MVAPICH2-GDR: High-Performance and Scalable CUDA-Aware MPI Library for HPC and AI

GPU Technology Conference (GTC 2019)

by

Dhabaleswar K. (DK) Panda
The Ohio State University
E-mail: panda@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~panda

Hari Subramoni
The Ohio State University
E-mail: subramon@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~subramon
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)

• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
  • Support for Deep Learning
  • Support for OpenPOWER with NVLink
  • Support for Container

• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning

• Conclusions
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 2,975 organizations in 88 countries
  - More than 529,000 (> 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
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)

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

Partner in the upcoming TACC Frontera System

- Empowering Top500 systems for over a decade
MVAPICH2 Release Timeline and Downloads
# Architecture of MVAPICH2 Software Family

## 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

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

### Support for Modern Multi-/Many-core Architectures

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

* Upcoming

---

Network Based Computing Laboratory
GTC 2019
# MVAPICH2 Software Family

## High-Performance Parallel Programming Libraries

<table>
<thead>
<tr>
<th>Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVAPICH2</td>
<td>Support for InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE</td>
</tr>
<tr>
<td>MVAPICH2-X</td>
<td>Advanced MPI features, OSU INAM, PGAS (OpenSHMEM, UPC, UPC++, and CAF), and MPI+PGAS programming models with unified communication runtime</td>
</tr>
<tr>
<td>MVAPICH2-GDR</td>
<td>Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning Applications</td>
</tr>
<tr>
<td>MVAPICH2-Virt</td>
<td>High-performance and scalable MPI for hypervisor and container based HPC cloud</td>
</tr>
<tr>
<td>MVAPICH2-EA</td>
<td>Energy aware and High-performance MPI</td>
</tr>
<tr>
<td>MVAPICH2-MIC</td>
<td>Optimized MPI for clusters with Intel KNC</td>
</tr>
</tbody>
</table>

## Microbenchmarks

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMB</td>
<td>Microbenchmarks suite to evaluate MPI and PGAS (OpenSHMEM, UPC, and UPC++) libraries for CPUs and GPUs</td>
</tr>
</tbody>
</table>

## Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSU INAM</td>
<td>Network monitoring, profiling, and analysis for clusters with MPI and scheduler integration</td>
</tr>
<tr>
<td>OEMT</td>
<td>Utility to measure the energy consumption of MPI applications</td>
</tr>
</tbody>
</table>
• 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 . . .

• NVLink is leading to more paths . . .
• 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:**

```c
MPI_Send(s_devbuf, size, ...);
```

**At Receiver:**

```c
MPI_Recv(r_devbuf, size, ...);
```

*High Performance and High Productivity*
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3.1 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
- Unified memory
MVAPICH2-GDR: Pre-requisites for OpenPOWER & x86 Systems

- MVAPICH2-GDR 2.3.1 requires the following software to be installed on your system:
  1. Mellanox OFED 3.2 and later
  2. NVIDIA Driver 367.48 or later
  3. NVIDIA CUDA Toolkit 7.5 and later
  4. NVIDIA Peer Memory (nv_peer_mem) module to enable GPUDirect RDMA (GDR) support
- Strongly Recommended for Best Performance
  5. GDRCOPY Library by NVIDIA: https://github.com/NVIDIA/gdrcopy
- Comprehensive Instructions can be seen from the MVAPICH2-GDR User Guide:
  - http://mvapich.cse.ohio-state.edu/userguide/gdr/
Simple Installation steps for both systems

Pick the right MVAPICH2-GDR RPM from Downloads page:

- [http://mvapich.cse.ohio-state.edu/downloads/](http://mvapich.cse.ohio-state.edu/downloads/)
- e.g. [http://mvapich.cse.ohio-state.edu/download/mvapich/gdr/2.3/mofed4.5/mvapich2-gdr-mcast.cuda10.0.mofed4.5.gnu4.8.5-2.3-1.el7.x86_64.rpm](http://mvapich.cse.ohio-state.edu/download/mvapich/gdr/2.3/mofed4.5/mvapich2-gdr-mcast.cuda10.0.mofed4.5.gnu4.8.5-2.3-1.el7.x86_64.rpm) (== <mv2-gdr-rpm-name>.rpm)

$ wget http://mvapich.cse.ohio-state.edu/download/mvapich/gdr/2.3/<mv2-gdr-rpm-name>.rpm

**Root Users:**

$ rpm -Uvh --nodeps <mv2-gdr-rpm-name>.rpm

**Non-Root Users:**

$ rpm2cpio <mv2-gdr-rpm-name>.rpm | cpio – id

Contact MVAPICH help list with any questions related to the package

[mvapich-help@cse.ohio-state.edu](mailto:mvapich-help@cse.ohio-state.edu)
MVAPICH2-GDR 2.3.1

- Released on 03/16/2018
- Major Features and Enhancements
  - Based on MVAPICH2 2.3.1
  - Enhanced intra-node and inter-node point-to-point performance for DGX-2 and IBM POWER8 and IBM POWER9 systems
  - Enhanced Allreduce performance for DGX-2 and IBM POWER8/POWER9 systems
  - Enhanced small message performance for CUDA-Aware MPI_Put and MPI_Get
  - Support for PGI 18.10
  - Flexible support for running TensorFlow (Horovod) jobs
  - Add support for Volta (V100) GPU
  - Support for OpenPOWER with NVLink
  - Efficient Multiple CUDA stream-based IPC communication for multi-GPU systems with and without NVLink
  - Leverage Linux Cross Memory Attach (CMA) feature for enhanced host-based communication
  - InfiniBand Multicast (IB-MCAST) based designs for GPU-based broadcast and streaming applications
  - Efficient broadcast designs for Deep Learning applications
Optimized MVAPICH2-GDR Design

**GPU-GPU Inter-node Latency**

- MV2-(NO-GDR)
- MV2-GDR 2.3.1

**GPU-GPU Inter-node Bandwidth**

- MV2-(NO-GDR)
- MV2-GDR-2.3.1

**Graphs:**
- Latency (us) vs. Message Size (Bytes)
- Bandwidth (MB/s) vs. Message Size (Bytes)

**Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores**
- NVIDIA Volta V100 GPU
- Mellanox Connect-X4 EDR HCA
- CUDA 10.0
- Mellanox OFED 4.2 with GPU-Direct-RDMA

**Key Metrics:**
- **MVAPICH2-GDR-2.3.1**
  - 1.85us 11X
- **MV2-(NO-GDR)**
  - 10X

**Additional Information:**
- MVAPICH2-GDR-2.3.1
ROCE and Optimized Collectives Support

- RoCE V1 and V2 support
- RDMA_CM connection support
- CUDA-Aware Collective Tuning
  - Point-point Tuning (available since MVAPICH2-GDR 2.0)
    - Tuned thresholds for the different communication patterns and features
    - Depending on the system configuration (CPU, HCA and GPU models)
  - Tuning Framework for GPU based collectives
    - Select the best algorithm depending on message size, system size and system configuration
    - Support for Bcast and Gather operations for different GDR-enabled systems

- Available since MVAPICH2-GDR 2.2RC1 release
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

### 64K Particles

- Average Time Steps per second (TPS)
- Number of Processes: 4, 8, 16, 32
- MV2 vs. MV2+GDR: 2X speedup

### 256K Particles

- Average Time Steps per second (TPS)
- Number of Processes: 4, 8, 16, 32
- MV2 vs. MV2+GDR: 2X speedup
Application-Level Evaluation (Cosmo) and Weather Forecasting in Switzerland

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


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)

Wilkes GPU Cluster

CSCS GPU cluster

Network Based Computing Laboratory

GTC 2019
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)

• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
  • Support for Deep Learning
  • Support for OpenPOWER with NVLink
  • Support for Container

• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning

• Conclusions
Multi-stream Communication using CUDA IPC on OpenPOWER and DGX-1

- Up to **16% higher** Device to Device (D2D) bandwidth on OpenPOWER + NVLink inter-connect
- Up to **30% higher** D2D bandwidth on DGX-1 with NVLink

Pt-to-pt (D-D) Bandwidth:
Benefits of Multi-stream CUDA IPC Design

- **30% better**
- **16% better**

Available since MVAPICH2-GDR-2.3a
CMA-based Intra-node Host-to-Host Communication Support

- Up to **30% lower** Host-to-Host (H2H) latency and **30% higher** H2H Bandwidth

**INTRA-NODE Pt-to-Pt (H2H) LATENCY**

**INTRA-NODE Pt-to-Pt (H2H) BANDWIDTH**

MVAPICH2-GDR-2.3a
Intel Broadwell (E5-2680 v4 @ 3240 GHz) node – 28 cores
NVIDIA Tesla K-80 GPU, and Mellanox Connect-X4 EDR HCA
CUDA 8.0, Mellanox OFED 4.0 with GPU-Direct-RDMA
Non-contiguous 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 support in MVAPICH2

- Datatypes support in MPI
  - Operate on customized datatypes to improve productivity
  - Enable MPI library to optimize non-contiguous data

  **At Sender:**
  ```c
  MPI_Type_vector (n_blocks, n_elements, stride, old_type, &new_type);
  MPI_Type_commit(&new_type);
  ...
  MPI_Send(s_buf, size, new_type, dest, tag, MPI_COMM_WORLD);
  ```

- Inside MVAPICH2
  - Use datatype specific CUDA Kernels to pack data in chunks
  - Efficiently move data between nodes using RDMA
  - In progress - currently optimizes vector and hindexed datatypes
  - Transparent to the user

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*
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)

• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
  • Support for Deep Learning
  • Support for OpenPOWER with NVLink
  • Support for Container

• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning

• Conclusions
Enhanced Support for Intra-node Unified Memory

- CUDA Unified Memory (UM) => no memory pin down
  - No IPC support for intra-node communication
  - No GDR support for Inter-node communication

- Initial and basic support in MVAPICH2-GDR
  - For both intra- and inter-nodes use “pipeline through” host memory

- Enhance intra-node UM to use IPC
  - Double buffering pair-wise IPC-based scheme
  - Brings IPC performance to UM
  - High performance and high productivity

- Available since MVAPICH2-GDR 2.2RC1

K. Hamidouche, A. Awan, A. Venkatesh, and D. K Panda, CUDA M3: Designing Efficient CUDA Managed Memory-aware MPI by Exploiting GDR and IPC, HiPC ‘16
Characterizing Unified Memory aware MPI on modern GPUs

- Improved UM support in Pascal & Volta GPUs through:
  - Advanced GPU page fault engines
  - `cudaMemPrefetch` and `cudaMemAdvise` APIs provide more control for UM data placement
- Are the UM designs developed during Kepler era still valid?
- Carried out an in-depth characterization
- Our characterization studies show:
  - The UM designs from Kepler era are still valid
  - They are 4.2X and 2.8X better in latency compared to MVAPICH2-GDR and Open MPI

K. V. Manian, A. Awan, A. Ruhela, C. Chu, H. Subramoni and D. K Panda, Characterizing CUDA Unified Memory (UM)-Aware MPI Designs on Modern GPU Architectures, GPGPU ‘19 Workshop, in conjunction with ASPLOS ‘19, April ‘19
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)

• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
  • Support for Deep Learning
  • Support for OpenPOWER with NVLink
  • Support for Container

• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning

• Conclusions
Streaming Applications

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

Data Source

Real-time streaming

HPC resources for real-time analytics

Data streaming-like broadcast operations

Worker

- CPU
- GPU
- GPU

Sender
Hardware Multicast-based Broadcast

- For GPU-resident data, using
  - GPUDirect RDMA (GDR)
  - InfiniBand Hardware Multicast (IB-MCAST)

- Overhead
  - IB UD limit
  - GDR limit

Available since MVAPICH2-GDR 2.3a

Streaming Benchmark @ CSCS (88 GPUs)

- **IB-MCAST + GDR + Topology-aware IPC-based schemes**
  - Up to 58% and 79% reduction for small and large messages

Application-based Evaluation: CUDA-Aware CNTK

- @ RI2 cluster, 16 GPUs, 1 GPU/node
  - CUDA-Aware Microsoft Cognitive Toolkit (CA-CNTK) \[^2\]
    - MV2-GDR-Knomial
    - MV2-GDR-Ring
    - MV2-MCAST-GDR-Opt


- Reduces up to 24%, 16% and 18% of latency for AlexNet, VGG and ResNet models
- Higher improvement can be observed for larger system sizes

Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)

• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
    • Support for Deep Learning
    • Support for OpenPOWER with NVLink
    • Support for Container

• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning

• Conclusions
Deep Learning: New Challenges for Runtimes

- **Scale-up:** Intra-node Communication
  - Many improvements like:
    - NVIDIA cuDNN, cuBLAS, NCCL, etc.
    - CUDA 9 Co-operative Groups

- **Scale-out:** Inter-node Communication
  - DL Frameworks – most are optimized for single-node only
  - Distributed (Parallel) Training is an emerging trend
    - OSU-Caffe – MPI-based
    - Microsoft CNTK – MPI/NCCL2
    - Google TensorFlow – gRPC-based/MPI/NCCL2
    - Facebook Caffe2 – Hybrid (NCCL2/Gloo/MPI)
Data Parallel Deep Learning and MPI Collectives

- **Major MPI Collectives** involved in Designing distributed frameworks
- **MPI_Bcast** – required for DNN parameter exchange
- **MPI_Reduce** – needed for gradient accumulation from multiple solvers
- **MPI_Allreduce** – use just one Allreduce instead of Reduce and Broadcast

MVAPICH2-GDR: Allreduce Comparison with Baidu and OpenMPI

- 16 GPUs (4 nodes) MVAPICH2-GDR vs. Baidu-Allreduce and OpenMPI 3.0

*Available since MVAPICH2-GDR 2.3a*
MVAPICH2-GDR vs. NCCL2 – Allreduce Operation

- Optimized designs since MVAPICH2-GDR 2.3 offer better/comparable performance for most cases
- \texttt{MPI\_Allreduce} (MVAPICH2-GDR) vs. \texttt{nccl\_Allreduce} (NCCL2) on 16 GPUs

Platform: Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 1 K-80 GPUs, and EDR InfiniBand Inter-connect
Optimized designs in MVAPICH2-GDR 2.3.1 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
Scalable TensorFlow using Horovod, MPI, and NCCL

- Efficient Allreduce is crucial for Horovod’s overall training performance
  - Both MPI and NCCL designs are available
- We have evaluated Horovod extensively and compared across a wide range of designs using gRPC and gRPC extensions
- MVAPICH2-GDR achieved up to 90% scaling efficiency for ResNet-50 Training on 64 Pascal GPUs

https://arxiv.org/abs/1810.11112
Distributed Training with TensorFlow and MVAPICH2-GDR

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

**Scaling Efficiency**

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

**Graphs:**

- Left graph: Image per second vs. Number of GPUs.
  - 1 GPU: 500, 2 GPUs: 1000, 4 GPUs: 2000, 8 GPUs: 3000.
  - NCCL-2.3 vs. MVAPICH2-GDR-Next.
  - MVAPICH2-GDR-Next is 7.5% higher than NCCL-2.3.

- Right graph: Scaling Efficiency (%) vs. Number of GPUs.
  - 1 GPU: 75, 2 GPUs: 80, 4 GPUs: 85, 8 GPUs: 90.
  - NCCL-2.3 vs. MVAPICH2-GDR-Next.

**Platform:**

Nvidia DGX-2 system (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2
OSU-Caffe: Scalable Deep Learning

- Benefits and Weaknesses
  - Multi-GPU Training within a single node
  - Performance degradation for GPUs across different sockets
  - Limited Scale-out
- 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/

Support on OPENPOWER will be available soon
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)

• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
  • Support for Deep Learning
  • Support for OpenPOWER with NVLink
  • Support for Container

• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning

• Conclusions
Point-to-Point Host-level Performance on OpenPOWER

Intra-node Latency

- MVAPICH2-GDR 2.3.1
- ~0.5 μs

Intra-node Bandwidth

- MVAPICH2-GDR 2.3.1
- ~30GB/s

Intra-node Bi-directional Bandwidth

- MVAPICH2-GDR 2.3.1
- ~60GB/s

Inter-node Latency

- MVAPICH2-GDR 2.3.1
- ~2.3 μs

Inter-node Bandwidth

- MVAPICH2-GDR 2.3.1
- ~12GB/s

Inter-node Bi-directional Bandwidth

- MVAPICH2-GDR 2.3.1
- ~24GB/s

Platform: OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA
Device-to-Device Performance on OpenPOWER (NVLink2 + Volta)

**INTRA-NODE LATENCY (SMALL)**

- Intra-Socket: 5.36 us (without GDRCopy)

**INTRA-NODE LATENCY (LARGE)**

- Intra-Socket: 5.36 us (without GDRCopy)

**INTER-NODE LATENCY (SMALL)**

- Inter-node Latency: 5.66 us (without GDRCopy)

**INTER-NODE LATENCY (LARGE)**

- Inter-node Latency: 5.66 us (without GDRCopy)

**INTRA-NODE BANDWIDTH**

- Intra-node Bandwidth: 70.4 GB/sec for 128MB (via NVLINK2)

**INTER-NODE BANDWIDTH**

- Inter-node Bandwidth: 23.7 GB/sec (2 port EDR)

Platform: OpenPOWER (POWER9-ppc64le) nodes equipped with a dual-socket CPU, 4 Volta V100 GPUs, and 2 port EDR InfiniBand Interconnect

Available since MVAPICH2-GDR 2.3a
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)
  • Current Features
    • Multi-stream Communication for IPC
    • CMA-based Intra-node Host-to-Host Communication Support
    • Maximal Overlap in MPI Datatype Processing
    • Efficient Support for Managed Memory
    • Streaming Support with InfiniBand Multicast and GDR
    • Support for Deep Learning
    • Support for OpenPOWER with NVLink
    • Support for Container
  • Upcoming Features
    • CMA-based Intra-node Collective Communication Support
    • XPMEM-based Collective Communication Support
    • Optimized Datatype Processing
    • Out-of-core processing for Deep Learning
• Conclusions
Container Support

- Increasing trend to provide container support for MPI Libraries
  - Ease of build
  - Portability
  - Reproducibility
- MVAPICH2-GDR 2.3.1 provides container (Docker) support
- More details are available in the MVAPICH2-GDR User Guide
  - [http://mvapich.cse.ohio-state.edu/userguide/gdr/](http://mvapich.cse.ohio-state.edu/userguide/gdr/)
- Synergistic with the HPC-Container-Maker and hpccm efforts by NVIDIA
  - [https://github.com/NVIDIA/hpc-container-maker](https://github.com/NVIDIA/hpc-container-maker)
MVAPICH2-GDR on Container with Negligible Overhead

**GPU-GPU Inter-node Latency**

![Graph showing latency vs message size for Docker and Native, with a notable overhead for Docker at large message sizes.]

**GPU-GPU Inter-node Bandwidth**

![Graph showing bandwidth vs message size for Docker and Native, with a significant advantage for Native at large message sizes.]

**GPU-GPU Inter-node Bi-Bandwidth**

![Graph showing bi-bandwidth vs message size for Docker and Native, with a clear advantage for Native at large message sizes.]

**System Specifications**

- **MVAPICH2-GDR-2.3.1**
- **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**
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)
• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
  • Support for Deep Learning
  • Support for OpenPOWER with NVLink
  • Support for Container

• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning

• Conclusions
Scalable Host-based Collectives on OpenPOWER with CMA (Intra-node Reduce & AlltoAll)

Up to 5X and 3x performance improvement by MVAPICH2 for small and large messages respectively.
Scalable Host-based Collectives on OpenPOWER with CMA (Multi-node, Reduce & Alltoall)

Up to 12.4X and 8.5X performance improvement by MVAPICH2 for small and large messages respectively.
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)
• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
  • Support for Deep Learning
  • Support for OpenPOWER with NVLink
  • Support for Container
• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning
• Conclusions
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

Benefits of XPMEM based MPI_Bcast

- **28 MPI Processes** on single dual-socket Broadwell E5-2680v4, 2x14 core processor
Benefits of XPMEM based MPI_Scatter

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

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

Latency (us) vs. Message Size (bytes)

- Intel MPI 2018
- OpenMPI 3.0.1
- MV2X-2.3rc1 (CMA Coll)
- MV2X-2.3rc2 (XPMEM Coll)
Optimized All-Reduce with XPMEM

- Optimized MPI All-Reduce Design in MVAPICH2
  - Up to 2X performance improvement over Spectrum MPI and 4X over OpenMPI for intra-node

Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch
Application-Level Benefits of XPMEM-Based Collectives

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

- 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
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)

• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
  • Support for Deep Learning
  • Support for OpenPOWER with NVLink
  • Support for Container

• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning

• Conclusions
MVAPICH2-GDR: Enhanced Derived Datatype Processing

- Kernel-based and GDRCOPY-based one-shot packing for inter-socket and inter-node communication
- Zero-copy (packing-free) for GPUs with peer-to-peer direct access over PCIe/NVLink

**GPU-based DDTBench mimics MILC communication kernel**

**Communication Kernel of COSMO Model**

- **OpenMPI 4.0.0**
- **MVAPICH2-GDR 2.3**
- **MVAPICH2-GDR-Next**

**Platform: Nvidia DGX-2 system**

(16 NVIDIA Volta GPUs connected with NVSwitch), CUDA 9.2

**Platform: Cray CS-Storm**

(8 NVIDIA Tesla K80 GPUs per node), CUDA 8.0

**Improved ~10X**
Outline

- Overview of the MVAPICH2 Project
- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- Current Features
  - Multi-stream Communication for IPC
  - CMA-based Intra-node Host-to-Host Communication Support
  - Maximal Overlap in MPI Datatype Processing
  - Efficient Support for Managed Memory
  - Streaming Support with InfiniBand Multicast and GDR
  - Support for Deep Learning
  - Support for OpenPOWER with NVLink
  - Support for Container
- Upcoming Features
  - CMA-based Intra-node Collective Communication Support
  - XPMEM-based Collective Communication Support
  - Optimized Datatype Processing
  - Out-of-core processing for Deep Learning
- Conclusions
Scalability and Large (Out-of-core) Models?

• Large DNNs cannot be trained on GPUs due to memory limitation!
  – ResNet-50 for Image Recognition but current frameworks can only go up to a small batch size of 45
  – Next generation models like Neural Machine Translation (NMT) are ridiculously large, consists of billions of parameters, and require even more memory
  – Can we design Out-of-core DNN training support using new software features in CUDA 8/9 and hardware mechanisms in Pascal/Volta GPUs?

• General intuition is that managed allocations “will be” slow!
  – The proposed framework called **OC-Caffe (Out-of-Core Caffe)** shows the potential of managed memory designs that can provide performance with negligible/no overhead.

• OC-Caffe-Opt: up to **80% better** than Intel-optimized CPU Caffe for ResNet-50 training on the Volta V100 GPU with CUDA9 and CUDNN7

Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)
• Current Features
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Host-to-Host Communication Support
  • Maximal Overlap in MPI Datatype Processing
  • Efficient Support for Managed Memory
  • Streaming Support with InfiniBand Multicast and GDR
  • Support for Deep Learning
  • Support for OpenPOWER with NVLink
  • Support for Container
• Upcoming Features
  • CMA-based Intra-node Collective Communication Support
  • XPMEM-based Collective Communication Support
  • Optimized Datatype Processing
  • Out-of-core processing for Deep Learning
• Conclusions
Conclusions

- MVAPICH2-GDR MPI library optimizes MPI communication on InfiniBand and RoCE (V1 and V2) clusters with GPUs on both x86 and OpenPOWER platforms (including NVLink)
- 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
- Allows flexible solutions for streaming applications with GPUs
- Provides optimized solutions for both HPC and High-Performance Deep Learning (HiDL) frameworks and applications
- Upcoming releases will be supporting advanced designs
Please join us for more events..

<table>
<thead>
<tr>
<th>Monday, March 18</th>
<th>Tuesday, March 19</th>
<th>Wednesday, March 20</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Research Poster</strong></td>
<td><strong>Talk</strong></td>
<td><strong>Instructor-Led Training</strong></td>
</tr>
<tr>
<td>1. P9243 - Exploiting CUDA Unified Memory for Efficient Out-of-Core DNN Training</td>
<td>S9476 - MVAPICH2-GDR: High-Performance and Scalable CUDA-Aware MPI Library for HPC and AI</td>
<td>L9121 - How to Boost the Performance of HPC/AI Applications Using MVAPICH2 Library</td>
</tr>
<tr>
<td>2. P9242 - Exploiting GPUDirect Technology and Hardware Multicast for Streaming and Deep Learning Applications</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SJCC Upper Concourse</strong></td>
<td><strong>SJCC Room 211A</strong></td>
<td><strong>SJCC Room LL21D</strong></td>
</tr>
<tr>
<td>06:00 PM - 08:00 PM</td>
<td>03:00 PM - 03:50 PM</td>
<td>08:00 AM - 10:00 AM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Personnel Acknowledgments

## Current Students (Graduate)
- A. Awan (Ph.D.)
- M. Bayatpour (Ph.D.)
- S. Chakraborthy (Ph.D.)
- C.-H. Chu (Ph.D.)
- S. Guganani (Ph.D.)

## Current Students (Undergraduate)
- J. Hashmi (Ph.D.)
- A. Jain (Ph.D.)
- K. S. Khorassani (Ph.D.)
- P. Kousha (Ph.D.)
- D. Shankar (Ph.D.)

## Current Students (Undergraduate)
- V. Gangal (B.S.)
- M. Haupt (B.S.)
- N. Sarkauskas (B.S.)
- A. Yeretzian (B.S.)

## Past Students
- A. Augustine (M.S.)
- P. Balaji (Ph.D.)
- R. Biswas (M.S.)
- S. Bhagvat (M.S.)
- A. Bhat (M.S.)
- D. Buntinas (Ph.D.)
- L. Chai (Ph.D.)
- B. Chandrasekharan (M.S.)
- N. Dandapanthula (M.S.)
- V. Dhanraj (M.S.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- W. Huang (Ph.D.)
- W. Jiang (M.S.)
- J. Jose (Ph.D.)
- S. Kini (M.S.)
- M. Koop (Ph.D.)
- K. Kulkarni (M.S.)
- R. Kamar (M.S.)
- S. Krishnamoorthy (M.S.)
- K. Kandalla (Ph.D.)
- M. Li (Ph.D.)
- P. Lai (M.S.)
- J. Liu (Ph.D.)
- M. Luo (Ph.D.)
- A. Mamidala (Ph.D.)
- G. Marsh (M.S.)
- V. Meshram (M.S.)
- A. Moody (M.S.)
- S. Naravula (Ph.D.)
- R. Noronha (Ph.D.)
- X. Ouyang (Ph.D.)
- S. Pai (M.S.)
- S. Potluri (Ph.D.)
- R. Rajachandrasekar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Sur (Ph.D.)
- H. Subramoni (Ph.D.)
- K. Vaidyanathan (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)
- J. Zhang (Ph.D.)

## Past Research Scientist
- K. Hamidouche
- S. Sur

## Past Post-Docs
- D. Banerjee
- X. Besseron
- H.-W. Jin
- J. Lin
- M. Luo
- E. Mancini
- S. Marcarelli
- J. Vienne
- H. Wang

## Current Research Asst. Professor
- X. Lu

## Current Post-doc
- A. Ruhela
- K. Manian

## Current Research Scientist
- H. Subramoni

## Current Research Specialist
- J. Smith

## Past Research Scientist
- D. Bureddy
- J. Perkins

## Past Programmers
- M. Arnold

## Past Research Specialist
- M. Arnold

Network Based Computing Laboratory

GTC 2019
Multiple Positions Available in My Group

• Looking for Bright and Enthusiastic Personnel to join as
  – PhD Students
  – Post-Doctoral Researchers
  – MPI Programmer/Software Engineer
  – Hadoop/Big Data Programmer/Software Engineer
  – Deep Learning and Cloud Programmer/Software Engineer

• If interested, please send an e-mail to panda@cse.ohio-state.edu
Thank You!

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