Overview of the MVAPICH Project: Latest Status and Future Roadmap

MVAPICH2 User Group (MUG) Conference

by

Dhabaleswar K. (DK) Panda
The Ohio State University
E-mail: panda@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~panda
High-End Computing (HEC): PetaFlop to ExaFlop

Expected to have an ExaFlop system in 2021-2022!

- 100 PetaFlops in 2017
- 442 PetaFlops in 2020 (Fugaku in Japan with 7.63M cores)
- 1 ExaFlops
Supporting Programming Models for Multi-Petaflop and Exaflop Systems: Challenges

Application Kernels/Applications (HPC and DL)

Middleware

Programming Models
MPI, PGAS (UPC, Global Arrays, OpenSHMEM), CUDA, OpenMP, OpenACC, Hadoop, Spark (RDD, DAG), TensorFlow, PyTorch, etc.

Communication Library or Runtime for Programming Models
Point-to-point Communication
Collective Communication
Energy-Awareness
Synchronization and Locks
I/O and File Systems
Fault Tolerance

Networking Technologies
(InfiniBand, Ethernet, RoCE, Omni-Path, and Slingshot)

Multi-/Many-core Architectures

Accelerators (GPU and FPGA)

Co-Design Opportunities and Challenges across Various Layers
Performance
Scalability
Resilience
Designing (MPI+X) at Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication (both two-sided and one-sided)
  - Scalable job start-up
  - Low memory footprint

- Scalable Collective communication
  - Offload
  - Non-blocking
  - Topology-aware

- Balancing intra-node and inter-node communication for next generation nodes (128-1024 cores)
  - Multiple end-points per node

- Support for efficient multi-threading

- Integrated Support for Accelerators (GPGPUs and FPGAs)

- Fault-tolerance/resiliency

- QoS support for communication and I/O

- Support for Hybrid MPI+PGAS programming (MPI + OpenMP, MPI + UPC, MPI + OpenSHMEM, MPI+UPC++, CAF, ...)

- Virtualization

- Energy-Awareness
Overview of the MVAPICH2 Project

- High Performance open-source MPI Library
- Support for multiple interconnects
  - InfiniBand, Omni-Path, Ethernet/iWARP, RDMA over Converged Ethernet (RoCE), and AWS EFA
- Support for multiple platforms
  - x86, OpenPOWER, ARM, Xeon-Phi, GPGPUs (NVIDIA and AMD)
- Started in 2001, first open-source version demonstrated at SC ’02
- Supports the latest MPI-3.1 standard
- http://mvapich.cse.ohio-state.edu
- Additional optimized versions for different systems/environments:
  - MVAPICH2-X (Advanced MPI + PGAS), since 2011
  - MVAPICH2-GDR with support for NVIDIA GPGPUs, since 2014
  - MVAPICH2-MIC with support for Intel Xeon-Phi, since 2014
  - MVAPICH2-Virt with virtualization support, since 2015
  - MVAPICH2-EA with support for Energy-Awareness, since 2015
  - MVAPICH2-Azure for Azure HPC IB instances, since 2019
  - MVAPICH2-X-AWS for AWS HPC+EFA instances, since 2019
- Tools:
  - OSU MPI Micro-Benchmarks (OMB), since 2003
  - OSU InfiniBand Network Analysis and Monitoring (INAM), since 2015
- Used by more than 3,200 organizations in 89 countries
- More than 1.43 Million downloads from the OSU site directly
- Empowering many TOP500 clusters (June ‘21 ranking)
  - 4th, 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China
  - 10th, 448, 448 cores (Frontera) at TACC
  - 20th, 288,288 cores (Lassen) at LLNL
  - 31st, 570,020 cores (Nurion) in South Korea and many others
- Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, OpenHPC, and Spack)
- Partner in the 10th ranked TACC Frontera system
- Empowering Top500 systems for more than 15 years
Architecture of MVAPICH2 Software Family for HPC and DL/ML

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/AMD GPGPU)

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

* Upcoming
Production Quality Software Design, Development and Release

• Rigorous Q&A procedure before making a release
  – Exhaustive unit testing
  – Various test procedures on diverse range of platforms and interconnects
  – Test 19 different benchmarks and applications including, but not limited to
    • OMB, IMB, MPICH Test Suite, Intel Test Suite, NAS, Scalapak, and SPEC
      – Spend about 18,000 core hours per commit
  – Performance regression and tuning
  – Applications-based evaluation
  – Evaluation on large-scale systems

• All versions (alpha, beta, RC1 and RC2) go through the above testing
## MVAPICH2 Software Family

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2)</strong></td>
<td>MVAPICH2</td>
</tr>
<tr>
<td><strong>Optimized Support for Microsoft Azure Platform with InfiniBand</strong></td>
<td>MVAPICH2-Azure</td>
</tr>
<tr>
<td><strong>Advanced MPI features/support (UMR, ODP, DC, Core-Direct, SHArP, XPMEM), OSU INAM (InfiniBand Network Monitoring and Analysis)</strong></td>
<td>MVAPICH2-X</td>
</tr>
<tr>
<td><strong>Advanced MPI features (SRD and XPMEM) with support for Amazon Elastic Fabric Adapter (EFA)</strong></td>
<td>MVAPICH2-X-AWS</td>
</tr>
<tr>
<td><strong>Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning Applications</strong></td>
<td>MVAPICH2-GDR</td>
</tr>
<tr>
<td><strong>Energy-aware MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2)</strong></td>
<td>MVAPICH2-EA</td>
</tr>
<tr>
<td><strong>MPI Energy Monitoring Tool</strong></td>
<td>OEMT</td>
</tr>
<tr>
<td><strong>InfiniBand Network Analysis and Monitoring</strong></td>
<td>OSU INAM</td>
</tr>
<tr>
<td><strong>Microbenchmarks for Measuring MPI and PGAS Performance</strong></td>
<td>OMB</td>
</tr>
</tbody>
</table>
MVAPICH2 2.3.6

• Released on 05/11/2021

• Major Features and Enhancements
  – Support collective offload using Mellanox’s SHARP for Reduce and Bcast
    • Enhanced tuning framework for Reduce and Bcast using SHARP
  – Enhanced performance for UD-Hybrid code
  – Add multi-rail support for UD-Hybrid code
  – Enhanced performance for shared-memory collectives
  – Enhanced job-startup performance for flux job launcher
  – Add support in mpirun_rsh to use srun daemons to launch jobs
  – Add support in mpirun_rsh to specify processes per node using ‘-ppn’ option
  – Use PMI2 by default when SLURM is selected as process manager
  – Add support to use aligned memory allocations for multi-threaded applications
  – Architecture detection and enhanced point-to-point tuning for Oracle BM.HPC2 cloud shape
  – Enhanced collective tuning for Frontera@TACC and Expanse@SDSC
  – Add support for GCC compiler v11
  – Add support for Intel IFX compiler
  – Update hwloc v1 code to v1.11.14 & hwloc v2 code to v2.4.2
Highlights of MVAPICH2 2.3.6-GA Release

- Support for highly-efficient inter-node and intra-node communication
- Scalable Start-up
- Collective offload using Mellanox’s SHARP support
- Performance Engineering with MPI_T
AMD Milan + HDR 200

Intra-Node CPU Point-to-Point

Latency

Bandwidth

Inter-Node CPU Point-to-Point

Latency

Bandwidth

AMD EPYC 7V13 64-Core Processor, Mellanox ConnectX-6 HDR HCA
• MPI_Init takes 31 seconds on 229,376 processes on 4,096 nodes
• All numbers reported with 56 processes per node

New designs available since MVAPICH2-2.3.4
Performance of Collectives with SHARP on TACC Frontera

Optimized SHARP designs in MVAPICH2-X

*Up to 9X* performance improvement with SHARP over MVAPICH2-X default for 1ppn MPI_Barrier, *6X* for 1ppn MPI_Reduce and *5X* for 1ppn MPI_Allreduce


Optimized Runtime Parameters: `MV2_ENABLE_SHARP = 1`
Performance Engineering Applications using MVAPICH2 and TAU

- Enhance existing support for MPI_T in MVAPICH2 to expose a richer set of performance and control variables
- Get and display MPI Performance Variables (PVARs) made available by the runtime in TAU
- Control the runtime’s behavior via MPI Control Variables (CVARs)
- Introduced support for new MPI_T based CVARs to MVAPICH2
  - `MPIR_CVAR_MAX_INLINE_MSG_SZ`, `MPIR_CVAR_VBUF_POOL_SIZE`, `MPIR_CVAR_VBUF_SECONDARY_POOL_SIZE`
- TAU enhanced with support for setting MPI_T CVARs in a non-interactive mode for uninstrumented applications
- S. Ramesh, A. Maheo, S. Shende, A. Malony, H. Subramoni, and D. K. Panda, *MPI Performance Engineering with the MPI Tool Interface: the Integration of MVAPICH and TAU*, *EuroMPI/USA ’17, Best Paper Finalist*
- More details in Sameer Shende’s talk (later today)

### VBUF usage without CVAR based tuning as displayed by ParaProf

<table>
<thead>
<tr>
<th>Metric</th>
<th>Uninstrumented</th>
<th>Instrumented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>3,313,016</td>
<td>3,313,016</td>
</tr>
<tr>
<td>Total</td>
<td>3,313,016</td>
<td>3,313,016</td>
</tr>
<tr>
<td>Total</td>
<td>3,313,016</td>
<td>3,313,016</td>
</tr>
</tbody>
</table>

### VBUF usage with CVAR based tuning as displayed by ParaProf

<table>
<thead>
<tr>
<th>Metric</th>
<th>Uninstrumented</th>
<th>Instrumented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>1,813,016</td>
<td>1,813,016</td>
</tr>
<tr>
<td>Total</td>
<td>1,813,016</td>
<td>1,813,016</td>
</tr>
<tr>
<td>Total</td>
<td>1,813,016</td>
<td>1,813,016</td>
</tr>
</tbody>
</table>
MVAPICH2 Upcoming Features

• Support for Rockport Networks adapter
• Java Binding for MVAPICH2
• Integration with Sandia SST for large-scale network simulation
Inter-node point-to-point Latency and Bandwidth (Rockport Networks)

**Small message Latency**

- MVAPICH2 delivers around 3.0 microsec latency for small messages
- Using multiple QPs gives up to 3.9x reduction in latency

**Medium/Large message Latency**

- MVAPICH2-4-QP delivers
  - 10229 MB/sec peak unidirectional bandwidth
  - 20165 MB/Sec peak bidirectional bandwidth

**Uni-directional Bandwidth**

**Bi-Directional Bandwidth**

More details on today’s talk from Rockport Networks
Java Binding in MVAPICH2: Preliminary Results Bcast Performance (8 processes)

1.51x better on average (buffer API)

More details in tomorrow’s talk by Aamir Shafi
MVAPICH2 + SST Integration - Understanding the impact of topology

- Large scale simulation of MPI_Alltoall using OMB
- Impact of topology on Alltoall collective implemented in MVAPICH2

- Large scale simulation of MPI_Alltoall using OMB
- Impact of topology on Alltoall collective implemented in MVAPICH2

More details in tomorrow’s talk
By Kaushik Kandadi Suresh
## MVAPICH2 Software Family

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2)</td>
<td>MVAPICH2</td>
</tr>
<tr>
<td>Optimized Support for Microsoft Azure Platform with InfiniBand</td>
<td>MVAPICH2-Azure</td>
</tr>
<tr>
<td>Advanced MPI features/support (UMR, ODP, DC, Core-Direct, SHArP, XPMEM), OSU INAM (InfiniBand Network Monitoring and Analysis)</td>
<td>MVAPICH2-X</td>
</tr>
<tr>
<td>Advanced MPI features (SRD and XPMEM) with support for Amazon Elastic Fabric Adapter (EFA)</td>
<td>MVAPICH2-X-AWS</td>
</tr>
<tr>
<td>Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning Applications</td>
<td>MVAPICH2-GDR</td>
</tr>
<tr>
<td>Energy-aware MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2)</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>
Overview of Some of the MVAPICH2-X Features

- Direct Connect (DC) Transport
- Cooperative Rendezvous Protocol
- Asynchronous Progress
- XPMEM-based Collectives
Impact of DC Transport Protocol on Neuron

- Up to 76% benefits over MVAPICH2 for Neuron using Direct Connected transport protocol at scale
  - VERSION 7.6.2 master (f5a1284) 2018-08-15
- Numbers taken on bbpv2.epfl.ch
  - Knights Landing nodes with 64 ppn
  - ./x86_64/special -mpi -c stop_time=2000 -c is_split=1 parinit.hoc
  - Used “runtime” reported by execution to measure performance
- Environment variables used
  - MV2_USE_DC=1
  - MV2_NUM_DC_TGT=64
  - MV2_SMALL_MSG_DC_POOL=96
  - MV2_LARGE_MSG_DC_POOL=96
  - MV2_USE_RDMA_CM=0

Available from MVAPICH2-X 2.3rc2 onwards

More details in talk
“Building Brain Circuits: Experiences with shuffling terabytes of data over MPI”, by Matthias Wolf at MUG’20
https://www.youtube.com/watch?v=TFi8O3-Hznw
Cooperative Rendezvous Protocols

- Use both sender and receiver CPUs to progress communication concurrently
- Dynamically select rendezvous protocol based on communication primitives and sender/receiver availability (load balancing)
- Up to 2x improvement in large message latency and bandwidth
- Up to 19% improvement for Graph500 at 1536 processes

Cooperative Rendezvous Protocols for Improved Performance and Overlap
S. Chakraborty, M. Bayatpour, J Hashmi, H. Subramoni, and DK Panda,
SC ‘18 (Best Student Paper Award Finalist)

Platform: 2x14 core Broadwell 2680 (2.4 GHz)
Mellanox EDR ConnectX-5 (100 GBps)
Baseline: MVAPICH2X-2.3rc1, Open MPI v3.1.0
Available in MVAPICH2-X 2.3rc2
Benefits of the New Asynchronous Progress Design: Broadwell + InfiniBand

Up to 33% performance improvement in P3DFFT application with 448 processes
Up to 29% performance improvement in HPL application with 896 processes


Available since MVAPICH2-X 2.3rc1
Shared Address Space (XPMEM)-based Collectives Design

**OSU_Allreduce (Broadwell 256 procs)**

- MVAPICH2-2.3b
- IMPI-2017v1.132
- MVAPICH2-X-2.3rc1

**OSU_Reduce (Broadwell 256 procs)**

- MVAPICH2-2.3b
- IMPI-2017v1.132
- MVAPICH2-2.3rc1

- "**Shared Address Space**"-based true **zero-copy** Reduction collective designs in MVAPICH2
- Offloaded computation/communication to peers ranks in reduction collective operation
- Up to **4X** improvement for 4MB Reduce and up to **1.8X** improvement for 4M AllReduce

J. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, and D. Panda, **Designing Efficient Shared Address Space Reduction Collectives for Multi-/Many-cores**, International Parallel & Distributed Processing Symposium (IPDPS '18), May 2018.
MVAPICH2-X Upcoming Features

- Optimized Derived Datatype Designs
- Exploiting Hardware Tag Matching
- Neighborhood Collectives
- Support for SHARP Streaming Aggregation (SAT)
- Support for AWS Cloud Arm HPC Instances
- Support for Oracle Cloud HPC Shapes
Performance of DDTbench with Optimized Derived Datatype Support

- NASMG: Block length is 8 bytes for X-direction and 256 bytes to 5KB in the Y-direction
- 28% improvement over MVAPICH2 and 2.5X over IntelMPI
- Inputs: \(A = (256,32,32)\), \(B = (512,66,66)\), \(C = (2048,66,120)\), \(D = (5120,92,120)\)

![NASMG Performance Graph](#)

- WRF: The datatypes used in WRF are struct of vectors for both X and Y direction
- We see improvements up to 1.75X compared to MVAPICH2 and up to 2.5X improvements over IntelMPI
- Inputs: \(A = (4,4018,8,4010)\), \(B = (4,2060,8,2056)\), \(C = (4,6012,8,6008)\)

![WRF Performance Graph](#)

More details in today’s talk by Kaushik Kandadi Suresh

*Latencies are normalized with respect to IntelMPI numbers
Performance of MPI_Ialltoall using HW Tag Matching

- Up to 1.8x Performance Improvement, Sustained benefits as system size increases
Optimized Designs for Neighborhood Collectives

- **SpMM**

- **NAS DT**

![Graph showing SpMM performance gain with up to 28,338 processes](image1)

- up to 34x speedup

![Graph showing NAS-DT performance gain with up to 1024 processes](image2)

- up to 15% improvement

Optimized MPI_Allreduce Performance with MVAPICH2-X + SHARP SAT

- SHARP provides flat scaling, even for large messages
- Up to 3.95X benefits over MVAPICH2-X-2.3 using SAT + optimized designs

Platform: Intel(R) Xeon(R) Gold 6138 nodes equipped with a dual-socket CPU and InfiniBand HDR-200 Interconnect
Optimized MPI_Reduce Performance with MVAPICH2-X + SHARP SAT

- Comparing max latency for MPI_Reduce as the root is the bottleneck
- Up to 15.6X benefits over MVAPICH2-X-2.3 using SAT + optimized designs

Platform: Intel(R) Xeon(R) Gold 6138 nodes equipped with a dual-socket CPU and InfiniBand HDR-200 Interconnect

More details in Today’s talk by Bharath Ramesh
MVAPICH2-X on AWS EFA Arm HPC Instances

- Collective Performance on 32 AWS c6gn.18xlarge instances

More details in Today’s Talk by Shulei Xu
MVAPICH2-X Performance on OCI HPC System

- Collective performance evaluation on 8 BM.HPC2 instances

![Graphs showing Bcast and Reduce performance comparisons between MVAPICH2-X, HPCx, and IntelMPI. The graphs illustrate latency (us) against message size (Bytes) for both Bcast and Reduce operations. The performance is compared for different message sizes ranging from 4 to 1M Bytes. MVAPICH2-X consistently shows better performance compared to HPCx and IntelMPI, with a 2.7x improvement in Bcast and a 21x improvement in Reduce. More details are available in today's talk by Shulei Xu.]
# MVAPICH2 Software Family

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2)</td>
<td>MVAPICH2</td>
</tr>
<tr>
<td>Optimized Support for Microsoft Azure Platform with InfiniBand</td>
<td>MVAPICH2-Azure</td>
</tr>
<tr>
<td>Advanced MPI features/support (UMR, ODP, DC, Core-Direct, SHArP, XPMEM), OSU INAM (InfiniBand Network Monitoring and Analysis)</td>
<td>MVAPICH2-X</td>
</tr>
<tr>
<td>Advanced MPI features (SRD and XPMEM) with support for Amazon Elastic Fabric Adapter (EFA)</td>
<td>MVAPICH2-X-AWS</td>
</tr>
<tr>
<td>Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning Applications</td>
<td>MVAPICH2-GDR</td>
</tr>
<tr>
<td>Energy-aware MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2)</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>
MVAPICH2-GDR 2.3.6

- Released on 08/12/2021
- Major Features and Enhancements
  - Based on MVAPICH2 2.3.6
  - Added support for 'on-the-fly' compression of point-to-point messages used for GPU-to-GPU communication
    - Applicable to NVIDIA GPUs
  - Added NCCL communication substrate for various MPI collectives
    - Support for hybrid communication protocols using NCCL-based, CUDA-based, and IB verbs-based primitives
    - MPI_Allreduce, MPI_Reduce, MPI_Allgather, MPI_Allgatherv, MPI_Alltoall, MPI_Alltoallv, MPI_Scatter, MPI_Scatterv, MPI_Gather, MPI_Gatherv, and MPI_Bcast
  - Full support for NVIDIA DGX, NVIDIA DGX-2 V-100, and NVIDIA DGX-2 A-100 systems
    - Enhanced architecture detection, process placement and HCA selection
    - Enhanced intra-node and inter-node point-to-point tuning
    - Enhanced collective tuning
  - Introduced architecture detection, point-to-point tuning and collective tuning for ThetaGPU @ANL
  - Enhanced point-to-point and collective tuning for NVIDIA GPUs on Frontera @TACC, Lassen @LLNL, and Sierra @LLNL
  - Enhanced point-to-point and collective tuning for Mi50 and Mi60 AMD GPUs on Corona @LLNL
  - Added several new MPI_T PVARs
  - Added support for CUDA 11.3
  - Added support for ROCm 4.1
  - Enhanced output for runtime variable MV2_SHOW_ENV_INFO
    - Tested with Horovod and common DL Frameworks
      - TensorFlow, PyTorch, and MXNet
    - Tested with MPI4Dask 0.2
      - MPI4Dask is a custom Dask Distributed package with MPI support
    - Tested with MPI4cuML 0.1
      - MPI4cuML is a custom cuML package with MPI support
- MVAPICH2-GDR 2.3.6
  - Added several new MPI_T PVARs
  - Added support for CUDA 11.3
  - Added support for ROCm 4.1
  - Enhanced output for runtime variable MV2_SHOW_ENV_INFO
  - Tested with Horovod and common DL Frameworks
    - TensorFlow, PyTorch, and MXNet
  - Tested with MPI4Dask 0.2
    - MPI4Dask is a custom Dask Distributed package with MPI support
  - Tested with MPI4cuML 0.1
    - MPI4cuML is a custom cuML package with MPI support
Highlights of some MVAPICH2-GDR Features for HPC and DL

• CUDA-Aware MPI
• Support for AMD GPU
• On-the-fly Compression for GPU-GPU Communication
• Optimized Collective Support for DGX-A100
• High-Performance
  – Deep Learning
  – Machine Learning
  – Data Science with Dask
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
Network Based Computing Laboratory

**MVAPICH2-GDR with CUDA-aware MPI Support**

---

**GPU-GPU Inter-node Latency**

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

Latency (us) vs. Message Size (Bytes)

- 1.85us
- 10x

---

**GPU-GPU Inter-node Bandwidth**

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

Bandwidth (MB/s) vs. Message Size (Bytes)

- 9x

---

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

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

Bandwidth (MB/s) vs. Message Size (Bytes)

- 11x

---

**System Configuration**

- **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**
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)

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

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)
**AMD ND A100 v4-series (NDv4) + (8 x HDR 200)**

**Intra-Node GPU Point-to-Point**
- **Latency**
- **Bandwidth**

**Inter-Node GPU Point-to-Point**
- **Latency**
- **Bandwidth**

**ND A100 v4-series**: [https://docs.microsoft.com/en-us/azure/virtual-machines/nda100-v4-series](https://docs.microsoft.com/en-us/azure/virtual-machines/nda100-v4-series)

- **AMD EPYC 7V12 64-Core Processor**
- **CUDA 11.3, NVIDIA A100 GPUs**
- **Mellanox ConnectX-6 HDR HCA**
ROCM-aware MVAPICH2-GDR - Support for AMD GPUs

Intra-Node Point-to-Point Latency

Inter-Node Point-to-Point Latency

Allreduce – 64 GPUs (8 nodes, 8 GPUs Per Node)

Bcast – 64 GPUs (8 nodes, 8 GPUs Per Node)

More details in the talk “Performance of ROCm-aware MVAPICH2-GDR on LLNL Corona Cluster with AMD GPUs" by Kawthar Shafie Khorassani on Tuesday (08/24/2021) @ 4:30 PM EDT

Performance with “On-the-fly” Compression Support in MVAPICH2-GDR

- Weak-Scaling of HPC application AWP-ODC on Lassen cluster (V100 nodes) [1]
- MPC-OPT achieves up to +18% GPU computing flops, -15% runtime per timestep
- ZFP-OPT achieves up to +35% GPU computing flops, -26% runtime per timestep

Collectives Performance on DGX2-A100 – Small Message

Allgather (4 nodes, 8 ppn, 32 GPUs)

Bcast (4 nodes, 8 ppn, 32 GPUs)

Reduce (4 nodes, 8 ppn, 32 GPUs)

Allreduce (4 nodes, 8 ppn, 32 GPUs)

More details in Today’s Talk by Chen Chun

MVAPICH2-GDR 2.3.6 vs NCCL

Size (bytes)

Latency (us)

MVAPICH2-GDR 2.3.6

NCCL

MVAPICH2-GDR 2.3.6

NCCL

MVAPICH2-GDR 2.3.6

NCCL

MVAPICH2-GDR 2.3.6

NCCL

Network Based Computing Laboratory

MVAPICH User Group Conference (MUG) 2021
MVAPICH2 (MPI)-driven Infrastructure for ML/DL Training

ML/DL Applications

TensorFlow
PyTorch
MXNet

Horovod

MVAPICH2 or MVAPICH2-X for CPU Training
MVAPICH2-GDR for GPU Training

ML/DL Applications

PyTorch

Torch.distributed
DeepSpeed

MVAPICH2 or MVAPICH2-X for CPU Training
MVAPICH2-GDR for GPU Training

More details available from: http://hidl.cse.ohio-state.edu
PyTorch, Horovod and DeepSpeed at Scale: Training ResNet-50 on 256 V100 GPUs

- Training performance for 256 V100 GPUs on LLNL Lassen
  - ~10,000 Images/sec faster than NCCL training!

<table>
<thead>
<tr>
<th>Distributed Framework</th>
<th>Torch.distributed</th>
<th>Horovod</th>
<th>DeepSpeed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Images/sec on 256 GPUs</strong></td>
<td>61,794</td>
<td>72,120</td>
<td>74,063</td>
</tr>
<tr>
<td><strong>Communication Backend</strong></td>
<td>NCCL</td>
<td>MVAPICH2-GDR</td>
<td>NCCL</td>
</tr>
</tbody>
</table>
Exploiting Model Parallelism in AI-Driven Digital Pathology

- Pathology whole slide image (WSI)
  - Each WSI = 100,000 x 100,000 pixels
  - Can not fit in a single GPU memory
  - Tiles are extracted to make training possible

- Two main problems with tiles
  - Restricted tile size because of GPU memory limitation
  - Smaller tiles lose structural information

- Can we use Model Parallelism to train on larger tiles to get better accuracy and diagnosis?

- Reduced training time significantly on OpenPOWER + NVIDIA V100 GPUs
  - 32 hours (1 node, 1 GPU) -> 7.25 hours (1 node, 4 GPUs) -> 27 mins (32 nodes, 128 GPUs)

More details in Tomorrow’s talk by Arpan Jain


Network Based Computing Laboratory
Benchmark #1: Sum of cuPy Array and its Transpose (RI2)

3.47x better on average

6.92x better on average


MPI4Dask 0.2 release (http://hibd.cse.ohio-state.edu)
MVAPICH2-GDR Upcoming Features for HPC and DL

• On-the-fly Compression for All-to-all Collective
• Scalable Distributed Training with Model-/Hybrid Parallelism for out-of-core DNN Models
• Scaling Single-Image Super-Resolution Training
All-to-All with On-the-Fly Compression

- All-to-All operation with collective level on-the-fly compression design
- ZFP-OPT(rate:4) achieves up to ~87% reduced latency at 16MB on Frontera RTX-5000 and Longhorn V100 nodes (4nodes, 4ppn)
Accelerating Transformers using SUPER

- We propose sub-graph parallelism integrated with data parallelism to accelerate the training of Transformers.

- **Approach**
  - Data and Sub-Graph Parallelism (D&SP)
    - #:way D&SP (#: number of sub-graphs)

- **Setup**
  - T5-Large-Mod on WMT Dataset
  - 1024 NVIDIA V100 GPUs

- **Speedup**
  - Up to **3.05X** over Data Parallelism (DP)

---

**More details in Tomorrow’s talk by Arpan Jain**

Single-Image Super-Resolution Training: Performance Improvement

• Throughput is improved at all scales

• MPI-Opt performs better than both NCCL and default MPI

<table>
<thead>
<tr>
<th>#GPUs</th>
<th>Throughput (Img/sec)</th>
<th>Percentage Improvement (MPI-Opt over default MPI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NCCL</td>
<td>MPI</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>8</td>
<td>76</td>
<td>75</td>
</tr>
<tr>
<td>16</td>
<td>142</td>
<td>126</td>
</tr>
<tr>
<td>32</td>
<td>269</td>
<td>242</td>
</tr>
<tr>
<td>64</td>
<td>511</td>
<td>490</td>
</tr>
<tr>
<td>128</td>
<td>989</td>
<td>933</td>
</tr>
<tr>
<td>256</td>
<td>1807</td>
<td>1675</td>
</tr>
<tr>
<td>512</td>
<td>3608</td>
<td>3258</td>
</tr>
</tbody>
</table>


More details in Tomorrow’s talk by Quentin Anthony
## MVAPICH2 Software Family

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and, RoCE (v1/v2)</td>
<td>MVAPICH2</td>
</tr>
<tr>
<td>Optimized Support for Microsoft Azure Platform with InfiniBand</td>
<td>MVAPICH2-Azure</td>
</tr>
<tr>
<td>Advanced MPI features/support (UMR, ODP, DC, Core-Direct, SHArP, XPMEM),</td>
<td>MVAPICH2-X</td>
</tr>
<tr>
<td>OSU INAM (InfiniBand Network Monitoring and Analysis)</td>
<td></td>
</tr>
<tr>
<td>Advanced MPI features (SRD and XPMEM) with support for Amazon Elastic Fabric</td>
<td>MVAPICH2-X-AWS</td>
</tr>
<tr>
<td>Adapter (EFA)</td>
<td></td>
</tr>
<tr>
<td>Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning</td>
<td>MVAPICH2-GDR</td>
</tr>
<tr>
<td>Applications</td>
<td></td>
</tr>
<tr>
<td>Energy-aware MPI with Support for InfiniBand, Omni-Path, Ethernet/iWARP and,</td>
<td>MVAPICH2-EA</td>
</tr>
<tr>
<td>RoCE (v1/v2)</td>
<td></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>
OSU Microbenchmarks

• Available since 2004
• Suite of microbenchmarks to study communication performance of various programming models
• Benchmarks available for the following programming models
  – Message Passing Interface (MPI)
  – Partitioned Global Address Space (PGAS)
    • Unified Parallel C (UPC), Unified Parallel C++ (UPC++), and OpenSHMEM
• Benchmarks available for multiple accelerator-based architectures
  – Compute Unified Device Architecture (CUDA)
  – OpenACC Application Program Interface
• Part of various national resource procurement suites like NERSC-8 / Trinity Benchmarks
• Continuing to add support for newer primitives and features (latest OMB 5.8)
  – ROCm support
  – NCCL support
• Will be adding Python and Java MPI support in OMB
• Please visit the following link for more information: http://mvapich.cse.ohio-state.edu/benchmarks/
**OMB-Py: Initial CPU Results**

- osu_latency inter-node small messages
- osu_latency inter-node large messages

- Small Python overhead around 0.5 microseconds.
- Overhead only apparent in smaller message sizes
OMB-Py: Initial GPU Results

- Different overheads depending on the CUDA-aware data structure
- Overhead is only apparent in smaller message sizes

To be available with the next OMB release

More details in Tomorrow’s Talk by Nawras Alnaasan
Applications-Level Tuning: Compilation of Best Practices

• MPI runtime has many parameters
• Tuning a set of parameters can help you to extract higher performance
• Compiled a list of such contributions through the MVAPICH Website
  – [http://mvapich.cse.ohio-state.edu/best_practices/](http://mvapich.cse.ohio-state.edu/best_practices/)
• Initial list of applications
  – Amber
  – HoomDBlue
  – HPCG
  – Lulesh
  – MILC
  – Neuron
  – SMG2000
  – Cloverleaf
  – SPEC (LAMMPS, POP2, TERA_TF, WRF2)
• Soliciting additional contributions, send your results to mvapich-help at cse.ohio-state.edu.
• We will link these results with credits to you.
MVAPICH2 Libraries available through Spack

• Easy Installation of MVAPICH2 Libraries through Spack
  – MVAPICH2
  – MVAPICH2-X
  – MVAPICH2-GDR

• Detailed Spack-based Installation User Guide is available:
  http://mvapich.cse.ohio-state.edu/userguide/userguide_spack/

More details in
Tomorrow’s Talk
by Todd Gamblin (LLNL)
Enabling Auto detection of Underlying features with dlopen

• Use dlopen to identify support available in underlying system
  – Removes dependencies on libraries like
    • ibverbs
    • ibmad
    • ibumad
    • rdmacm
    • xpmem
  – Simplifies installation and deployment
    • No need for multiple configure options
    • One binary can have all necessary support

• Already available with the latest MVAPICH2 releases
• More such functionalities will be added in future releases
MVAPICH2-Next: One Stack to Conquer all Architectures and Interconnects

- Various libraries will be merged into one release
- Simplifies installation and deployment
- Enables utilizing the best advanced features for all architectures
MVAPICH2 – Plans for Exascale

- Performance and Memory scalability toward 1-10M cores
- Hybrid programming (MPI + OpenSHMEM, MPI + UPC, MPI + CAF ...)
  - MPI + Task*
- Enhanced Optimization for GPU Support and Accelerators
- Taking advantage of advanced features of Mellanox InfiniBand
  - Tag Matching*
  - Adapter Memory*
- Enhanced communication schemes for upcoming architectures
  - Intel Optane*
  - CAPI*
- Extended topology-aware collectives
- Extended Energy-aware designs and Virtualization Support
- Extended Support for MPI Tools Interface (as in MPI 3.0)
- Extended FT support
- Support for * features will be available in future MVAPICH2 Releases
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 being provided to National Laboratories and International HPC centers
Value-Added Products with Support

- Multiple value-added products with support
  - X-ScaleHPC
  - X-ScaleAI
  - MVAPICH2-DPU (Tutorial and Live Demo yesterday)
  - SCR-Exa

More details in Donglai Dai’s Talk (Tomorrow)
Funding Acknowledgments

Funding Support by

Equipment Support by
Acknowledgments to all the Heroes (Past/Current Students and Staffs)

**Current Students (Graduate)**
- N. Alnaasan (Ph.D.)
- Q. Anthony (Ph.D.)
- C.-C. Chun (Ph.D.)
- N. Contini (Ph.D.)
- A. Jain (Ph.D.)
- K. S. Khorassani (Ph.D.)
- A. H. Tu (Ph.D.)

**Past Students**
- A. Awan (Ph.D.)
- A. Augustine (M.S.)
- P. Balaji (Ph.D.)
- M. Bayatpour (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.)
- C.-H. Chu (Ph.D.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- J. Hashmi (Ph.D.)
- W. Huang (Ph.D.)
- W. Jiang (M.S.)
- J. Jose (Ph.D.)
- M. Kedia (M.S.)
- 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.)

**Current Research Scientists**
- 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. Potluri (Ph.D.)
- K. Raj (M.S.)
- R. Rajachandrasekar (Ph.D.)

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

**Past Senior Research Associate**
- J. Hashmi

**Past Post-Docs**
- D. Banerjee
- X. Besseron
- M. S. Ghazimeersaeed
- H.-W. Jin
- E. Mancini
- A. Ruhela
- J. Lin
- K. Manian
- J. Vienne
- M. Luo
- S. Marcarelli
- H. Wang

**Current Software Engineers**
- B. Seeds
- N. Shineman

**Current Students (Undergrads)**
- M. Lieber
- J. Zhang (Ph.D.)
- D. Shankar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- N. Sarkauskas (B.S.)
- N. Senthil Kumar (M.S.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Srivastava (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 Programmers**
- A. Reifsteck
- D. Bureddy
- J. Perkins

**Past Research Specialist**
- M. Arnold
- J. Smith
Multiple Positions Available in MVAPICH2, BigData and DL/ML Projects in my Group

- Looking for Bright and Enthusiastic Personnel to join as
  - PhD Students
  - Post-Doctoral Researchers
  - MPI Programmer/Software Engineer
  - Spark/Big Data Programmer/Software Engineer
  - Deep Learning, Machine 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

Network-Based Computing Laboratory

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

The High-Performance MPI/PGAS Project
http://mvapich.cse.ohio-state.edu/
Follow us on Twitter: @mvapich

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

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