MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

OMB-UM: Design, Implementation, and Evaluation of CUDA Unified Memory Aware MPI Benchmarks

SC '19 Booth Talk

Karthik Vadambacheri Manian, Ching-Hsiang Chu, Ammar Ahmad Awan, Kawthar Shafie Khorassani, Hari Subramoni, and Dhabaleswar K. Panda

Network Based Computing Laboratory (NBCL)

Dept. of Computer Science and Engineering

The Ohio State University

{vadambacherimanian.1, chu.368, awan.10, shafiekhorassani.1, subramoni.1, panda.2}@osu.edu

Agenda

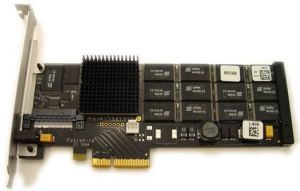
- **Introduction**
- Motivation
- Research Challenges
- Design
- Evaluation
- Discussion
- Conclusion

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)

Summit

Sunway TaihuLight

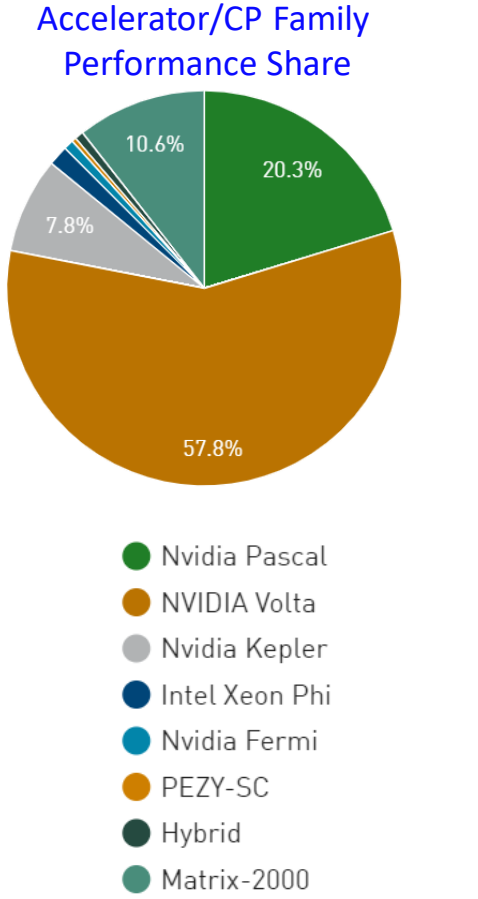
Sierra

K - Computer

GPUs in HPC

- GPUs are growing faster in the HPC arena

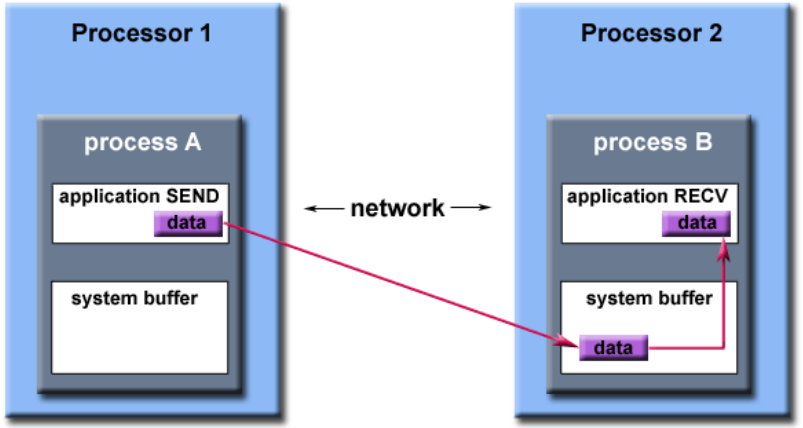
- NVIDIA GPUs - main driving force for
  - Accelerating traditional HPC applications
  - Accelerating AI thru faster training of Deep Neural Networks (DNNs)



https://www.top500.org (Nov '18)

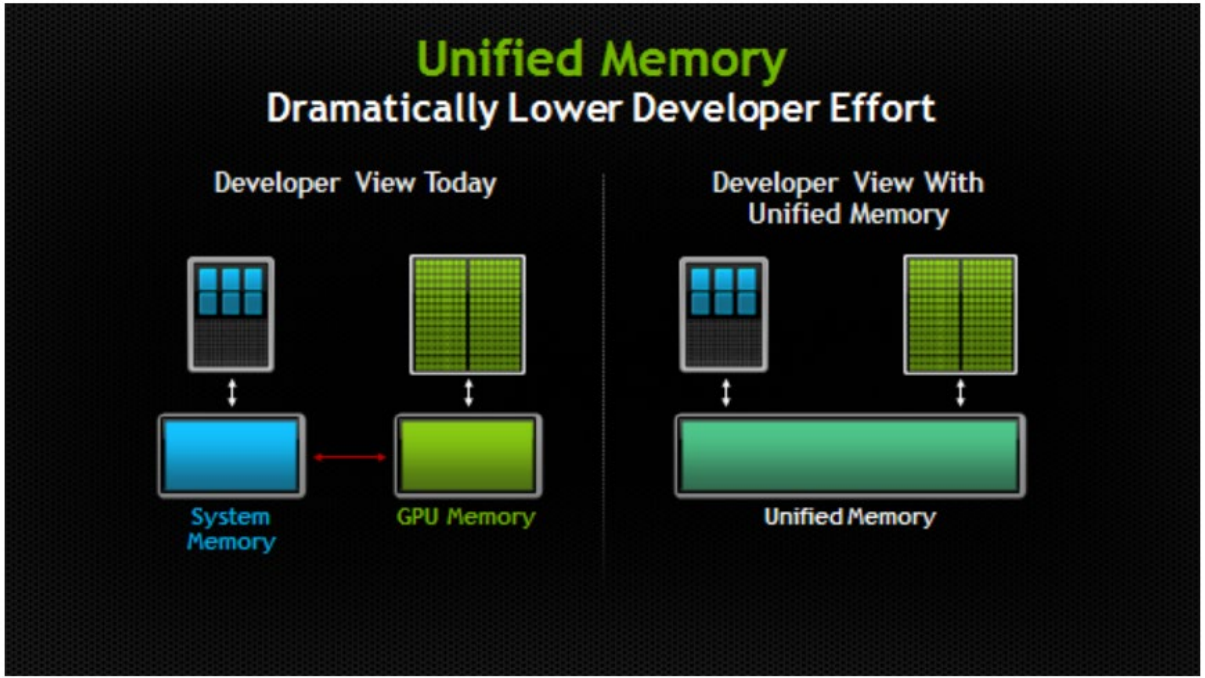
Message Passing Interface (MPI)

- Popular parallel programming model to harness the power of nodes in a clusters
- Data is explicitly sent by one process and received by another process in a cooperative fashion.
- MPI libraries can be
  - CUDA Aware
  - UM Aware



Path of a message buffered at the receiving process

Managed/Unified Memory



UM effective location => where UM currently resides

Courtesy: NVIDIA developer blogs

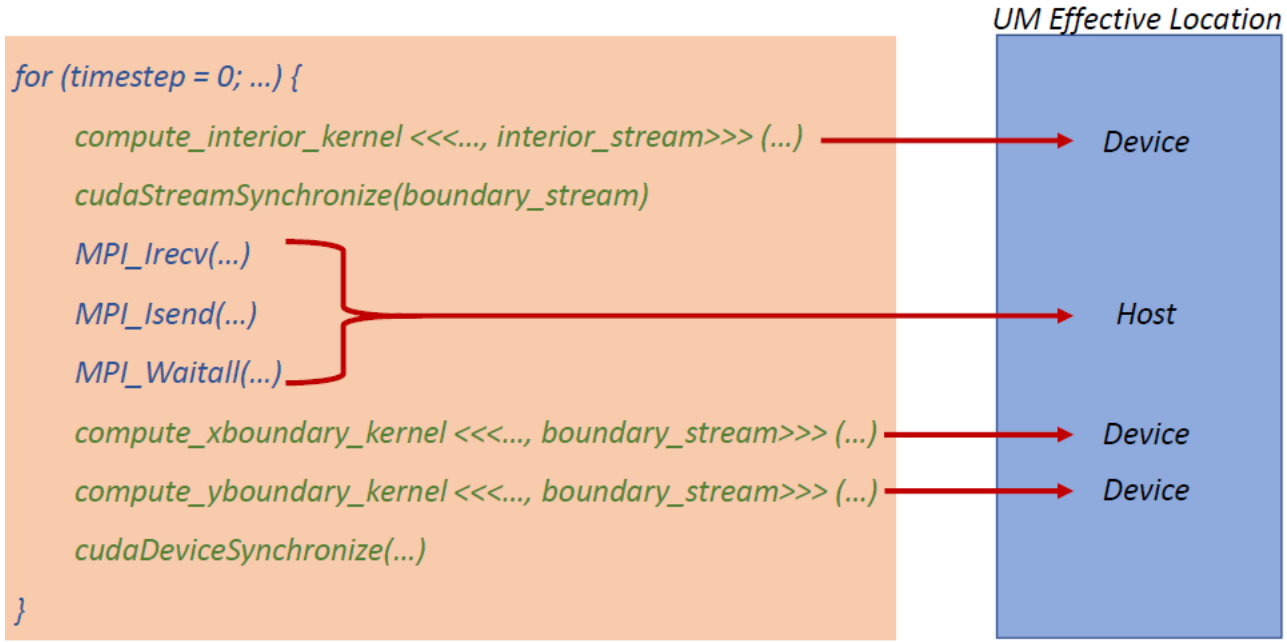
OSU Micro-Benchmarks (OMB)

- MPI, OpenSHMEM, UPC & UPC++ benchmarks
- Pt2Pt, Collective & One-sided
  - Blocking & Non-blocking
- Support for CUDA & OpenACC extensions
  - Support for **CUDA Managed/Unified Memory**

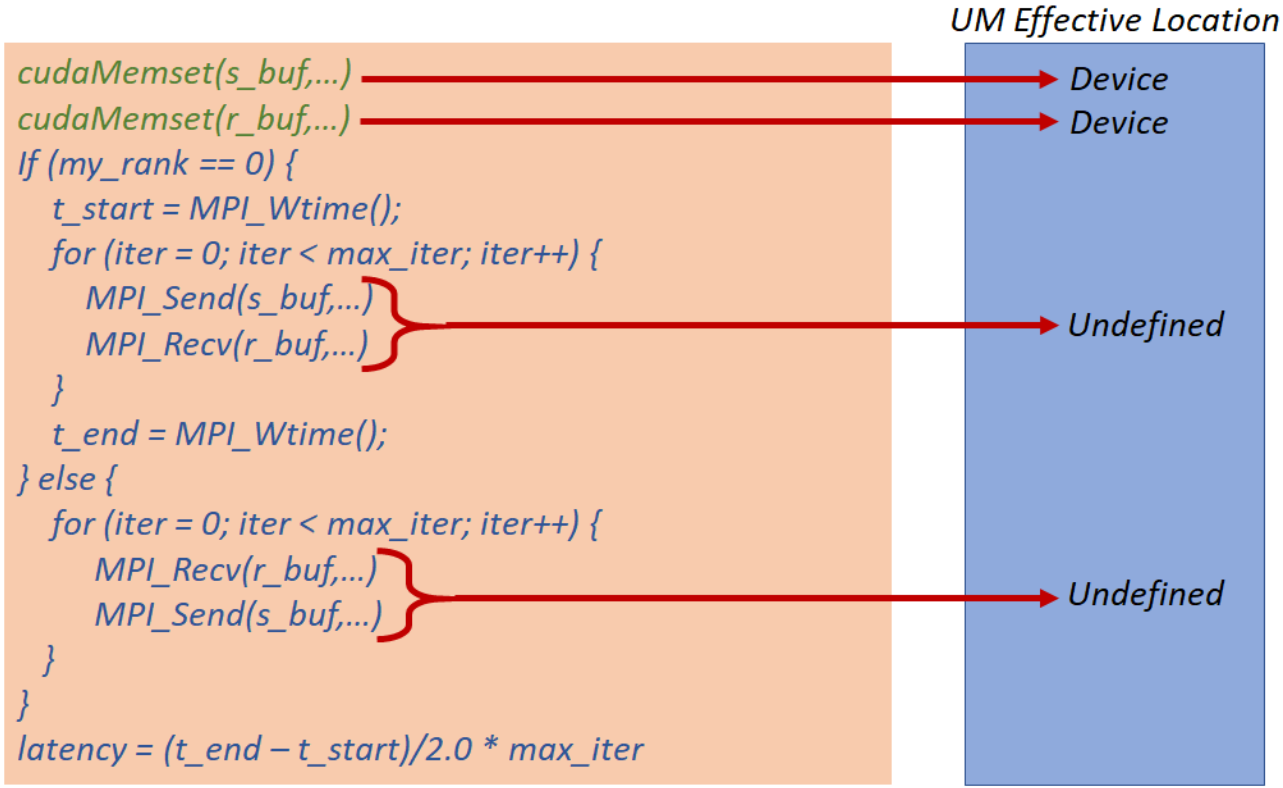
Agenda

- Introduction
- **Motivation**
- Research Challenges
- Design
- Evaluation
- Discussion
- Conclusion

2D Stencil Pseudocode



OSU_latency Micro-Benchmark Pseudocode



Limitations in current state of the art

- Oblivious to the effective location of UM buffers
- No provision to set the 4 possible UM effective locations
  - MH-MH
  - MD-MH
  - MH-MD
  - MD-MD
- In conclusion, there is a need for properly benchmarking middleware libraries on UM buffers

Agenda

- Introduction
- Motivation
- **Research Challenges**
- Design
- Evaluation
- Discussion
- Conclusion

Broad Challenge

How can a full-fledged UM Aware OMB (OMB-UM) be designed to provide the facility to set the four possible effective locations for UM buffers leading to the full characterization of UM aware MPI on modern GPU clusters?

Research Challenges

How to achieve the different data placements for UM buffer?

What are the characteristics of the CUDA kernels employed for UM data placement?

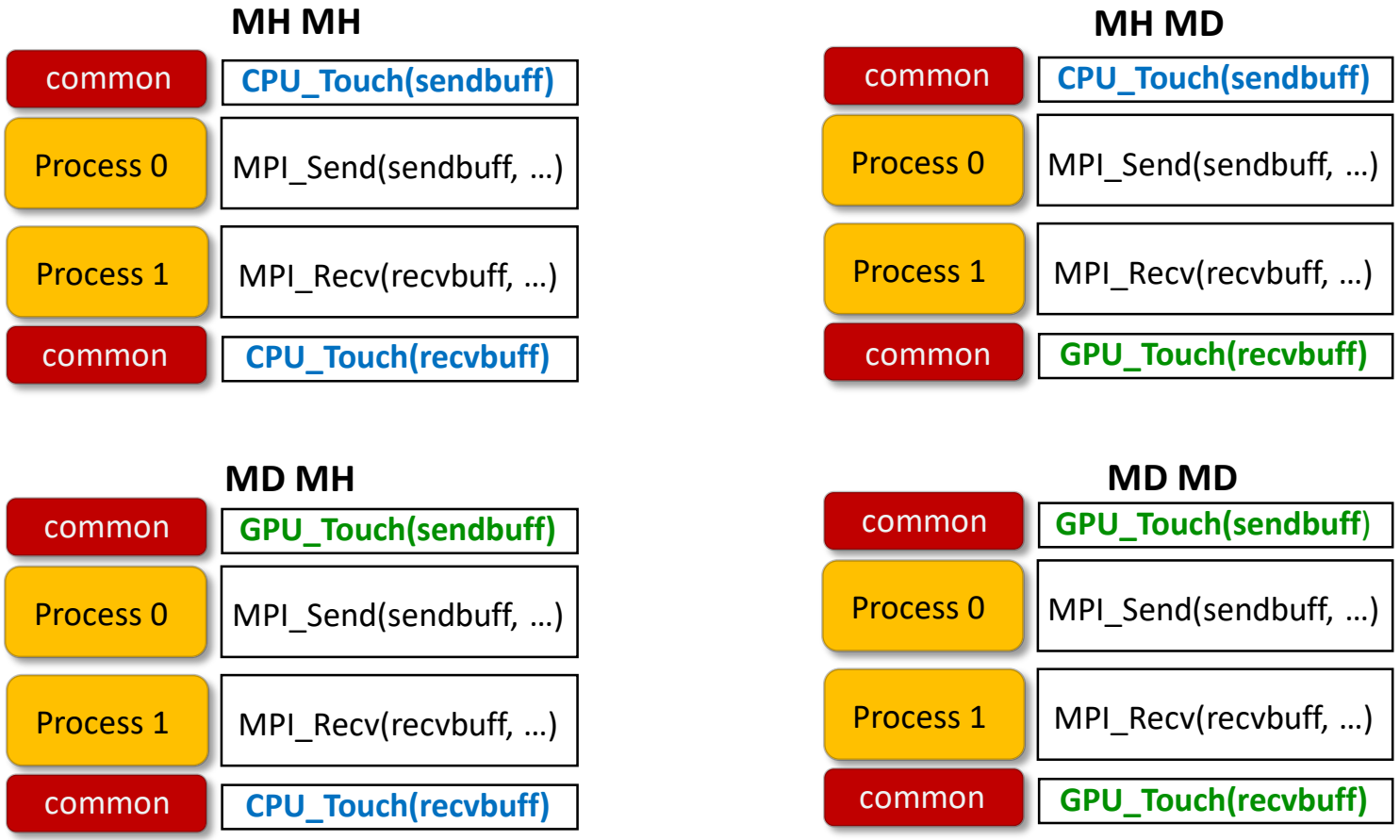
Can the performance of UM aware MPI be characterized fully?

Let's design OMB-UM

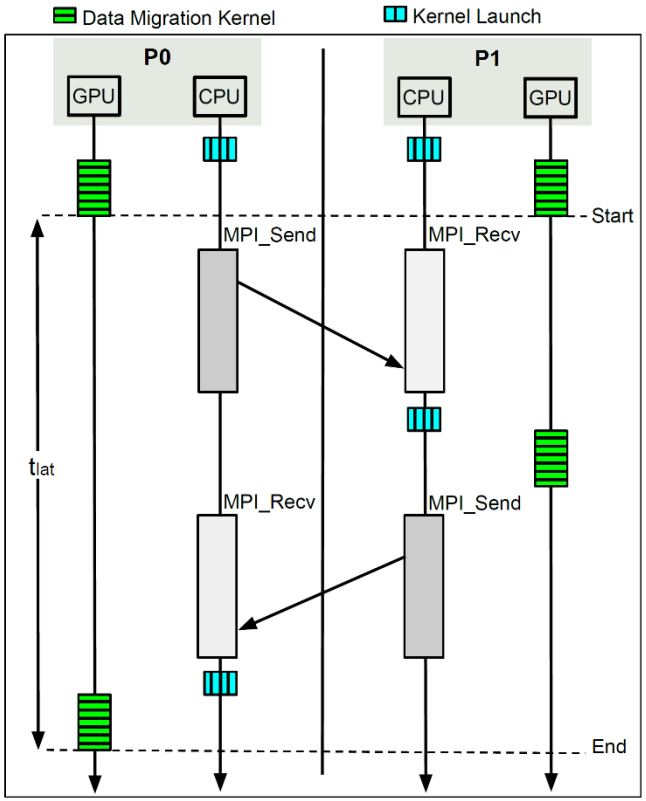
Agenda

- Introduction
- Motivation
- Research Challenges
- **Design**
- Evaluation
- Discussion
- Conclusion

UM Buffer Placements

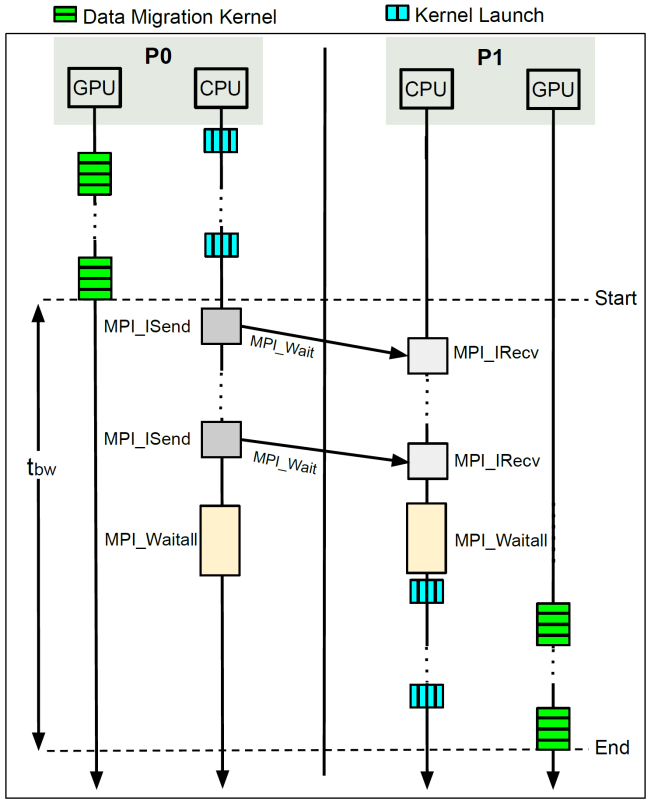


Proposed Latency Benchmark (MD MD)



$$Latency_{MD\text{-}MD} = (t_{End} - t_{start} - 2 \times t_{Kernel\_Launch}) / 2$$

Proposed Bandwidth Benchmark (MD MD)



$$Bandwidth_{MD\text{-}MD} = (M \times window\_size)/(t_{bw} - t_{Kernel\_Launch})$$

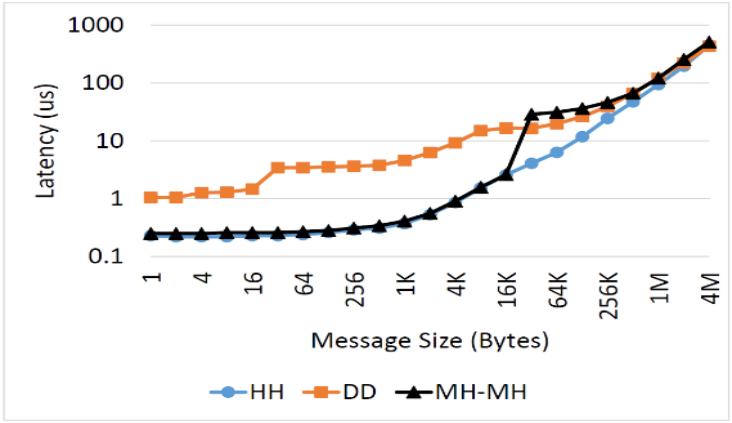
Agenda

- Introduction
- Motivation
- Research Challenges
- Design
- **Evaluation**
- Discussion
- Conclusion

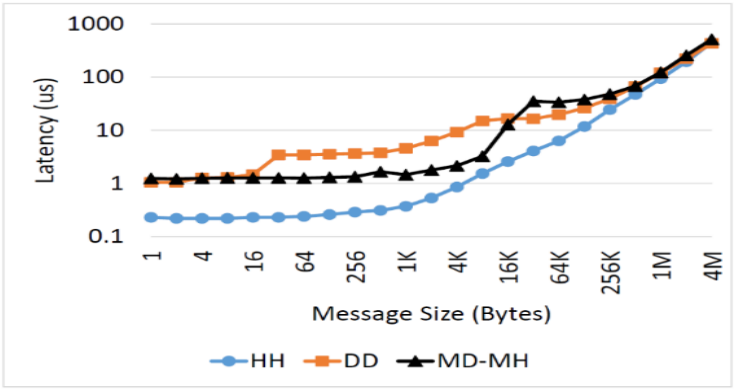
Evaluation Platforms

CPU	GPU	Interconnect	CPU-GPU Interconnect	OS
Sandy Bridge E5-2670	Volta V100	Infiniband EDR	PCIe Gen3	RHEL 7.5.1804
Haswell E5-2687W	Volta V100	Infiniband EDR	PCIe Gen3	RHEL 7.5.1804
Summit	Volta V100	Infiniband EDR	NVLink 2.0	RHEL 7.6

X86 Intra-node Pt2Pt Evaluation on MVAPICH2-GDR

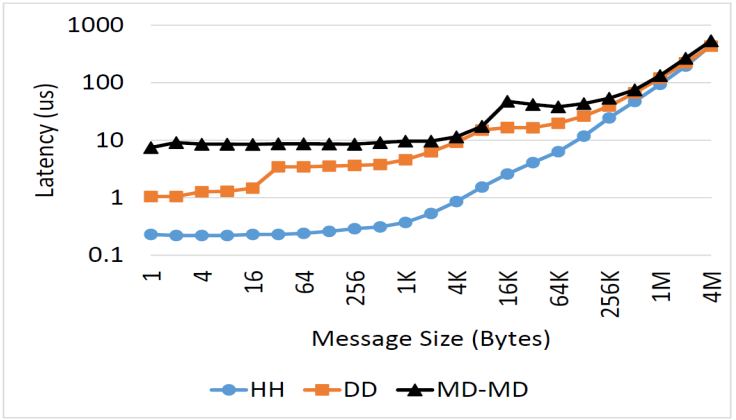


Latency MH MH



Latency MD MH

- Latency MH MH has bump due to advanced managed memory designs
- Performance of managed buffers on par with device and host buffers



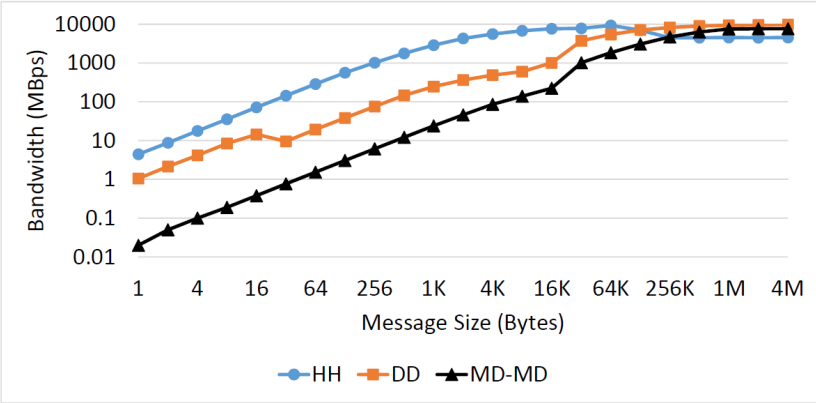
Latency MD MD

X86 Intra-node Pt2Pt Evaluation on MVAPICH2-GDR

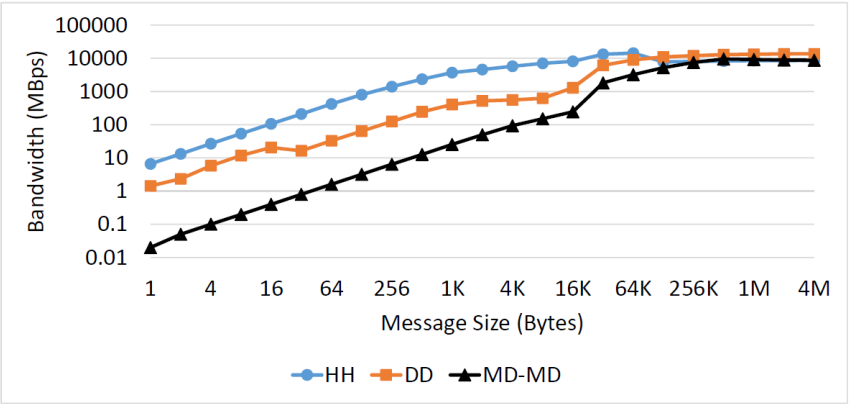
- Small to medium message bandwidth for managed buffers needs improvement
  - Caused by excessive movement of UM buffers between host & device
  - Performance worsens when the size of the message buffer increases

Managed Buffer Page Faults

On GPU	On CPU
65293	101630



Bandwidth MD MD



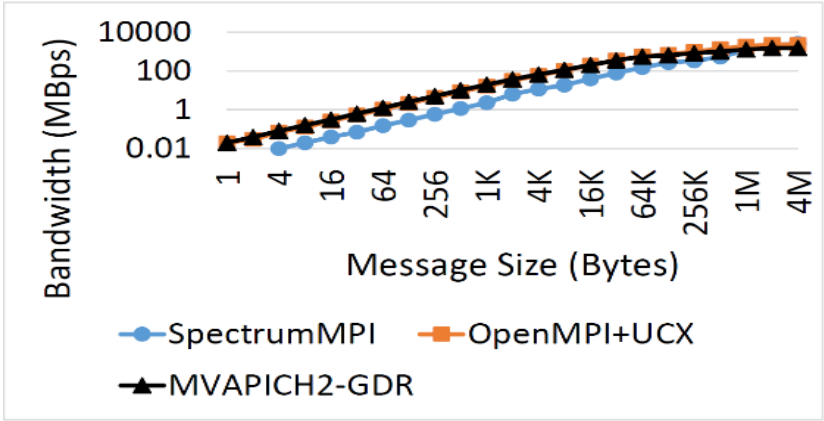
Bi-Bandwidth MD MD

OpenPOWER Inter-node Pt2Pt Evaluation

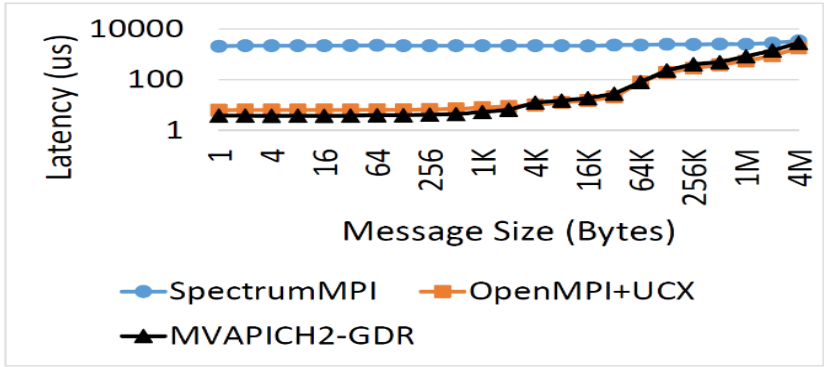
- SpectrumMPI shows very high latency with managed buffers
- Might be due to unnecessary data movement
- Page faults are 5X compared to OpenMPI

Managed Page Faults

MPI Library	On GPU	On CPU
OpenMPI+UCX	70445	74020
SpectrumMPI	351864	390526



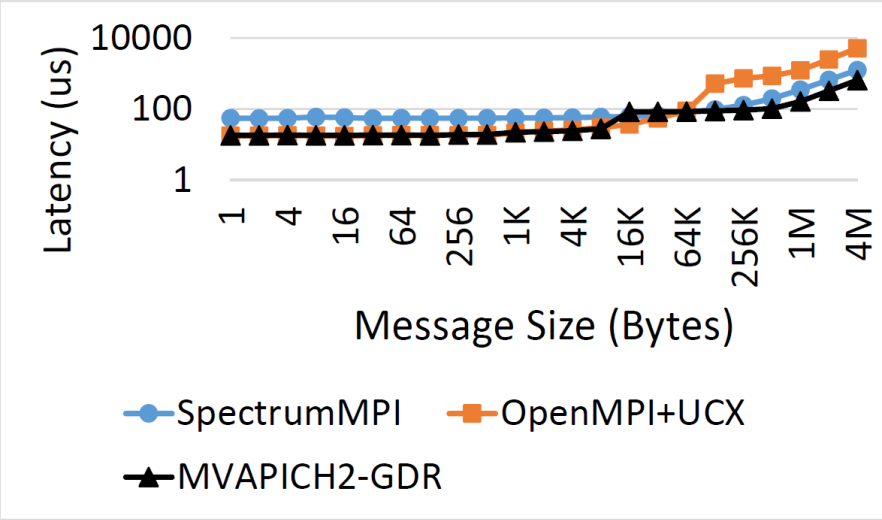
Bandwidth MD MD



Latency MD MD

OpenPOWER Intra-node Collective Evaluation

- Evaluated MPI_Bcast() operation on 6 GPUs
- SpectrumMPI and suffers with performance issues for small messages
- OpenMPI suffers with performance issues for large messages



Evaluating broadcast on managed buffers

Conclusion

- Current state of the art UM-Aware benchmarks do not accurately capture the effective location of UM buffer
- The proposed OMB-UM provides necessary options to set the effective location of UM buffer
- Insights obtained from OMB-UM benchmark results can be used to improve MPI performance
- OMB-UM design will be a part of the OMB suite in the future releases

Thank You!

{vadambacherimanian.1, chu.368, awan.10, shafiekhorassani.1, subramoni.1, panda.2}@osu.edu

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

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

Backup Slides

Enhanced H/W Support for Unified Memory on Pascal/Volta

- GPU page faulting hardware support introduced in Pascal/Volta
  - Only faulting pages need to be migrated on-demand
- Hardware access counters makes only the most needed pages migrated on-demand
- On IBM Power systems, new Address Translation Services (ATS) allows a GPU to access CPU's page table directly

Enhanced API Support for Unified Memory on Pascal/Volta

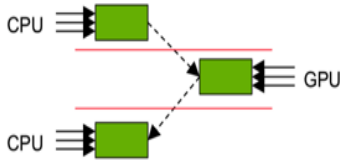
- Hints like **cudaMemAdvise** and **cudaMemPrefetchAsync**() are very useful

USER HINTS
Prefetching

```
char *data;
cudaMallocManaged(&data, N);

init_data(data, N);

cudaMemPrefetchAsync(data, N, myGpuId, s);
mykernel<<<..., s>>>(data, N);
cudaMemPrefetchAsync(data, N, cudaCpuDeviceId, s);
cudaStreamSynchronize(s);

use_data(data, N);

cudaFree(data);
```

Page faults can be expensive and they stall SM execution

Avoid faults by prefetching data to the accessing processor

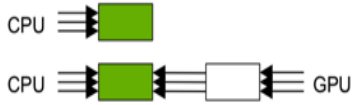


USER HINTS
Preferred Location

```
char *data;
cudaMallocManaged(&data, N);

init_data(data, N);

cudaMemAdvise(data, N, ..PreferredLocation, cudaCpuDeviceId);

mykernel<<<..., s>>>(data, N);


use_data(data, N);

cudaFree(data);
```

Here the kernel will *page fault* and generate direct mapping to data on the CPU

The driver will "resist" migrating data away from the preferred location



Courtesy: http://on-demand.gputechconf.com/gtc/2017/presentation/s7285-nikolay-sakharnykh-unified-memory-on-pascal-and-volta.pdf

Microbenchmarks

- Helps to characterize a system
- Provides various options for experimentation
- Benchmark results should be unambiguous

Managed/Unified Memory (Contd)

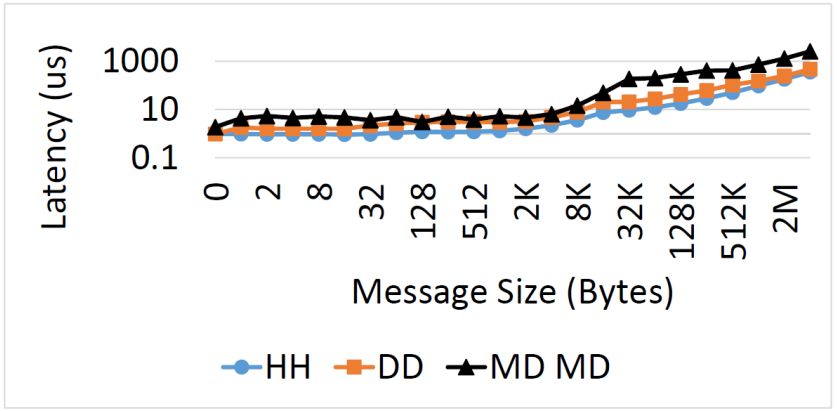
CPU CODE

```
void sortfile(FILE *fp, int N) {
    char *data;
    data = (char *)malloc(N);
    fread(data, 1, N, fp);
    qsort(data, N, 1, compare);
    use_data(data);
    free(data);
}
```

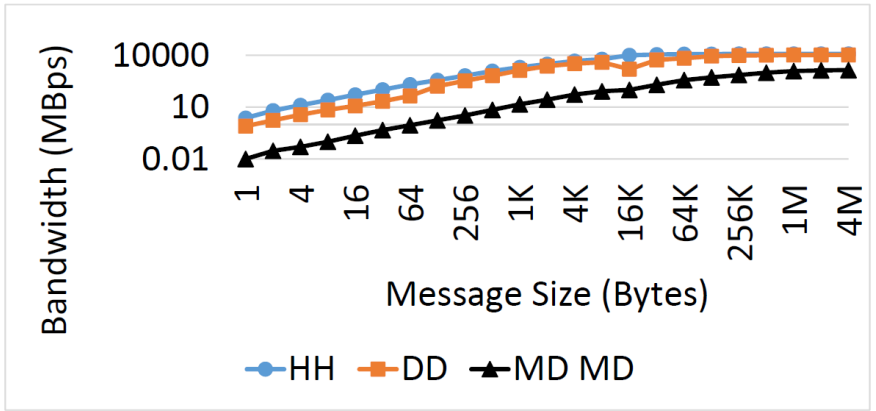
CUDA CODE with Unified Memory

```
void sortfile(FILE *fp, int N) {
    char *data;
    cudaMallocManaged(&data, N);
    fread(data, 1, N, fp);
    qsort<<<…>>>(data, N, 1, compare);
    cudaDeviceSynchronize();
    use_data(data);
    cudaFree(data);
}
```

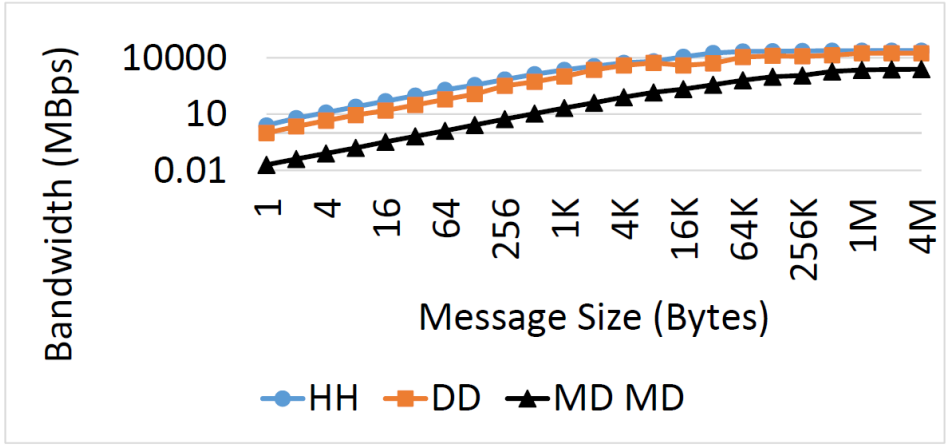
X86 Inter-node Evaluation on MVAPICH2-GDR



Latency MD MD



Bandwidth MD MD



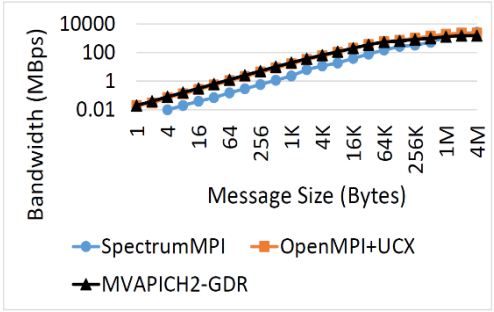
Bi-Bandwidth MD MD

OpenPOWER Inter-node Pt2Pt Evaluation

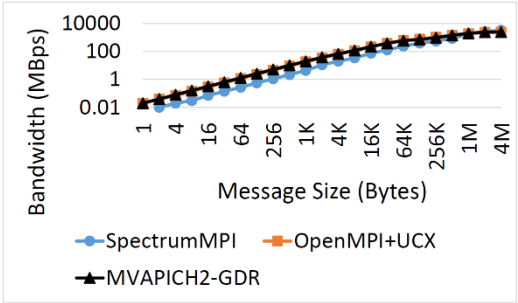
- SpectrumMPI shows very high latency with managed buffers
- Might be due to unnecessary data movement
- Page faults are 5X compared to OpenMPI

Managed Page Faults

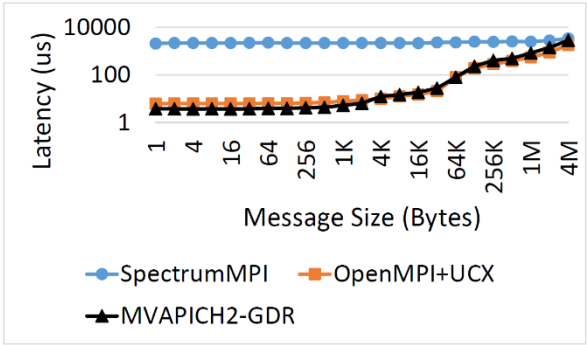
MPI Library	On GPU	On CPU
OpenMPI+UCX	70445	74020
SpectrumMPI	351864	390526



Bandwidth MD MD



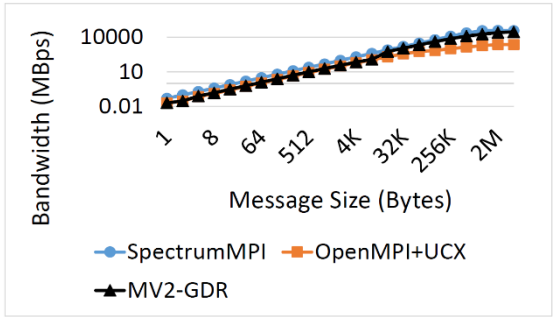
Bi-Bandwidth MD MD



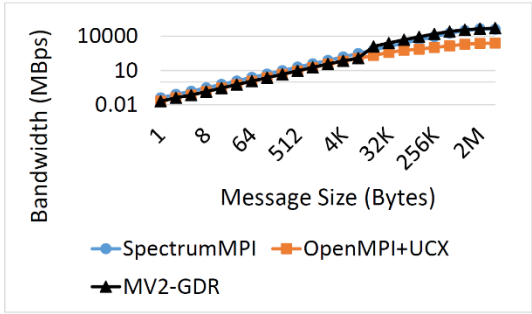
Latency MD MD

OpenPOWER Intra-node Evaluation

- intra-node bibw: OpenMPI needs improvement



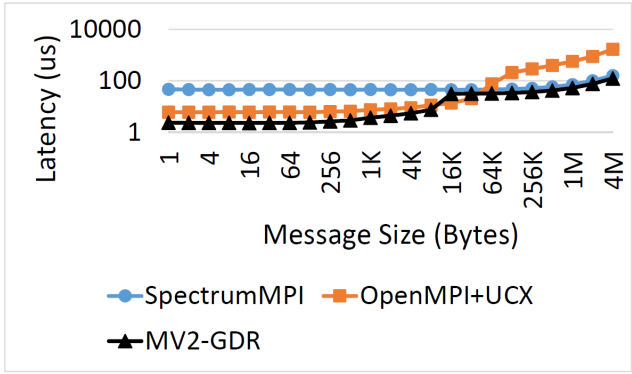
Bandwidth MD MD



Bi-Bandwidth MD MD

Managed Buffer Page Faults

MPI Library	On GPU	On CPU
OpenMPI+UCX	284557	295680
SpectrumMPI	1248	---



Latency MD MD