



MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

High Performance Broadcast with GPUDirect RDMA and
InfiniBand Hardware Multicast for Streaming Applications

GTC 2015



MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

Presented By

Dhabaleswar K. (DK) Panda

The Ohio State University

Email: panda@cse.ohio-state.edu

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



- **Introduction**
- Motivation and Problem Statement
- Design Considerations
- Proposed Approach
- Results
- Conclusion and Future Work

- Examples - surveillance, habitat monitoring, 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**



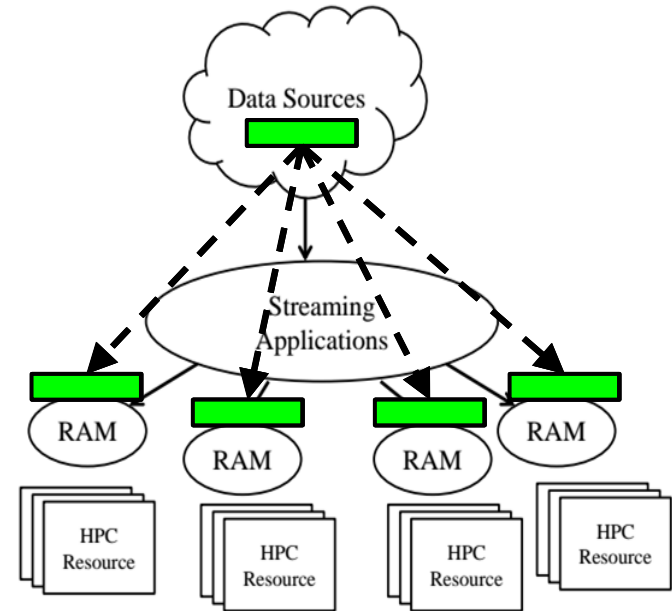
- Proliferation of Multi-Petaflop systems
- Heterogeneity in compute resources with GPGPUs
- High performance interconnects with RDMA capabilities to host and GPU memories
- Streaming applications leverage on such resources





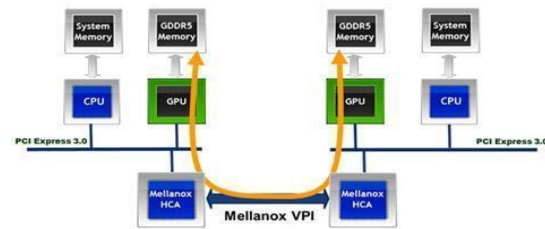
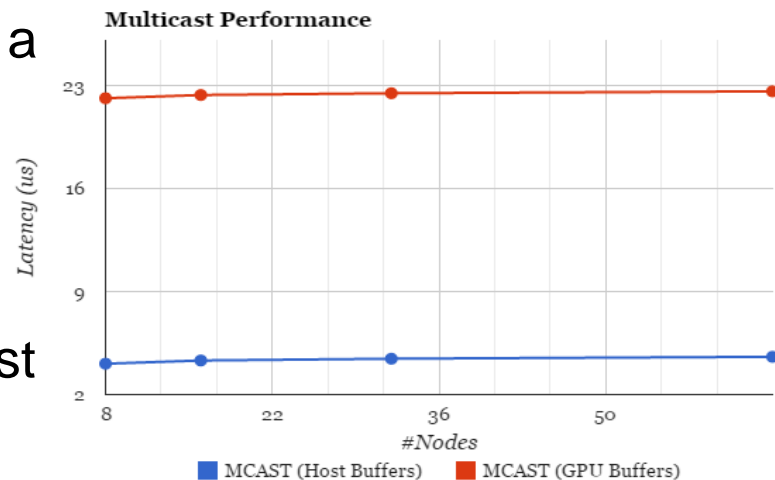
- Introduction
- **Motivation and Problem Statement**
- Design Considerations
- Proposed Approach
- Results
- Conclusion and Future Work

- Pipelined data parallel compute phases that form the crux of streaming applications lend themselves for GPGPUs
- Data distribution to GPGPU sites occur over PCIe within the node and over InfiniBand interconnects across nodes
- **Broadcast operation is a key dictator of throughput of streaming applications**
- **Reduced latency for each operation**
- **Support multiple back-to-back operations**
- **More critical with accelerators**



Courtesy: Agarwalla, Bikash, et al. "Streamline: A scheduling heuristic for streaming applications on the grid." *Electronic Imaging 2006*

- Traditional short message broadcast operation between GPU buffers involves a **Host-Staged Multicast (HSM)**
- Data copied from GPU buffers to host memory
- Using InfiniBand Unreliable Datagram(UD)-based hardware multicast
- **Sub-optimal use of near-scale invariant UD-multicast performance**
- **PCIe resources wasted and benefits of multicast nullified**
- **GPU-Direct RDMA capabilities unused**





- Can we design a GPU broadcast mechanism that can completely avoid host-staging for streaming applications?
- Can we harness the capabilities of GPU-Direct RDMA (GDR)?
- Can we overcome limitations of UD transport and realize the true potential of multicast for GPU buffers?
- Succinctly, how do we multicast GPU data using GDR efficiently?



- Introduction
- Motivation and Problem Statement
- **Design Considerations**
- Proposed Approach
- Results
- Conclusion and Future Work



- Introduction
- Motivation and Problem Statement
- **Design Considerations**
 - **Critical Factors**
- Proposed Approach
- Results
- Conclusion and Future Work

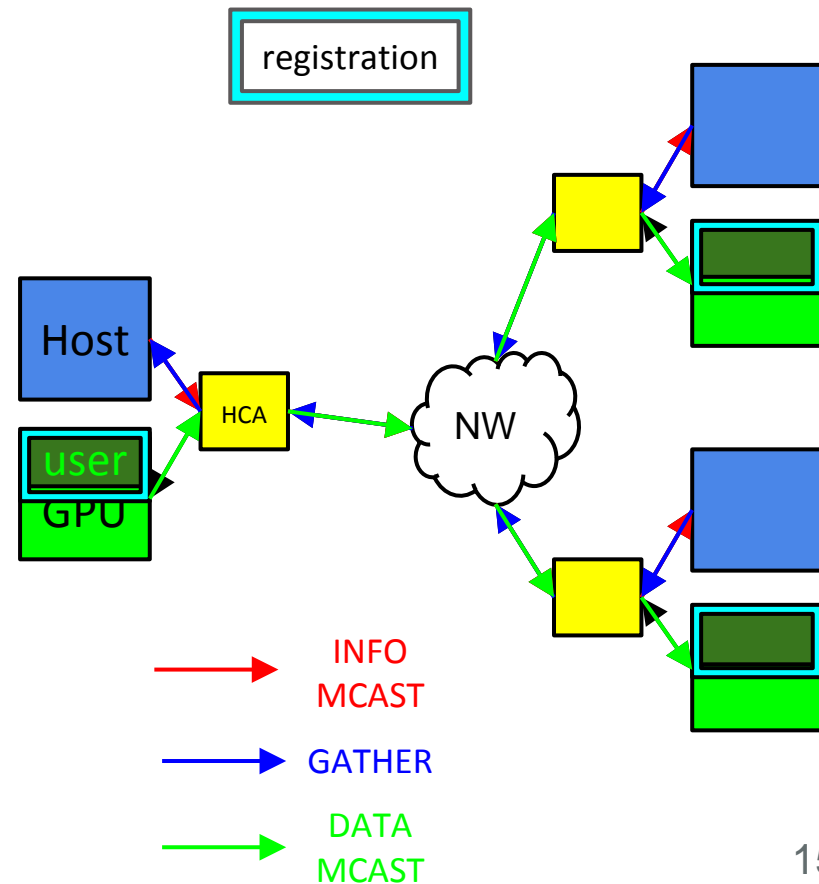


- Goal is to be able to multicast GPU data in lesser time than the host-staged multicast (~20us)
- Cost of `cudaMemcpy` is ~8us for short messages for host->gpu, gpu->host and gpu->gpu transfers
- `CudaMemcpy` costs and memory registration costs determine the viability of a multicast protocol for GPU buffers



- Introduction
- Motivation and Problem Statement
- **Design Considerations**
 - Eager Protocol
 - Rendezvous Protocol
- Proposed Approach
- Results
- Conclusion and Future Work

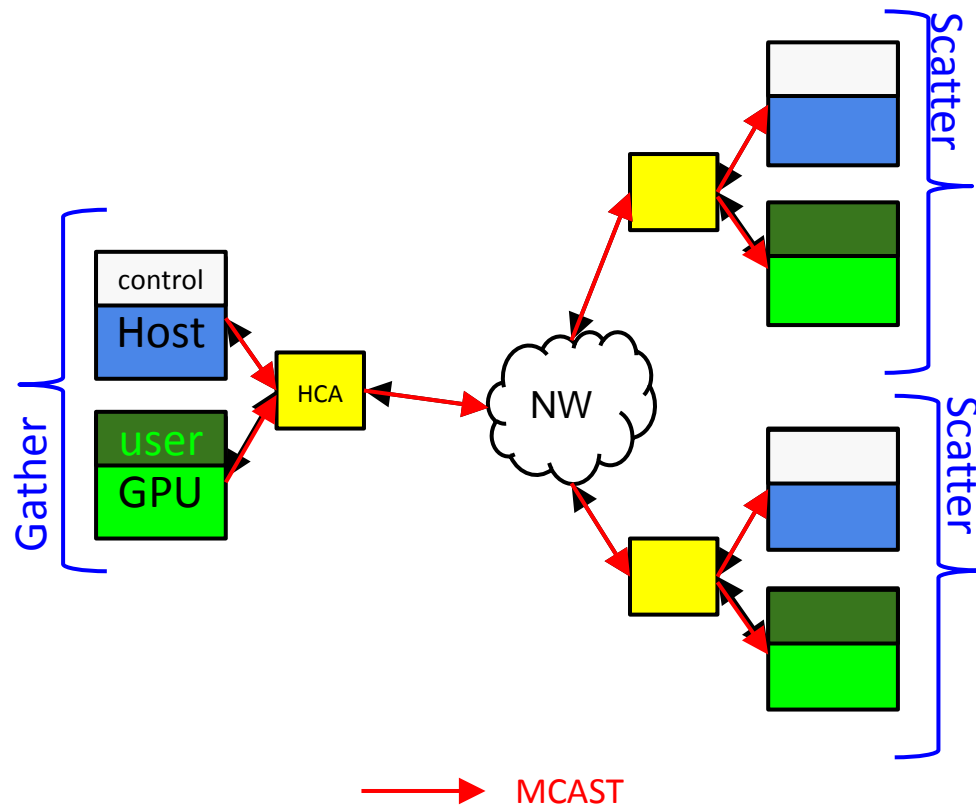
- Register user GPU data and start *RTS multicast* with control info
- Confirm ready receivers \equiv 0-byte gather
- Perform Data Multicast
- Registration cost and gather limitations
- Handshake for each operation – not required for streaming applications which are error tolerant





- Introduction
- Motivation and Problem Statement
- Design Considerations
- **Proposed Approach**
- Results
- Conclusion and Future Work

- One time registration of window of persistent buffers in streaming apps
- Combine control and user data at the source and scatter them at the destinations using *Scatter-Gather-List* abstraction
- Scheme lends itself for pipelined phases abundant in Streaming Applications and avoids stressing PCIe

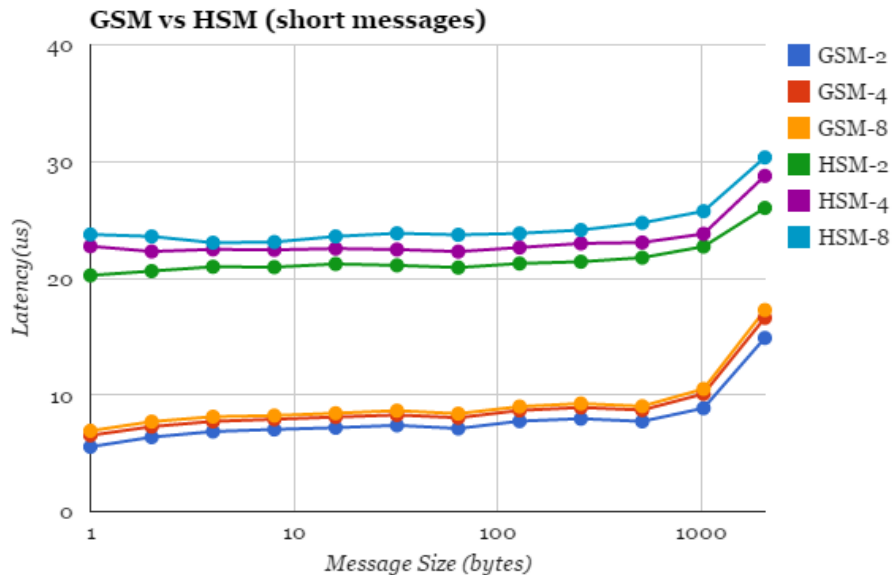




- Introduction
- Motivation and Problem Statement
- Design Considerations
- Proposed Approach
- **Results**
- Conclusion and Future Work

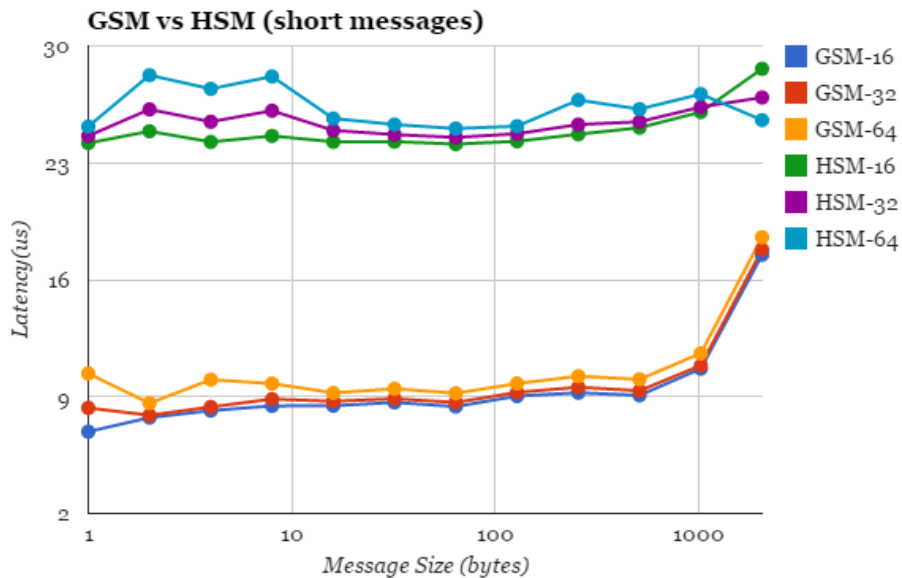


- Experiments were run on Wilkes @ University of Cambridge
- 12-core Ivy Bridge Intel(R) Xeon(R) E5-2630 @ 2.60 GHz with 64 GB RAM
- FDR ConnectX2 HCAs
- NVIDIA K20c GPUs
- Mellanox OFED version MLNX OFED LINUX-2.1-1.0.6 which supports GPUDirect-RDMA (GDR) required
- Baseline Host-based MCAST uses MVAPICH2-GDR (<http://mvapich.cse.ohio-state.edu/downloads>)
- GDR-SGL-MCAST is based on MVAPICH2-GDR



- GDR-SGL-MCAST (**GSM**)
- Host-Staged-MCAST (**HSM**)
- GSM Latency $\leq \sim 10\mu\text{s}$ vs HSM Latency $\leq \sim 23\mu\text{s}$
- Small latency increase with scale

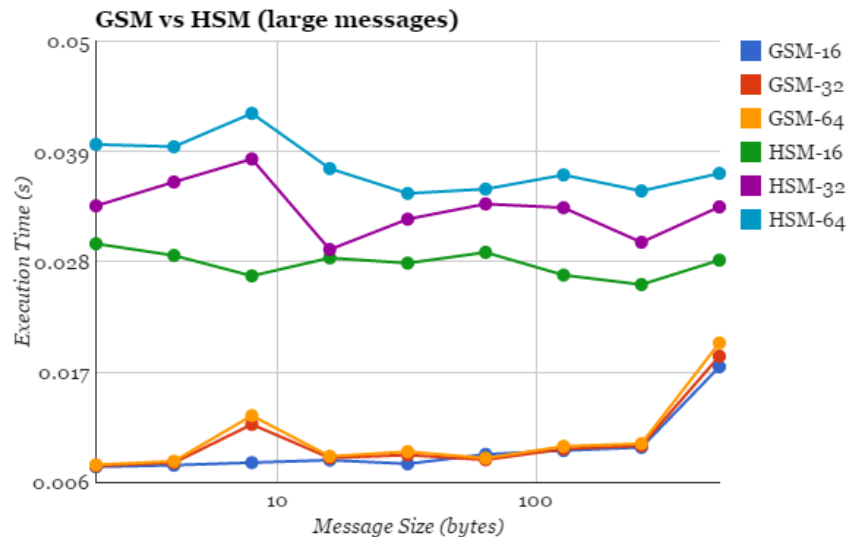
A. Venkatesh, H. Subramoni, K. Hamidouche and D. K. Panda, A High Performance Broadcast Design with Hardware Multicast and GPUDirect RDMA for Streaming Applications on InfiniBand Clusters, IEEE International Conference on High Performance Computing (HiPC '14), Dec 2014.



- Both GSM and HSM continue to show near scale invariant latency with **60%** improvement (8 bytes)



- Based on a synthetic benchmark that mimics broadcast patterns in Streaming Applications
- Long window of persistent m-byte buffers with 1,000 back-to-back multicast operations issued
- Execution time reduces by 3x-4x





- Introduction
- Motivation and Problem Statement
- Design Considerations
- Proposed Approach
- Results
- **Conclusion and Future Work**



- Designed an efficient GPU data broadcast for streaming applications which uses near-constant-latency hardware multicast feature and GPUDirect RDMA
- Proposed a new methodology which overcomes the performance challenges posed by UD transport
- Benefits shown with latency and streaming-application-communication mimicking throughput benchmark
- Exploration of NVIDIA's Fastcopy module for MPI_Bcast

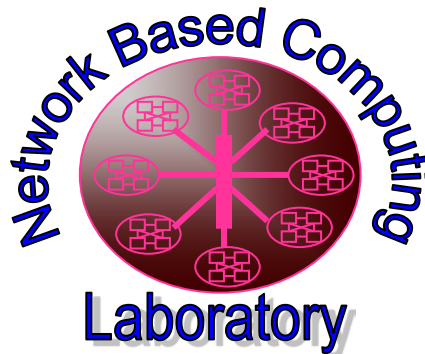


Learn about recent advances and upcoming features in CUDA-aware MVAPICH2-GPU library

- S5461 - Latest Advances in MVAPICH2 MPI Library for NVIDIA GPU Clusters with InfiniBand
- Thursday, 03/19 (Today)
- Time: 17:00–17:50
- Room 212 B

Contact

panda@cse.ohio-state.edu



<http://mvapich.cse.ohio-state.edu>

<http://nowlab.cse.ohio-state.edu>