MVAPICH2-GDR: Pushing the Frontier of MPI Libraries Enabling GPUDirect Technologies

GPU Technology Conference (GTC 2018)

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 Collectives for Deep Learning
  • 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 2,875 organizations in 86 countries
- More than 460,000 (> 0.46 million) downloads from the OSU site directly
- Empowering many TOP500 clusters (Nov ‘17 ranking)
  - 1st, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China
  - 9th, 556,104 cores (Oakforest-PACS) in Japan
  - 12th, 368,928-core (Stampede2) at TACC
  - 17th, 241,108-core (Pleiades) at NASA
  - 48th, 76,032-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](http://mvapich.cse.ohio-state.edu)
- Empowering Top500 systems for over a decade
MVAPICH2 Release Timeline and Downloads

- MV 0.9.4
- MV2 0.9.0
- MV2 0.9.8
- MV2 1.0
- MV2 1.0.3
- MV 1.1
- MV2 1.4
- MV2 1.5
- MV2 1.6
- MV2 1.7
- MV2 1.8
- MV2 1.9
- MV2 1.10
- MV2-GDR 2.0b
- MV2-MIC 2.0
- MV2-GDR 2.3a
- MV2-GDR 2.3rc1
- MV2 Virt 2.2
- MV2 2.3rc1
- OSU INAM 0.9.3

Number of Downloads vs Timeline
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

<table>
<thead>
<tr>
<th>High-Performance Parallel Programming Libraries</th>
<th></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</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>

<table>
<thead>
<tr>
<th>Microbenchmarks</th>
<th></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>

<table>
<thead>
<tr>
<th>Tools</th>
<th></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>
MVAPIC2H-GDR: 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 . . .

- 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 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
Using MVAPICH2-GPUDirect Version

• MVAPICH2-2.3 with GDR support can be downloaded from

  https://mvapich.cse.ohio-state.edu/download/mvapich2gdr/

• System software requirements
  • Mellanox OFED 3.2 or later
  • NVIDIA Driver 367.48 or later
  • NVIDIA CUDA Toolkit 7.5/8.0/9.0 or later
  • Plugin for GPUDirect RDMA


• Strongly recommended
  • GDRCOPY module from NVIDIA

    https://github.com/NVIDIA/gdrcopy

• Contact MVAPICH help list with any questions related to the package

  mvapich-help@cse.ohio-state.edu
MVAPICH2-GDR 2.3a

- Released on 11/09/2017
- Major Features and Enhancements
  - Based on MVAPICH2 2.2
  - Support for CUDA 9.0
  - 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
  - Enhanced performance of GPU-based point-to-point communication
  - Leverage Linux Cross Memory Attach (CMA) feature for enhanced host-based communication
  - Enhanced performance of MPI_Allreduce for GPU-resident data
  - InfiniBand Multicast (IB-MCAST) based designs for GPU-based broadcast and streaming applications
    - Basic support for IB-MCAST designs with GPUDirect RDMA
    - Advanced support for zero-copy IB-MCAST designs with GPUDirect RDMA
    - Advanced reliability support for IB-MCAST designs
  - Efficient broadcast designs for Deep Learning applications
  - Enhanced collective tuning on Xeon, OpenPOWER, and NVIDIA DGX-1 systems
Optimized MVAPICH2-GDR Design

**GPU-GPU Inter-node Latency**

- **MV2-(NO-GDR)**
- **MV2-GDR-2.3a**

**Latency (μs)**

- 1.88us
- 10x

**Message Size (Bytes)**

- 0
- 1
- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512
- 1K
- 2K
- 4K
- 8K

**GPU-GPU Inter-node Bandwidth**

- **MV2-(NO-GDR)**
- **MV2-GDR-2.3a**

**Bandwidth (MB/s)**

- 0
- 1000
- 2000
- 3000
- 4000
- 5000
- 6000

**Message Size (Bytes)**

- 1
- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512
- 1K
- 2K
- 4K

**Latency (us)**

- 10x
- 9x

**Message Size (Bytes)**

- 0
- 1
- 2
- 4
- 8
- 16
- 32
- 64
- 128
- 256
- 512
- 1K
- 2K
- 4K
- 8K

**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

Network Based Computing Laboratory

GTC 2018
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

**256K Particles**
- Average Time Steps per second (TPS)
- Number of Processes
Application-Level Evaluation (Cosmo) and Weather Forecasting in Switzerland

Wilkes GPU Cluster

CSCS GPU cluster

- 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


Cosmo model: http://www2.cosmo-model.org/content/tasks/operational/meteoSwiss/
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 Collectives for Deep Learning
  • 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

Available with 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

- **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-2.3a**

- 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

---

Waste of computing resources on CPU and GPU
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 Collectives for Deep Learning
  • Out-of-core processing for Deep Learning

• Conclusions
Initial (Basic) Support for GPU Managed Memory

- CUDA 6.0 NVIDIA introduced CUDA Managed (or Unified) memory allowing a common memory allocation for GPU or CPU through `cudaMallocManaged()` call
- Significant productivity benefits due to abstraction of explicit allocation and `cudaMemcpy()`
- Extended MVAPICH2 to perform communications directly from managed buffers (Available since MVAPICH2-GDR 2.2b)
- OSU Micro-benchmarks extended to evaluate the performance of point-to-point and collective communications using managed buffers

D. S. Banerjee, K Hamidouche, and D. K Panda, Designing High Performance Communication Runtime for GPU Managed Memory: Early Experiences, GPGPU-9 Workshop, held in conjunction with PPoPP ‘16
Enhanced Support for Intra-node Managed Memory

- CUDA Managed => 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 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
- Available since MVAPICH2-GDR 2.2RC1
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 Collectives for Deep Learning
  • 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

Sender

HPC resources for real-time analytics

Data streaming-like broadcast operations
Hardware Multicast-based Broadcast

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

- Overhead
  - IB UD limit
  - GDR limit

Optimized Broadcast Send

• Preparing Intermediate buffer ($im_{buf}$)
  – Page-locked (pinned) host buffer
    ➢ Fast Device-Host data movement
  – Allocated at initialization phase
    ➢ Low overhead
• Streaming data through host
  – Fine-tuned chunked data
  – Asynchronous copy operations
    ➢ Three-stage pipeline

Optimized Broadcast Receive

- Zero-copy broadcast receive
  - Pre-posted user buffer \((d_{in})\)
  - Avoids additional data movement
  - Leverages IB Scatter and GDR features
    - Low-latency
    - Free-up PCIe resources for applications

Broadcast on Multi-GPU systems

- Proposed Intra-node Topology-Aware Broadcast
  - CUDA InterProcess Communication (IPC)

Available in 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


Network Based Computing Laboratory GTC 2018
Application-based Evaluation: CUDA-Aware CNTK

- @ RI2 cluster, 16 GPUs, 1 GPU/node
  - CUDA-Aware Microsoft Cognitive Toolkit (CA-CNTK) \(^{[2]}\)

<table>
<thead>
<tr>
<th>Model</th>
<th>8 GPUs</th>
<th>16 GPUs</th>
<th>8 GPUs</th>
<th>16 GPUs</th>
<th>8 GPUs</th>
<th>16 GPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td><img src="image" alt="Speedup" /></td>
<td>16%</td>
<td><img src="image" alt="Speedup" /></td>
<td>18%</td>
<td><img src="image" alt="Speedup" /></td>
<td>24%</td>
</tr>
<tr>
<td>VGG</td>
<td></td>
<td><img src="image" alt="Speedup" /></td>
<td></td>
<td><img src="image" alt="Speedup" /></td>
<td></td>
<td><img src="image" alt="Speedup" /></td>
</tr>
<tr>
<td>ResNet-50</td>
<td></td>
<td><img src="image" alt="Speedup" /></td>
<td></td>
<td><img src="image" alt="Speedup" /></td>
<td></td>
<td><img src="image" alt="Speedup" /></td>
</tr>
</tbody>
</table>

- 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 Collectives for Deep Learning
  • Out-of-core processing for Deep Learning
• Conclusions
Deep Learning: New Challenges for MPI Runtimes

- Deep Learning frameworks are a different game altogether
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- Existing State-of-the-art
  - cuDNN, cuBLAS, NCCL → scale-up performance
  - NCCL2, CUDA-Aware MPI → scale-out performance
    - For small and medium message sizes only!
- Proposed: Can we co-design the MPI runtime (MVAPICH2-GDR) and the DL framework (Caffe) to achieve both?
  - Efficient Overlap of Computation and Communication
  - Efficient Large-Message Communication (Reductions)
  - What application co-designs are needed to exploit communication-runtime co-designs?

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
Initial Evaluation shows promising performance gains for MVAPICH2-GDR 2.3a*

8 GPUs (2 nodes) MVAPICH2-GDR vs. Baidu-Allreduce and OpenMPI 3.0

MVAPICH2: Allreduce Comparison with Baidu and OpenMPI

*Available with MVAPICH2-GDR 2.3a
MVAPICH2: Allreduce Comparison with Baidu and OpenMPI

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

*Available with MVAPICH2-GDR 2.3a
OSU-Caffe: Scalable Deep Learning

• 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
  – 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 Collectives for Deep Learning
  • Out-of-core processing for Deep Learning

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

Intra-node Latency

- MVAPICH2-GDR 2.3a
- ~0.5 μs

Intra-node Bandwidth

- MVAPICH2-GDR 2.3a
- ~30GB/s

Intra-node Bi-directional Bandwidth

- MVAPICH2-GDR 2.3a
- ~60GB/s

Inter-node Latency

- MVAPICH2-GDR 2.3a
- ~2.3 μs

Inter-node Bandwidth

- MVAPICH2-GDR...
- ~12GB/s

Inter-node Bi-directional Bandwidth

- MVAPICH2-GDR...
- ~24GB/s

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

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

- **Intra-node Bandwidth:** 33.9 GB/sec (NVLink)
- **Inter-node Latency:** 23.8 us (without GPUDirectRDMA)
- **Inter-node Bandwidth:** 11.9 GB/sec (EDR)

---

**Intra-node Latency:** 14.6 us (without GPUDirectRDMA)

**Inter-node Latency:** 23.8 us (without GPUDirectRDMA)

Available in 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 Collectives for Deep Learning
  • 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 2.3a provides container (Docker) support
- More details are available in the MVAPICH2-GDR User Guide
  - [http://mvapich.cse.ohio-state.edu/userguide/gdr/2.3a/](http://mvapich.cse.ohio-state.edu/userguide/gdr/2.3a/)
- 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

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
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 Collectives for Deep Learning
  • Out-of-core processing for Deep Learning
• Conclusions
Scalable Host-based Collectives with CMA (Intra-node Reduce & AlltoAll)

(Nodes=1, PPN=20)

Reduce

(Nodes=1, PPN=20)

Alltoall

Up to 5X and 3x performance improvement by MVAPICH2 for small and large messages respectively
Scalable Host-based Collectives with CMA (Intra-node, Gather & Scatter)

Up to 24X and 15x performance improvement by MVAPICH2 for medium to large messages respectively
Scalable Host-based Collectives with CMA (Multi-node, Reduce & Alltoall)

Up to 12.4X and 8.5X performance improvement by MVAPICH2 for small and large messages respectively.
Scalable Host-based Collectives with CMA (Multi-node, Gather & Scatter)

Up to 17.8X and 3.9X improvement with MVAPICH2-GDR-Next for medium to large messages respectively

(Nodes=4, PPN=20)
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 Collectives for Deep Learning
  • Out-of-core processing for Deep Learning

• Conclusions
• **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*
Optimized All-Reduce with XPMEM

- Optimized MPI All-Reduce Design in MVAPICH2
  - \textit{Up to 2X} performance improvement over OpenMPI for inter-node.

Optimized Runtime Parameters: \texttt{MV2\_CPU\_BINDING\_POLICY=hybrid MV2\_HYBRID\_BINDING\_POLICY=bunch}
- Optimized MPI Reduce Design in MVAPICH2
  - *Up to 3.1X* performance improvement over OpenMPI at scale and up to *37%* over spectrum MPI on intra-node

**Network Based Computing Laboratory**

Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch
Optimized Reduce with XPMEM

- **Optimized MPI Reduce Design in MVAPICH2**
  - *Up to 7X* performance improvement over OpenMPI at scale

*Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch*
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 Collectives for Deep Learning
  • Out-of-core processing for Deep Learning

• Conclusions
MVAPICH2-GDR vs. NCCL2 – Broadcast Operation

- Optimized designs in MVAPICH2-GDR 2.3b* offer better/comparable performance for most cases
- MPI_Bcast (MVAPICH2-GDR) vs. ncclBcast (NCCL2) on 16 K-80 GPUs

*Will be available with upcoming MVAPICH2-GDR 2.3b

Platform: Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 2 K-80 GPUs, and EDR InfiniBand Inter-connect
MVAPICH2-GDR vs. NCCL2 – Reduce Operation

- Optimized designs in MVAPICH2-GDR 2.3b* offer better/comparable performance for most cases
- MPI_Reduce (MVAPICH2-GDR) vs. ncclReduce (NCCL2) on 16 GPUs

*Will be available with upcoming MVAPICH2-GDR 2.3b

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 2.3b* offer better/comparable performance for most cases

MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) on 16 GPUs

*Will be available with upcoming MVAPICH2-GDR 2.3b

Platform: Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 1 K-80 GPUs, and EDR InfiniBand Inter-connect
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 Collectives for Deep Learning
- Out-of-core processing for Deep Learning

Conclusions
Out-of-Core Deep Neural Network Training with Caffe

• Large DNNs cannot be trained on GPUs due to memory limitation!
  – ResNet-50 is the state-of-the-art DNN architecture 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.
  – In addition to Out-of-core Training support, productivity can be greatly enhanced in terms of DL framework design by using the new Unified Memory features.
Performance Trends for OC-Caffe

- Comparable performance to Caffe-Default for “in-memory” batch sizes
- OC-Caffe-Opt: up to 5X improvement over Intel-MKL-optimized CPU-based AlexNet training on Volta V100 GPU with CUDA9 and CUDNN7

OC-Caffe will be released by the HiDL Team@OSU
hidl.cse.ohio-state.edu

Submission Under Review
Performance Trends for OC-Caffe

- 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
- OC-Caffe allows efficient scale-up on DGX-1 system with Pascal P100 GPUs with CUDA9 and CUDNN7
Can Unified-Memory also Simplify Framework Design?

- Enhanced and simplified the Caffe framework
- Simplified Layer class and all inherited classes (e.g. ConvolutionLayer)
- Remove almost all of the memory allocation, movement, and state management code in SyncedMemory and Blob class
- Estimated 3,000 lines of repetitive and error-prone code can be eliminated
- Developers can add new inherited Layer classes in a much simpler manner
  - E.g. Implement a single Forward function instead of forward_gpu() and forward_cpu() function separately.

Submission Under Review
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 Collectives for Deep Learning
  • 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
• 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 High-Performance Deep Learning (HiDL) frameworks and applications
Join Us for a Connect with the Experts Session

• CE8118 - Connect with the Experts: MVAPICH for GPU

• Session Schedule
  • Wednesday, Mar 28, 3:00 PM - 4:00 PM, LL Pod A

• Session Speakers
  • Dhabaleswar Panda - Professor and University Distinguished Scholar, OSU
  • Davide Rossetti - Senior Software Engineer, NVIDIA
  • Hari Subramoni - Research Scientist, OSU

• Session Description
  • MVAPICH is a well established MPI library supporting GPUDirect. Meet with the experts to discuss how you can optimize your HPC and AI application using GPUDirect RDMA, with MVAPICH GDR, and learn about MVAPICH GDS, newest feature, supporting GPUDirect Async.
Personnel Acknowledgments

<table>
<thead>
<tr>
<th>Current Students (Graduate)</th>
<th>Current Students (Undergraduate)</th>
<th>Current Research Scientists</th>
<th>Current Research Specialist</th>
</tr>
</thead>
<tbody>
<tr>
<td>– A. Awan (Ph.D.)</td>
<td>– J. Hashmi (Ph.D.)</td>
<td>– X. Lu</td>
<td>– J. Smith</td>
</tr>
<tr>
<td>– M. Bayatpour (Ph.D.)</td>
<td>– P. Kousha (Ph.D.)</td>
<td>– A. Mamidala (Ph.D.)</td>
<td></td>
</tr>
<tr>
<td>– S. Chakraborty (Ph.D.)</td>
<td>– D. Shankar (Ph.D.)</td>
<td>– G. Marsh (M.S.)</td>
<td></td>
</tr>
<tr>
<td>– S. Guganani (Ph.D.)</td>
<td>– J. Zhang (Ph.D.)</td>
<td>– A. Moody (M.S.)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Past Students</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>– A. Augustine (M.S.)</td>
<td>– W. Huang (Ph.D.)</td>
<td>– J. Liu (Ph.D.)</td>
<td>– R. Rajachandrasekar (Ph.D.)</td>
</tr>
<tr>
<td>– S. Bhagvat (M.S.)</td>
<td>– J. Jose (Ph.D.)</td>
<td>– A. Mamidala (Ph.D.)</td>
<td>– A. Singh (Ph.D.)</td>
</tr>
<tr>
<td>– A. Bhat (M.S.)</td>
<td>– S. Kini (M.S.)</td>
<td>– G. Marsh (M.S.)</td>
<td>– J. Sridhar (M.S.)</td>
</tr>
<tr>
<td>– D. Buntinas (Ph.D.)</td>
<td>– M. Koop (Ph.D.)</td>
<td>– V. Meshram (M.S.)</td>
<td>– S. Sur (Ph.D.)</td>
</tr>
<tr>
<td>– B. Chandrasekharan (M.S.)</td>
<td>– R. Kumar (M.S.)</td>
<td>– S. Naravula (Ph.D.)</td>
<td>– K. Vaidyanathan (Ph.D.)</td>
</tr>
<tr>
<td>– N. Dandapanthula (M.S.)</td>
<td>– S. Krishnamoorthy (M.S.)</td>
<td>– R. Noronha (Ph.D.)</td>
<td>– A. Vishnu (Ph.D.)</td>
</tr>
<tr>
<td>– V. Dhanraj (M.S.)</td>
<td>– K. Kandalla (Ph.D.)</td>
<td>– X. Ouyang (Ph.D.)</td>
<td>– J. Wu (Ph.D.)</td>
</tr>
<tr>
<td>– T. Gangadharappa (M.S.)</td>
<td>– M. Li (Ph.D.)</td>
<td>– S. Pai (M.S.)</td>
<td>– W. Yu (Ph.D.)</td>
</tr>
<tr>
<td>– K. Gopalakrishnan (M.S.)</td>
<td>– P. Lai (M.S.)</td>
<td>– S. Potluri (Ph.D.)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Past Post-Docs</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>– X. Besseron</td>
<td>– M. Luo</td>
<td>– J. Vienne</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Current Post-doc</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>– A. Ruhela</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Past Research Scientist</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>– K. Hamidouche</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>– S. Sur</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Past Programmers</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>– D. Bureddy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>– J. Perkins</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Network Based Computing Laboratory
GTC 2018
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/