MVAPICH2-GDR: Pushing the Frontier of HPC and Deep Learning

Talk at Mellanox booth (SC ‘19)

by

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

- Overview of the MVAPICH2 Project
- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What’s new with MVAPICH2-GDR
- High-Performance Deep Learning (HiDL) with MVAPICH2-GDR
- 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 (Supercomputing 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 3,050 organizations in 89 countries
- More than 615,000 (> 0.6 million) downloads from the OSU site directly
- Empowering many TOP500 clusters (Nov ‘19 ranking)
  - 3rd, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China
  - 5th, 448,448 cores (Frontera) at TACC
  - 8th, 391,680 cores (ABCI) in Japan
  - 14th, 570,020 cores (Neurion) in South Korea and many others
- Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
- Partner in the 5th ranked TACC Frontera System

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

Empowering Top500 systems for over a decade
MVAPICH2 Release Timeline and Downloads

Number of Downloads

Timeline

MV0.9.4
MV2.9.0
MV2 0.9.8
MV2 1.0
MV21.0
MV21.0.3
MV 1.1
MV2 1.1.4
MV2 1.4
MV2 1.5
MV2 1.6
MV2 1.7
MV2 1.8
MV2 1.9
MV2-GDR 2.0b
MV2-Virt 2.2
MV2-MIC 2.0
OSU INAM 0.9.3
MV2-X 2.3
MV2-GDR 2.3.2
MV2 Virt 2.2
MV2-MIC 2.0
MV2-GDR 2.3.2
MV2-Azure 2.3.2
MV2-AWS 2.3
### Architecture of MVAPICH2 Software Family (HPC and DL)

#### 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, Omni-Path, Elastic Fabric Adapter)

- Transport Protocols: RC, SRD, UD, DC
- Modern Features: UMR, ODP, SR-IOV, Multi Rail

**Support for Modern Multi-/Many-core Architectures** (Intel-Xeon, OpenPOWER, Xeon-Phi, ARM, NVIDIA GPGPU)

- Transport Mechanisms: Shared Memory, CMA, IVSHMEM, XPMEM
- Modern Features: Optane*, NVLink, CAPI*

* Upcoming
# MVAPICH2 Software Family

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI with IB, iWARP, Omni-Path, and RoCE</td>
<td>MVAPICH2</td>
</tr>
<tr>
<td>Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE</td>
<td>MVAPICH2-X</td>
</tr>
<tr>
<td>MPI with IB, RoCE &amp; GPU and Support for Deep Learning</td>
<td>MVAPICH2-GDR</td>
</tr>
<tr>
<td>HPC Cloud with MPI &amp; IB</td>
<td>MVAPICH2-Virt</td>
</tr>
<tr>
<td>Energy-aware MPI with IB, iWARP and RoCE</td>
<td>MVAPICH2-EA</td>
</tr>
<tr>
<td>MPI Energy Monitoring Tool</td>
<td>OEMT</td>
</tr>
<tr>
<td>InfiniBand Network Analysis and Monitoring</td>
<td>OSU INAM</td>
</tr>
<tr>
<td>Microbenchmarks for Measuring MPI and PGAS Performance</td>
<td>OMB</td>
</tr>
</tbody>
</table>
• Connected as PCIe devices – Flexibility but Complexity

- Memory buffers
  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
- Unified memory
MVAPICH2-GDR 2.3.2

- Released on 08/08/2019
- Major Features and Enhancements
  - Based on MVAPICH2 2.3.1
  - Support for CUDA 10.1
  - Support for PGI 19.x
  - Enhanced intra-node and inter-node point-to-point performance
  - Enhanced MPI_Allreduce performance for DGX-2 system
  - Enhanced GPU communication support in MPI_THREAD_MULTIPLE mode
  - Enhanced performance of datatype support for GPU-resident data
  - Zero-copy transfer when P2P access is available between GPUs through NVLink/PCIe
  - Enhanced GPU-based point-to-point and collective tuning
    - OpenPOWER systems such as ORNL Summit and LLNL Sierra ABCI system @AIST, Owens and Pitzer systems @Ohio Supercomputer Center
  - Scaled Allreduce to 24,576 Volta GPUs on Summit
  - 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
  - Flexible support for running TensorFlow (Horovod) jobs
Optimized MVAPICH2-GDR Design

**GPU-GPU Inter-node Latency**

<table>
<thead>
<tr>
<th>Message Size (Bytes)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1K</th>
<th>2K</th>
<th>4K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency (us)</td>
<td>1.85</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>40</td>
<td>80</td>
<td>160</td>
<td>320</td>
<td>640</td>
<td>1280</td>
<td>2560</td>
<td>5120</td>
<td>10240</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Message Size (Bytes)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1K</th>
<th>2K</th>
<th>4K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth (MB/s)</td>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>50</td>
<td>100</td>
<td>200</td>
<td>400</td>
<td>800</td>
<td>1600</td>
<td>3200</td>
<td>6400</td>
<td>12800</td>
<td>25600</td>
</tr>
</tbody>
</table>

**GPU-GPU Inter-node Bandwidth**

<table>
<thead>
<tr>
<th>Message Size (Bytes)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1K</th>
<th>2K</th>
<th>4K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth (MB/s)</td>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>50</td>
<td>100</td>
<td>200</td>
<td>400</td>
<td>800</td>
<td>1600</td>
<td>3200</td>
<td>6400</td>
<td>12800</td>
<td>25600</td>
</tr>
</tbody>
</table>

**Products and Technologies**

- 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

### 64K Particles

<table>
<thead>
<tr>
<th>Number of Processes</th>
<th>MV2</th>
<th>MV2+GDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1000</td>
<td>2000</td>
</tr>
<tr>
<td>8</td>
<td>1500</td>
<td>2250</td>
</tr>
<tr>
<td>16</td>
<td>2000</td>
<td>2800</td>
</tr>
<tr>
<td>32</td>
<td>2500</td>
<td>3750</td>
</tr>
</tbody>
</table>

### 256K Particles

<table>
<thead>
<tr>
<th>Number of Processes</th>
<th>MV2</th>
<th>MV2+GDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>500</td>
<td>250</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>5000</td>
</tr>
<tr>
<td>16</td>
<td>1500</td>
<td>7500</td>
</tr>
<tr>
<td>32</td>
<td>2000</td>
<td>10000</td>
</tr>
</tbody>
</table>

**Average Time Steps per second (TPS)**

- **MV2**
- **MV2+GDR**

2X increase in performance.
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)
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)
• What’s new with MVAPICH2-GDR
  • Multi-stream Communication for IPC
  • CMA-based Intra-node Communication Support
  • Support for OpenPower and NVLink with GDRCOPY2
  • Maximal overlap in MPI Datatype Processing
• High-Performance Deep Learning (HiDL) with MVAPICH2-GDR
• 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

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

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

---

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

- **MV2-GDR (w/out CMA)**
- **MV2-GDR (w/ CMA)**

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

- **MV2-GDR (w/out CMA)**
- **MV2-GDR (w/ CMA)**

---

**MVAPICH2-GDR-2.3.2**

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
Scalable Host-based Collectives on OpenPOWER (Intra-node Reduce & AlltoAll)

Reduce

- MVAPICH2-GDR
- SpectrumMPI-10.1.0.2
- OpenMPI-3.0.0

Up to 5X and 3x performance improvement by MVAPICH2 for small and large messages respectively

Alltoall

- MVAPICH2-GDR
- SpectrumMPI-10.1.0.2
- OpenMPI-3.0.0

Network Based Computing Laboratory

Mellanox Booth (SC ‘19)
D-to-D Performance on OpenPOWER w/ GDRCopy (NVLink2 + Volta)

Intra-Node Latency (Small Messages)
Intra-Node Latency (Large Messages)
Intra-Node Bandwidth

Intra-node Latency: 0.76 us (with GDRCopy)
Intra-node Bandwidth: 65.48 GB/sec for 4MB (via NVLINK2)

Inter-Node Latency (Small Messages)
Inter-Node Latency (Large Messages)
Inter-Node Bandwidth

Inter-node Latency: 2.18 us (with GDRCopy 2.0)
Inter-node Bandwidth: 23 GB/sec for 4MB (via 2 Port EDR)

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

Network Based Computing Laboratory
Mellanox Booth (SC ’19)
D-to-H & H-to-D Performance on OpenPOWER w/ GDRCopy (NVLink2 + Volta)

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

**Intra-node D-H Latency:** 0.49 us (with GDRCopy)

**H-D INTRA-NODE LATENCY (SMALL)**

Intra-node H-D Latency: 0.49 us (with GDRCopy 2.0)

**H-D INTRA-NODE LATENCY (LARGE)**

Intra-node H-D Bandwidth: 26.09 GB/sec for 2MB (via NVLINK2)

**D-H INTRA-NODE LATENCY (SMALL)**

**D-H INTRA-NODE LATENCY (LARGE)**

Intra-node D-H Bandwidth: 16.70 GB/sec for 2MB (via NVLINK2)

**D-H INTRA-NODE BW**

**H-D INTRA-NODE BW**
Managed Memory Performance (OpenPOWER Intra-node)

Latency vs Message Size (Bytes)
- SpectrumMPI
- OpenMPI+UCX
- MV2-GDR

Bandwidth vs Message Size (Bytes)
- SpectrumMPI
- OpenMPI+UCX
- MV2-GDR

Bi-Bandwidth vs Message Size (Bytes)
- SpectrumMPI
- OpenMPI+UCX
- MV2-GDR
**MVAPICH2 with SHARP Support (Preliminary Results)**

**GPU AllReduce w/ SHARP Support – 2 nodes**

- MVAPICH2-GDR-Next
- MVAPICH2-GDR-Next w/ SHARP

![Graph showing latency for GPU AllReduce with SHARP Support – 2 nodes.](image)

**GPU AllReduce w/ SHARP Support – 4 nodes**

- MVAPICH2-GDR-Next
- MVAPICH2-GDR-Next w/ SHARP

![Graph showing latency for GPU AllReduce with SHARP Support – 4 nodes.](image)

**HOST AllReduce w/ SHARP Support – 2 nodes**

- MVAPICH2-2.3.2
- MVAPICH2-2.3.2 w/ SHARP

![Graph showing latency for HOST AllReduce with SHARP Support – 2 nodes.](image)

**HOST AllReduce w/ SHARP Support – 8 nodes**

- MVAPICH2-2.3.2
- MVAPICH2-2.3.2 w/ SHARP

![Graph showing latency for HOST AllReduce with SHARP Support – 8 nodes.](image)

**Platform:** OpenPOWER (POWER9-ppc64le) nodes equipped with a dual-socket CPU, 4 Volta V100 GPUs, and 2port EDR InfiniBand Interconnect
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
Waste of computing resources on CPU and GPU

**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*
## Application: COMB

Run Scripts pushed to COMB Github repo: [https://github.com/LLNL/Comb/pull/2](https://github.com/LLNL/Comb/pull/2)

<table>
<thead>
<tr>
<th>16 GPUs on POWER9 system (test Comm mpi Mesh cuda Device Buffers mpi_type)</th>
<th>pre-comm</th>
<th>post-rekv</th>
<th>post-send</th>
<th>wait-rekv</th>
<th>wait-send</th>
<th>post-comm</th>
<th>start-up</th>
<th>test-comm</th>
<th>bench-comm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectrum MPI 10.3</td>
<td>0.0001</td>
<td>0.0000</td>
<td>1.6021</td>
<td>1.7204</td>
<td>0.0112</td>
<td>0.0001</td>
<td>0.0004</td>
<td>7.7383</td>
<td>83.6229</td>
</tr>
<tr>
<td>MVAPICH2-GDR 2.3.2</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0862</td>
<td>0.0871</td>
<td>0.0018</td>
<td>0.0001</td>
<td>0.0009</td>
<td>0.3558</td>
<td>4.4396</td>
</tr>
<tr>
<td>MVAPICH2-GDR 2.3.3 (Upcoming)</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0030</td>
<td>0.0032</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0009</td>
<td>0.0133</td>
<td>0.1602</td>
</tr>
</tbody>
</table>

- Improvements due to enhanced support for GPU-kernel based packing/unpacking routines
**Application: HYPRE - BoomerAMG**

**HYPRE - BoomerAMG**

![](chart.png)

- **Spectrum-MPI 10.3.0.1**
- **MVAPICH2-GDR 2.3.2**

**RUN MVAPICH2-GDR 2.3.2:**

```bash
export MV2_USE_CUDA=1 MV2_USE_GDRCOPY=0 MV2_USE_RDMA_CM=0
export MV2_USE_GPUDIRECT_LOOPBACK=0 MV2_HYBRID_BINDING_POLICY=spread MV2_IBA_HCA=mlx5_0:mlx5_3
OMP_NUM_THREADS=20 lrun -n 128 -N 32 mpibind ./ij -P 8 4 4 -n 50 50 50 -pmis -Pmx 8 -keepT 1 -rlx 18
```

**RUN Spectrum-MPI 10.3.0.1:**

```bash
OMP_NUM_THREADS=20 lrun -n 128 -N 32 --smpiargs "-gpu --disable_gdr" mpibind ./ij -P 8 4 4 -n 50 50 50 -pmis -Pmx 8 -keepT 1 -rlx 18
```
Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)
• What’s new with MVAPICH2-GDR
• High-Performance Deep Learning (HiDL) with MVAPICH2-GDR
  • Benefits of CUDA-Aware MPI with TensorFlow
  • Optimized Collectives for Deep Learning
  • Out-of-core DNN Training
• Conclusions
Data Parallel Training with TensorFlow (TF)

- Need to understand several options currently available

- gRPC (official support)
  - Open-source – can be enhanced by others
  - Accelerated gRPC (add RDMA to gRPC)

- gRPC+X
  - Use gRPC for bootstrap and rendezvous
  - *Actual communication is in “X”*
  - X → MPI, Verbs, GPUDirect RDMA (GDR), etc.

- No-gRPC
  - Baidu – the first one to use MPI Collectives for TF
  - Horovod – Use NCCL, or MPI, or any other future library (e.g. IBM DDL support recently added)

---

Exploiting CUDA-Aware MPI for TensorFlow (Horovod)

- 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**


**MVAPICH2-GDR vs. NCCL2: Allreduce Operation**

- Optimized designs in MVAPICH2-GDR 2.3 offer better/comparable performance for most cases.
- `MPI_Allreduce` (MVAPICH2-GDR) vs. `ncclAllreduce` (NCCL2) on 16 GPUs.

---

**Platform:** Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 1 K-80 GPUs, and EDR InfiniBand Inter-connect.

---

- **Latency (us)**
  - **Message Size (Bytes):** 4, 8, 16, 32, 64, 128, 256, 512, 1K, 2K, 4K, 8K, 16K, 32K, 64K
  - **Latency (us):** 1000, 100, 10, 1
  - **~3X better**

- **Latency (us)**
  - **Message Size (Bytes):** 128K, 256K, 512K, 1M, 2M, 4M, 8M, 16M, 32M, 64M, 128M, 256M
  - **Latency (us):** 100000, 10000, 1000, 100, 10, 1
  - **~1.2X better**

---

Network Based Computing Laboratory

Mellanox Booth (SC ’19)
MVAPICH2-GDR vs. NCCL2: Allreduce Optimization (DGX-2)

- Optimized designs in upcoming MVAPICH2-GDR offer better performance for most cases
- MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) on a DGX-2 machine

Platform: Nvidia DGX-2 system @ PSC (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2
Optimized designs in MVAPICH2-GDR offer better performance for most cases

MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) up to 1,536 GPUs

Platform: Dual-socket IBM POWER9 CPU, 6 NVIDIA Volta V100 GPUs, and 2-port InfiniBand EDR Interconnect
Distributed Training with TensorFlow and MVAPICH2-GDR on Summit

- ResNet-50 Training using TensorFlow benchmark on SUMMIT -- 1536 Volta GPUs!
- 1,281,167 (1.2 mil.) images
- Time/epoch = 3.6 seconds
- Total Time (90 epochs) = 3.6 x 90 = 332 seconds = 5.5 minutes!

*We observed errors for NCCL2 beyond 96 GPUs

**Platform:** The Summit Supercomputer (#1 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 9.2

**ImageNet-1k has 1.2 million images**

**MVAPICH2-GDR reaching ~0.35 million images per second for ImageNet-1k!**
New Benchmark for Image Segmentation on Summit

- Near-linear scaling may be achieved by tuning Horovod/MPI
  - Optimizing MPI/Horovod towards large message sizes for high-resolution images
- Develop a generic Image Segmentation benchmark
- Tuned DeepLabV3+ model using the benchmark and Horovod, up to 1.3X better than default

*Anthony et al., “Scaling Semantic Image Segmentation using Tensorflow and MVAPICH2-GDR on HPC Systems” (Submission under review)
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/
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: Neural Machine Translation (NMT)
    - Ridiculously large (billions of parameters),
    - Will require even more memory!
  - Can we exploit 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

---

HyPar-Flow (HF): Hybrid Parallelism for TensorFlow

- CPU based results
  - AMD EPYC
  - Intel Xeon
- Excellent speedups for
  - VGG-19
  - ResNet-110
  - ResNet-1000 (1k layers)
- Able to train “future” models
  - E.g. ResNet-5000 (a synthetic 5000-layer model we benchmarked)

110x speedup on 128 Intel Xeon Skylake nodes (TACC Stampede2 Cluster)

Outline

• Overview of the MVAPICH2 Project
• MVAPICH2-GPU with GPUDirect-RDMA (GDR)
• What’s new with MVAPICH2-GDR
• High-Performance Deep Learning (HiDL) with MVAPICH2-GDR
• Conclusions
Conclusions

• MVAPICH2-GDR Library provides optimized MPI communication on InfiniBand and RoCE clusters with GPUs
• Supports both X86 and OpenPower with NVLink
• Takes advantage of CUDA features like IPC and GPUDirect RDMA families
• Allows flexible solutions for streaming applications with GPUs
• Provides optimized solutions (scale-up and scale-out) for High-Performance Deep Learning
Commercial Support for MVAPICH2, HiBD, and HiDL Libraries

- Supported through X-ScaleSolutions (http://x-scalesolutions.com)
- Benefits:
  - Help and guidance with installation of the library
  - Platform-specific optimizations and tuning
  - Timely support for operational issues encountered with the library
  - Web portal interface to submit issues and tracking their progress
  - Advanced debugging techniques
  - Application-specific optimizations and tuning
  - Obtaining guidelines on best practices
  - Periodic information on major fixes and updates
  - Information on major releases
  - Help with upgrading to the latest release
  - Flexible Service Level Agreements
- Support provided to Lawrence Livermore National Laboratory (LLNL) for the last two years
Multiple Events at SC ‘19

- Presentations at OSU and X-Scale Booth (#2094)
  - Members of the MVAPICH, HiBD and HiDL members
  - External speakers

- Presentations at SC main program (Tutorials, Workshops, BoFs, Posters, and Doctoral Showcase)

- Presentation at many other booths (Mellanox, Intel, Microsoft, and AWS) and satellite events

- Complete details available at
  
  http://mvapich.cse.ohio-state.edu/conference/752/talks/
Funding Acknowledgments

**Funding Support by**

![Logos of various sponsors](image1)

**Equipment Support by**

![Logos of various sponsors](image2)
Personnel Acknowledgments

Current Students (Graduate)
- A. Awan (Ph.D.)
- M. Bayatpour (Ph.D.)
- C.-H. Chu (Ph.D.)
- J. Hashmi (Ph.D.)
- A. Jain (Ph.D.)
- K. S. Kandadi (M.S.)
- Q. Zhou (Ph.D.)

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.)
- S. Chakraborthy (Ph.D.)
- 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. Kumar (M.S.)
- S. Krishnamoorthy (M.S.)
- K. Kandalla (Ph.D.)
- M. Li (Ph.D.)

Past Post-Docs
- D. Banerjee
- X. Besseron
- H.-W. Jin
- J. Lin
- M. Luo
- E. Mancini

Past Research Scientists
- K. Hamidouche
- S. Sur
- X. Lu

Past Programmers
- D. Bureddy
- J. Perkins

Past Research Specialists
- M. Arnold

Current Research Scientist
- H. Subramoni

Current Post-doc
- M. S. Ghazimeersaeed
- A. Ruhela
- K. Manian

Current Students (Undergraduate)
- V. Gangal (B.S.)
- N. Sarkauskas (B.S.)

Current Research Specialist
- J. Smith
Thank You!

panda@cse.ohio-state.edu

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

Follow us on
https://twitter.com/mvapich

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

The High-Performance Big Data Project
http://hibd.cse.ohio-state.edu/

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