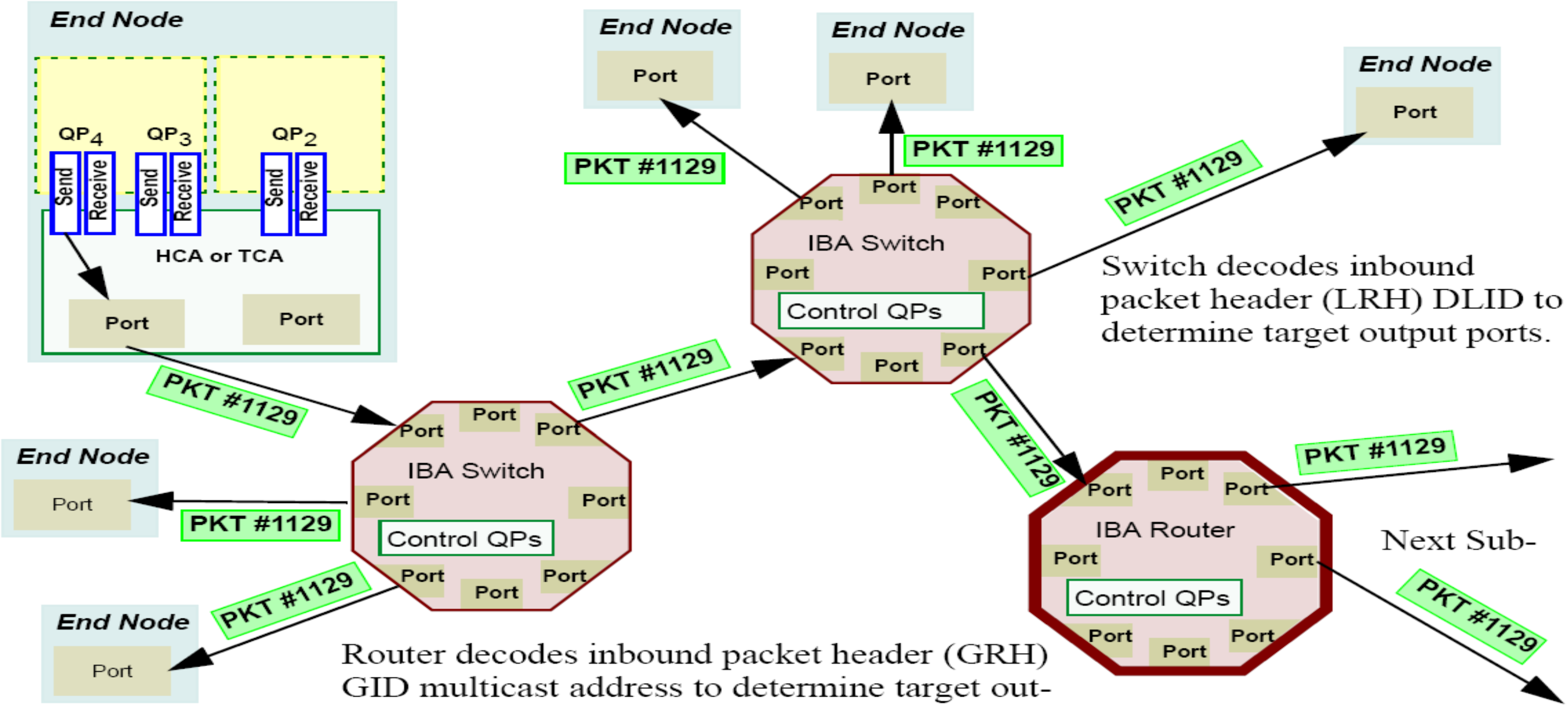
Src: http://www.symmetrymagazine.org/article/april-2012/proton-beam-on

Motivation

- **Streaming applications on HPC systems**
  1. **Communication (MPI)**
     - Broadcast-type operations
  2. **Computation (CUDA)**
     - Multiple GPU nodes as workers



IB Multicast Example



Switch decodes inbound packet header (LRH) DLID to determine target output ports.

Router decodes inbound packet header (GRH) GID multicast address to determine target out-

Problem Statement

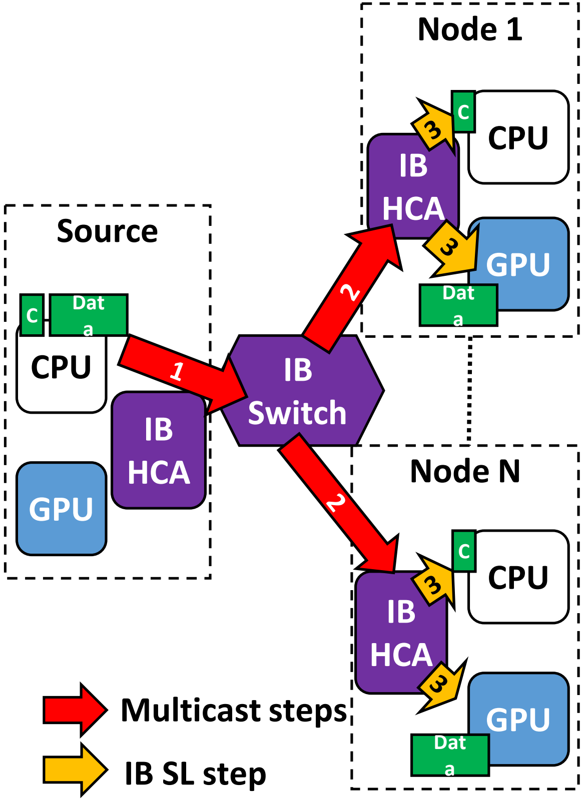
- Can we design a GPU broadcast and allreduce mechanism that can deliver low latency and high throughput for streaming applications?
- Can we combine GPUDirect RDMA (GDR) and IB-MCAST features to
  - Achieve the best performance and scalability
  - Free-up the Host-Device PCIe bandwidth for application needs
- Can such design be extended to support heterogeneous configuration (host-to-device)?
- Can we design an efficient MCAST based broadcast for multi-GPU systems?
- Can we design an efficient reliability support on top of the UD-based MCAST broadcast?
- Can we design an efficient MCAST based allreduce for GPU systems?
- How can we demonstrate such benefits at benchmark and applications level?

Related Publications

- Handling Efficient and Reliable Broadcast on Multi-GPU Clusters
  - C.-H. Chu, K. Hamidouche, H. Subramoni, A. Venkatesh, B. Elton, and D. K. Panda. “Designing High Performance Heterogeneous Broadcast for Streaming Applications on GPU Clusters, “ SBAC-PAD’16, Oct 2016.
  - C.-H. Chu, K. Hamidouche, H. Subramoni, A. Venkatesh, B. Elton, and D. K. Panda. “Efficient Reliability Support for Hardware Multicast-based Broadcast in GPU-enabled Streaming Applications,“ COMHPC 2016 (SC Workshop), Nov 2016.
- Optimizing Broadcast for GPU-based Deep Learning
  - Ching-Hsiang Chu, Xiaoyi Lu, Ammar A. Awan, Hari Subramoni, Jahanzeb Hashmi, Bracy Elton, and Dhabaleswar K. Panda, "Efficient and Scalable Multi-Source Streaming Broadcast on GPU Clusters for Deep Learning , " *ICPP’17.*
- High-Performance Broadcast with IB-MCAST and GDR
  - Ching-Hsiang Chu, Xiaoyi Lu, Ammar A. Awan, Hari Subramoni, Bracy Elton, and Dhabaleswar K. Panda., "Exploiting Hardware Multicast and GPUDirect RDMA for Efficient Broadcast , ” *submitted to IEEE TPDS.* (Under review)

SL-based Design for Heterogeneous Configuration (Host-Device)

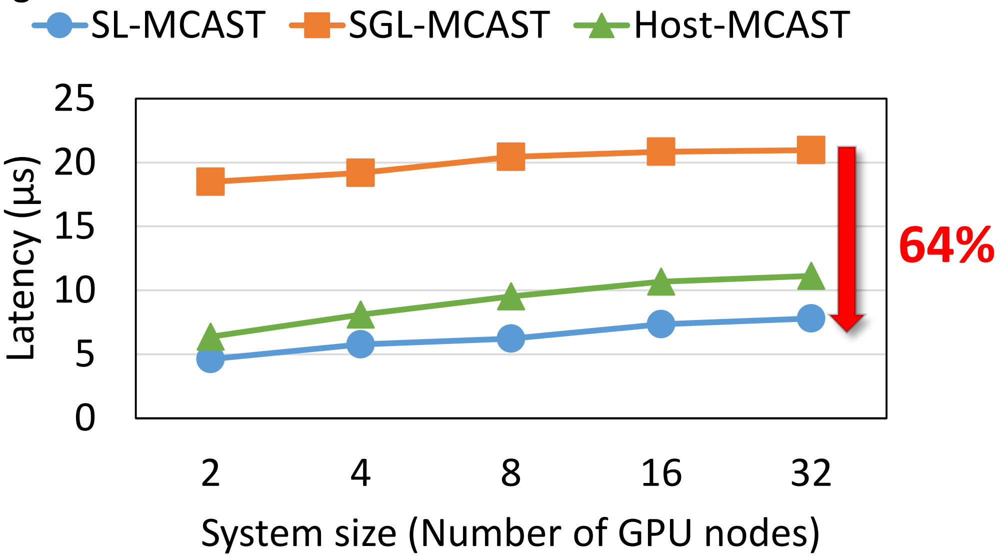
- Combining MCAST+GDR hardware features for heterogeneous configurations:
  - Source on the Host and destination on Device
  - SL design: Scatter at destination
    - Source: Data and Control on Host
    - Destinations: Data on Device and Control on Host
  - Combines IB MCAST and GDR features at receivers
  - CUDA IPC-based topology-aware intra-node broadcast
  - Minimize use of PCIe resources (Maximizing availability of PCIe Host-Device Resources)
- Available in MVAPICH2-GDR 2.3a



C.-H. Chu, K. Hamidouche, H. Subramoni, A. Venkatesh, B. Elton, and D. K. Panda.
“Designing High Performance Heterogeneous Broadcast for Streaming Applications on GPU Clusters, “ SBAC-PAD’16, Oct 2016.

Scalability Evaluation of the Proposed Design

- Inter-node experiments @ Wilkes cluster, 32 GPUs, 1 GPU/node
  - 1K byte messages

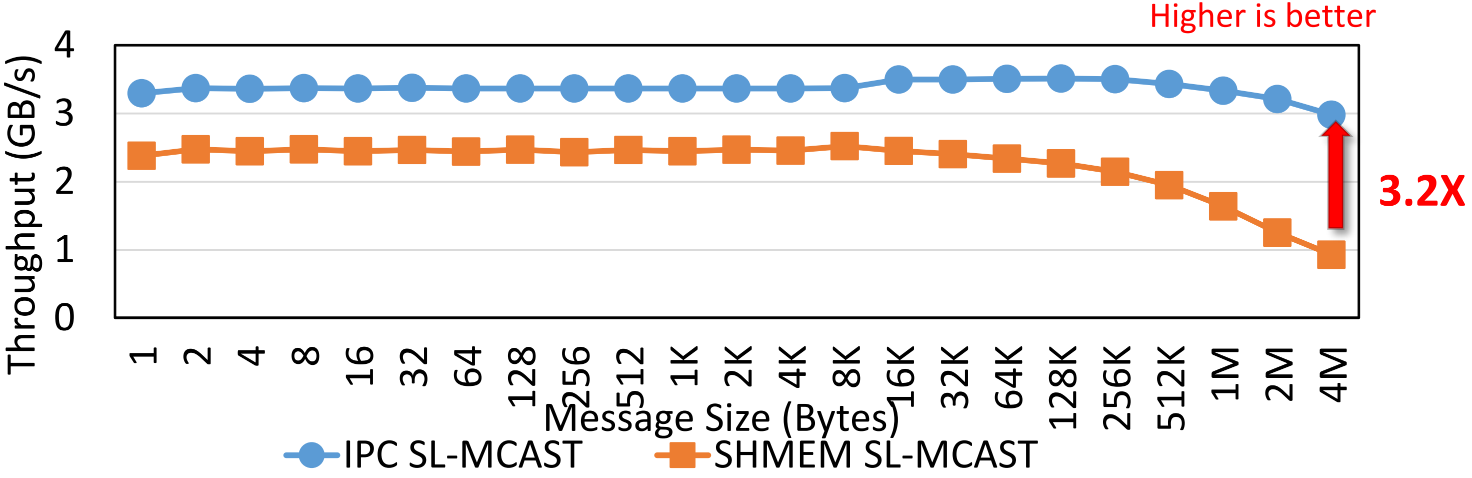


- **Maintain good Scalability while yielding up to 64% reduction of latency**

C.-H. Chu, K. Hamidouche, H. Subramoni, A. Venkatesh, B. Elton, and D. K. Panda.
“Designing High Performance Heterogeneous Broadcast for Streaming Applications on GPU Clusters, “ SBAC-PAD’16, Oct 2016.

Benefits of the Availability of Host-Device PCI Resources

- Mimic the behavior of streaming applications @ CSCS cluster, 88 GPUs, 8 NVIDIA K80 GPUs per node
  - Broadcast operations overlapped with application level Host-Device transfers



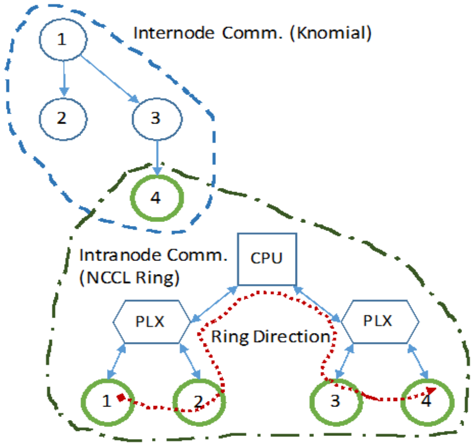
- **Maintain near-peak throughput over all message sizes**

C.-H. Chu, K. Hamidouche, H. Subramoni, A. Venkatesh, B. Elton, and D. K. Panda.
“Designing High Performance Heterogeneous Broadcast for Streaming Applications on GPU Clusters, “ SBAC-PAD’16, Oct 2016.

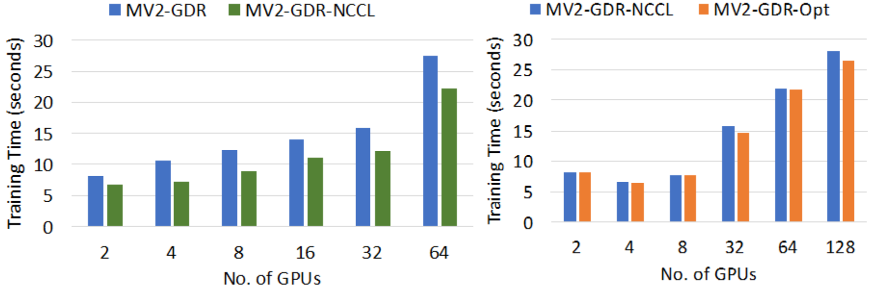
Outline

- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What’s new with MVAPICH2-GDR
    - Maximal overlap in MPI Datatype Processing
    - Efficient Support for Managed Memory
    - Support for OpenPower and NVLink
    - Initial support for GPUDirect Async feature
- Streaming Support with IB Multicast and GDR
- High-Performance Deep Learning with MVAPICH2-GDR
- Conclusions

Efficient Broadcast: MVAPICH2-GDR and NCCL



- NCCL 1.x had some limitations
  - Only worked for a single node; no scale-out on multiple nodes
  - Degradation across IOH (socket) for scale-up (within a node)
- We propose optimized MPI_Bcast design that exploits NCCL [1]
  - Communication of very large GPU buffers
  - Scale-out on large number of dense multi-GPU nodes
- Hierarchical Communication that efficiently exploits:
  - CUDA-Aware MPI_Bcast in MV2-GDR
  - NCCL Broadcast for intra-node transfers
- Can pure MPI-level designs be done that achieve similar or better performance than NCCL-based approach? [2]



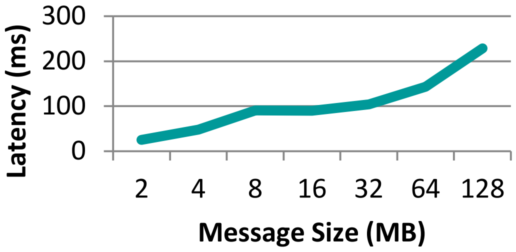
VGG Training with CNTK

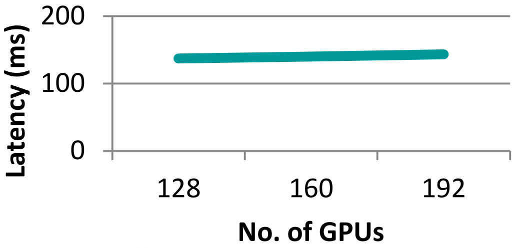
1. A. A. Awan, K. Hamidouche, A. Venkatesh, and D. K. Panda, Efficient Large Message Broadcast using NCCL and CUDA-Aware MPI for Deep Learning. In *Proceedings of the 23rd European MPI Users' Group Meeting* (EuroMPI 2016). [Best Paper Nominee]

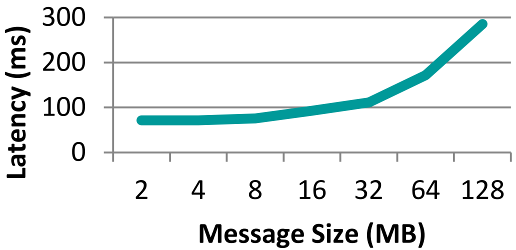
2. A. A. Awan, C-H. Chu, H. Subramoni, and D. K. Panda. Optimized Broadcast for Deep Learning Workloads on Dense-GPU InfiniBand Clusters: MPI or NCCL?, arXiv '17 (https://arxiv.org/abs/1707.09414)

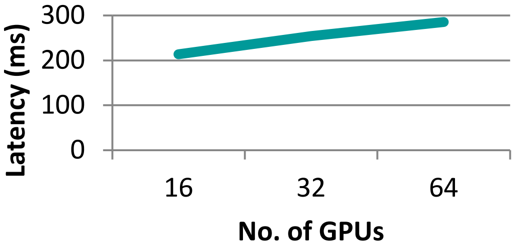
Large Message Optimized Collectives for Deep Learning

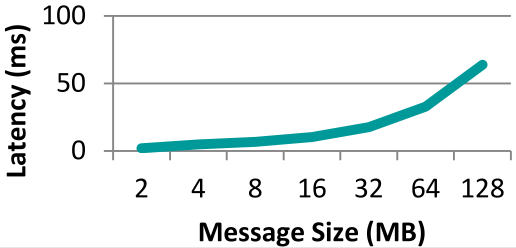
- MV2-GDR provides optimized collectives for **large message sizes**
- Optimized Reduce, Allreduce, and Bcast
- **Good scaling with large number of GPUs**
- **Available since MVAPICH2-GDR 2.2GA**

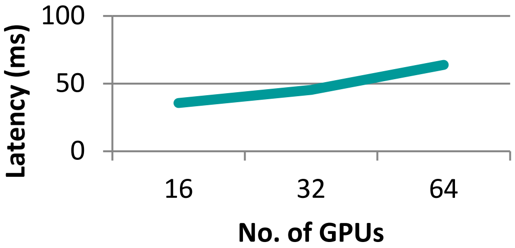

Reduce – 192 GPUs


Reduce – 64 MB


Allreduce – 64 GPUs

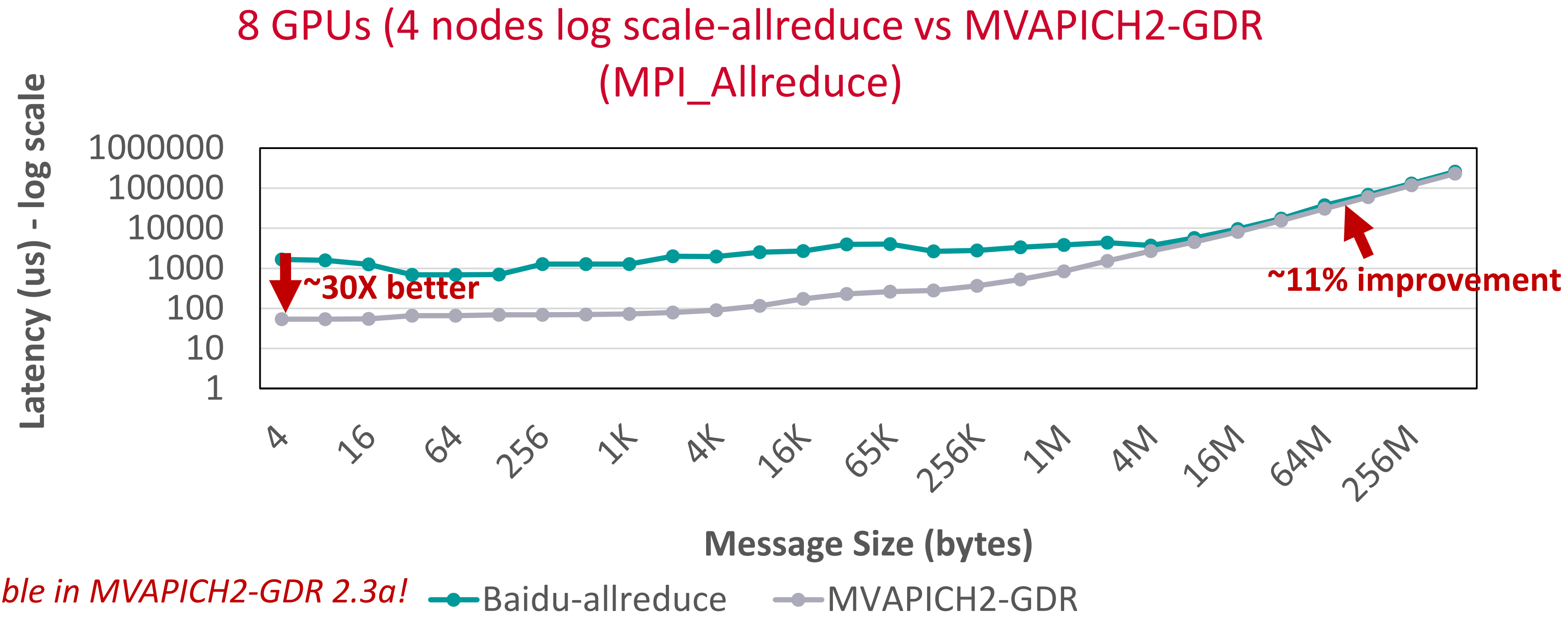

Allreduce - 128 MB


Bcast – 64 GPUs


Bcast 128 MB

MVAPICH2-GDR vs. Baidu-allreduce

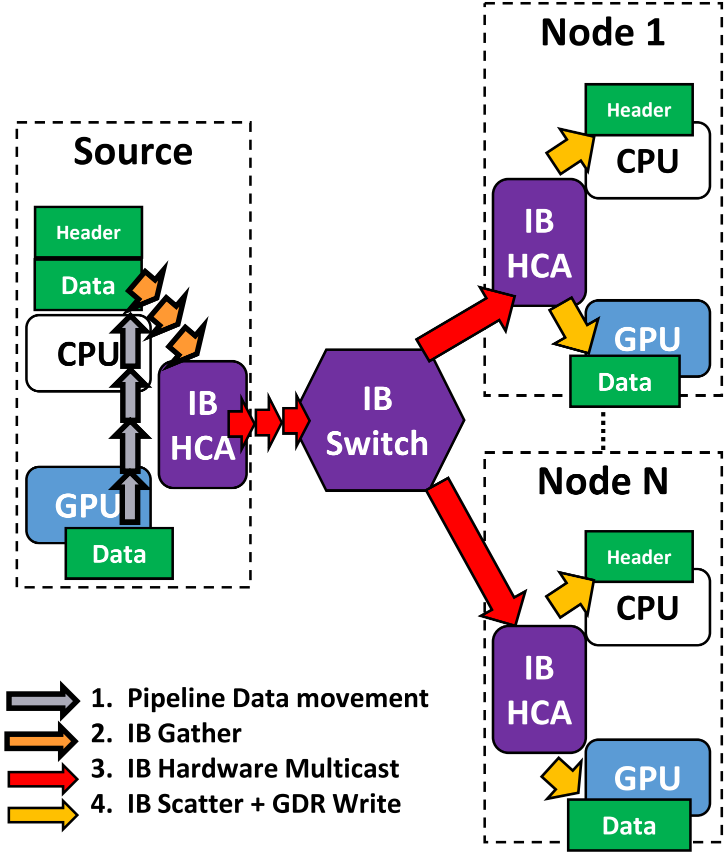
- Initial Evaluation shows promising performance gains for MVAPICH2-GDR 2.3a compared to Baidu-allreduce



Available in MVAPICH2-GDR 2.3a!

Exploiting GDR+IB-Mcast Design for Deep Learning Applications

- Optimizing MCAST+GDR Broadcast for deep learning:
  - Source and destination buffers are on GPU Device
    - Typically very large messages (>1MB)
  - Pipelining data from Device to Host
    - Avoid GDR read limit
    - Leverage high-performance SL design
  - Combines IB MCAST and GDR features
  - Minimize use of PCIe resources on the receiver side
    - Maximizing availability of PCIe Host-Device Resources
  - Available MVAPICH2-GDR 2.3a!



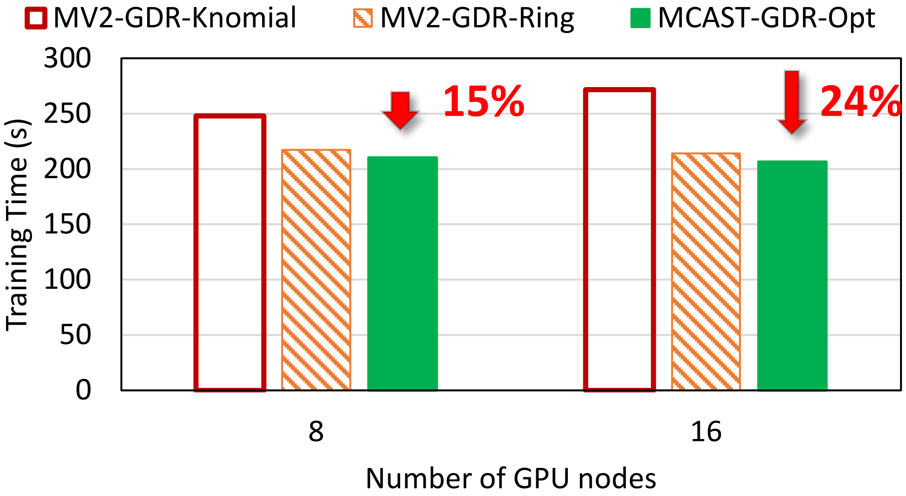
1. Pipeline Data movement
2. IB Gather
3. IB Hardware Multicast
4. IB Scatter + GDR Write

Ching-Hsiang Chu, Xiaoyi Lu, Ammar A. Awan, Hari Subramoni, Jahanzeb Hashmi, Bracy Elton, and Dhabaleswar K. Panda, "Efficient and Scalable Multi-Source Streaming Broadcast on GPU Clusters for Deep Learning , " ICPP'17.

Application Evaluation: Deep Learning Frameworks

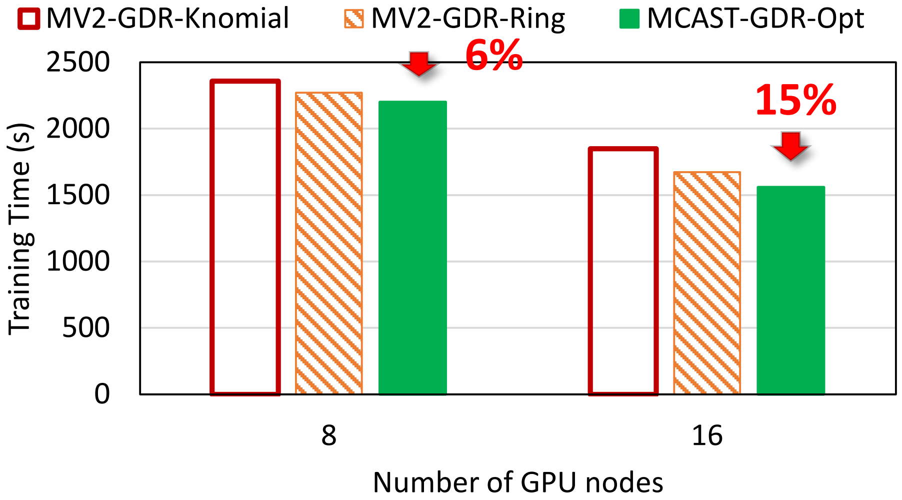
- @ RI2 cluster, 16 GPUs, 1 GPU/node
  - Microsoft Cognitive Toolkit (CNTK) *[https://github.com/Microsoft/CNTK]*

AlexNet model



Lower is better

VGG model

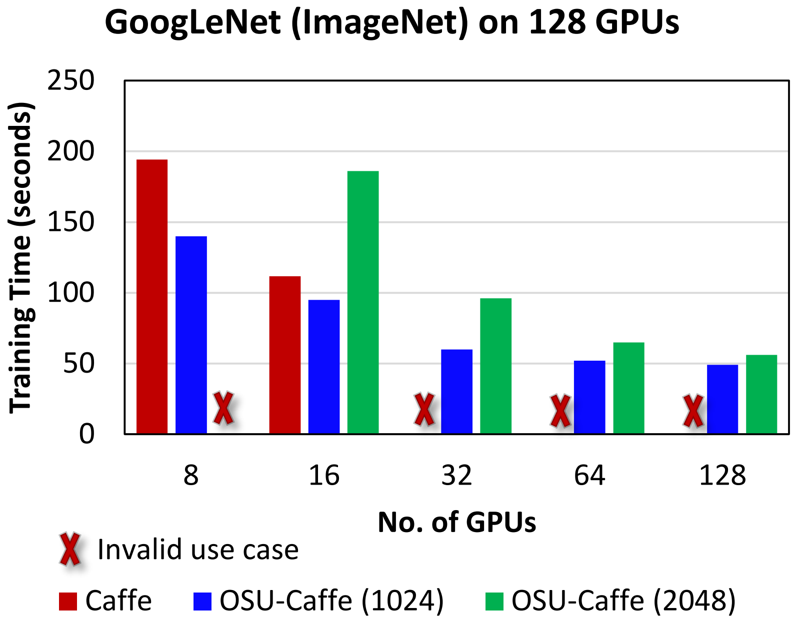


- Reduces up to 24% and 15% of latency for AlexNet and VGG models
- Higher improvement can be observed for larger system sizes

C.-H. Chu, X. Lu, A. A. Awan, H. Subramoni, J. Hashmi, B. Elton, and D. K. Panda, Efficient and Scalable Multi-Source Streaming Broadcast on GPU Clusters for Deep Learning, ICPP'17.

High-Performance Deep Learning (HiDL) with MVAPICH2-GDR

- Caffe : A flexible and layered Deep Learning framework.
- Benefits and Weaknesses
  - Multi-GPU Training within a single node
  - Performance degradation for GPUs across different sockets
  - No Scale-out available
- OSU-Caffe: MPI-based Parallel Training
  - Enable Scale-up (within a node) and Scale-out (across multi-GPU nodes)
  - Scale-out on 64 GPUs for training CIFAR-10 network on CIFAR-10 dataset
  - Scale-out on 128 GPUs for training GoogLeNet network on ImageNet dataset



OSU-Caffe publicly available from

http://hidl.cse.ohio-state.edu/

Outline

- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What’s new with MVAPICH2-GDR
  - Maximal overlap in MPI Datatype Processing
  - Efficient Support for Managed Memory
  - Support for OpenPower and NVLink
  - Initial support for GPUDirect Async feature
- Streaming Support with IB Multicast and GDR
- High-Performance Deep Learning with MVAPICH2-GDR
- Conclusions

Conclusions

- MVAPICH2 optimizes MPI communication on InfiniBand clusters with GPUs
- Provides optimized designs for point-to-point two-sided and one-sided communication, datatype processing and collective operations
- Takes advantage of CUDA features like IPC and GPUDirect RDMA families
- New designs help to get good performance for streaming and deep learning applications

Thank You!

panda@cse.ohio-state.edu

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

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