Experiences in Designing, Developing, Packaging, and Deploying the MVAPICH2 Libraries in Spack

Talk at E4S Forum (September '20)

by

Hari Subramoni

The Ohio State University

E-mail: subramon@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~subramon

Follow us on

https://twitter.com/mvapich

Presentation Overview

- MVAPICH Project
  - MPI and PGAS Library with CUDA-Awareness for HPC and DL
  - MVAPICH on Cloud
- Deployment Solutions
  - Public Cloud Deployment
  - RPM/Debian-based Deployment
  - Spack-based Deployment
- Conclusions

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 (upcoming))**
- 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 3,100 organizations in 89 countries**
- **More than 862,000 (> 0.8 million) downloads from the OSU site directly**
- Empowering many TOP500 clusters (June '20 ranking)
  - 4th , 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China
  - 8th, 448, 448 cores (Frontera) at TACC
  - 12th, 391,680 cores (ABCI) in Japan
  - 18th, 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 8th ranked TACC Frontera system
- **Empowering Top500 systems for more than 15 years**

Architecture of MVAPICH2 Software Family (for HPC, DL, and 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, EFA, Omni-Path)

Transport Protocols

- RC
- XRC
- UD
- DC

Modern Features

- SHARP2*
- ODP
- SR-IOV
- Multi Rail

Support for Modern Multi-/Many-core Architectures (Intel-Xeon, OpenPower, Xeon-Phi, ARM, NVIDIA & AMD GPUs)

Transport Mechanisms

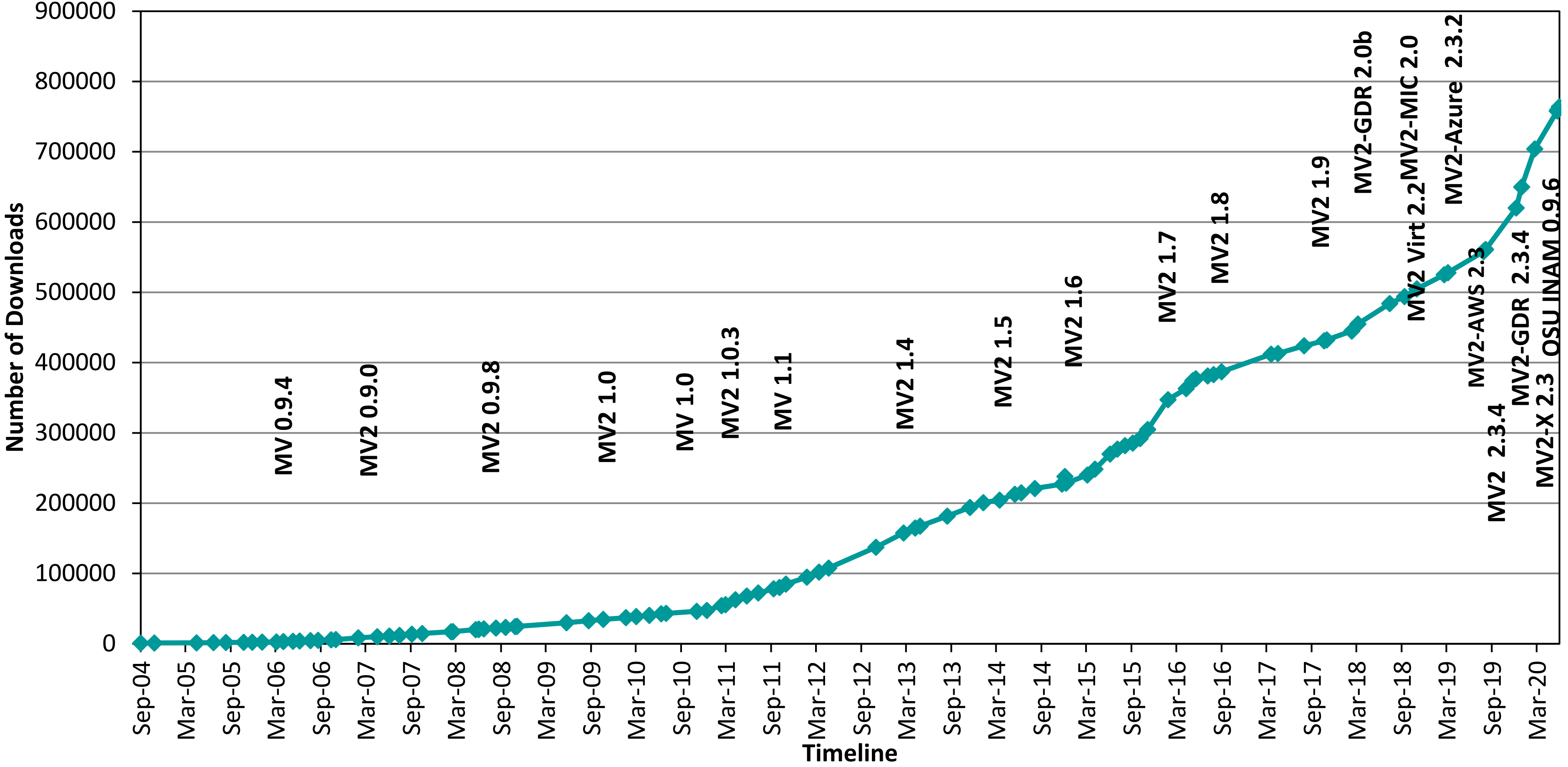
- Shared Memory
- CMA
- IVSHMEM
- XPMEM

Modern Features

- MCDRAM*
- NVLink
- CAPI*

* Upcoming

MVAPICH2 Release Timeline and Downloads



Production Quality Software Design, Development and Release

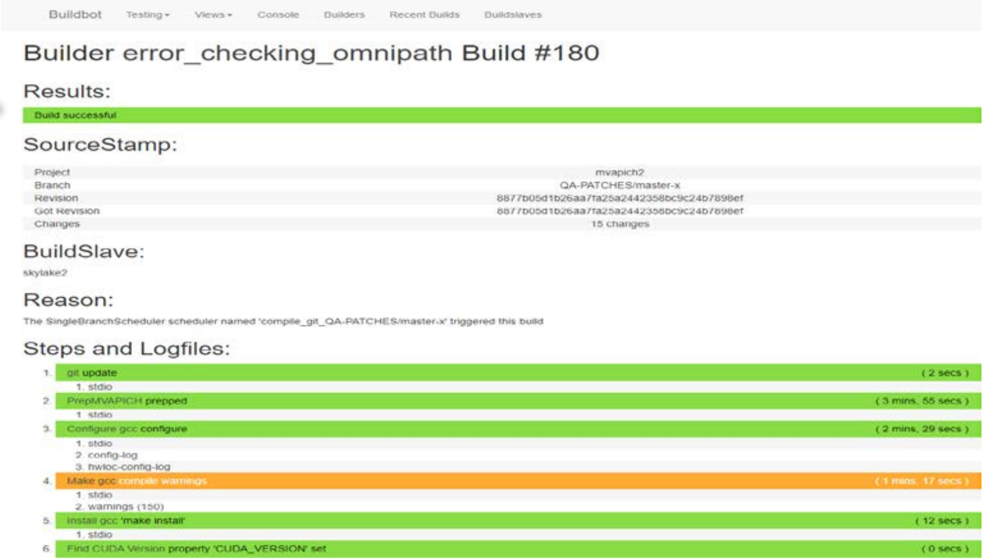
- Rigorous Q&A procedure before making a release
  - Exhaustive unit testing
  - Various test procedures on diverse range of platforms and interconnects
  - Test 19 different benchmarks and applications including, but not limited to
    - OMB, IMB, MPICH Test Suite, Intel Test Suite, NAS, Scalapak, and 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

Automated Process for Performing Builds

- Use automated process built on “buildbot” infrastructure to perform various types of builds
  - Compiler
    - GNU, Intel, PGI, Clang
  - Network Interconnect
    - Intel Omni-Path, QLogic PSM, InfiniBand, Ethernet
  - Different debugging levels
  - Different compile time options
- Ability to build on remote HPC systems
  - e.g. Frontera@TACC

Summary page of various builds

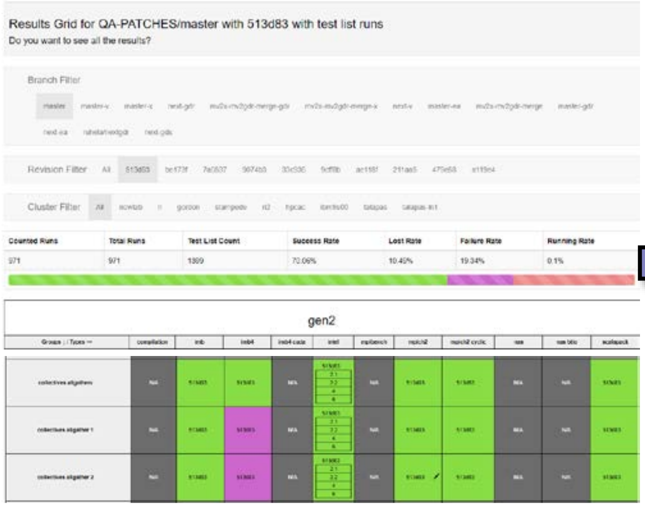
Results of individual builds



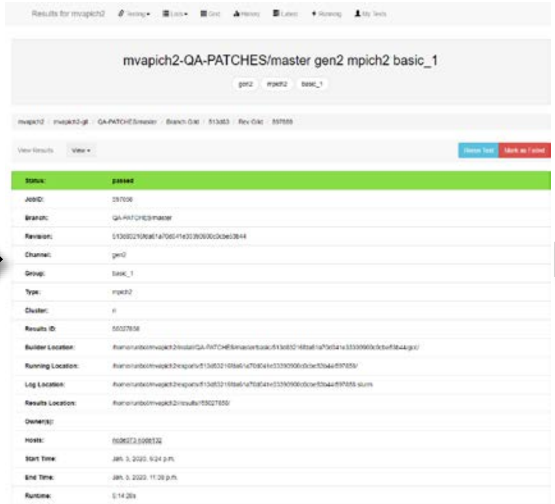
Automated Procedure for Testing Functionality

- Test OMB, IMB, MPICH Test Suite, Intel Test Suite, NAS, Scalapak, and SPEC
- Tests done for each build done build “buildbot”
- Test done for various different combinations of *environment variables* meant to trigger different communication paths in MVAPICH2

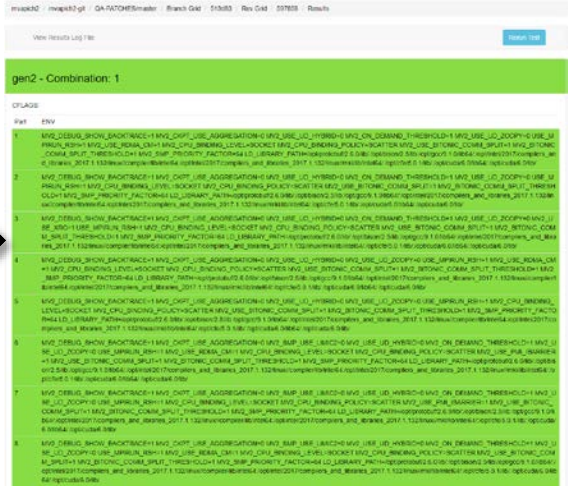
Summary of all tests for one commit



Summary of an individual test



Details of individual combinations in one test



Scripts to Determine Performance Regression

- Automated method to identify performance regression between different commits
- Tests different MPI primitives
  - Point-to-point; Collectives; RMA
- Works with different
  - Job Launchers/Schedulers
    - SLURM, PBS/Torque, JSM
  - Works with different interconnects
- Works on multiple HPC systems
- Works on CPU-based and GPU-based systems

Performance regression of mvapich2-2.3rc2-x-3e5551 and mvapich2-masterx-2950c8 on FRONTERA (cascadelake architecture) Thu Aug 15 09:23:48 CDT 2019

OLD_TUNEVAR=

NEW_TUNEVAR=

Legend

Dark Green : Performance of mvapich2-masterx-2950c8 is more than 5 % better than mvapich2-2.3rc2-x-3e5551

Light Green : Performance of mvapich2-masterx-2950c8 is less than 5 % better than mvapich2-2.3rc2-x-3e5551

Grey : Performance of mvapich2-masterx-2950c8 is same as mvapich2-2.3rc2-x-3e5551

Light Red : Performance of mvapich2-masterx-2950c8 is less than 5 % worse compared to mvapich2-2.3rc2-x-3e5551

Dark Red : Performance of mvapich2-masterx-2950c8 is more than 5 % worse compared to mvapich2-2.3rc2-x-3e5551

Inter-node

getbw	1 4.12 4.24 2 %	2 8.13 8.52 4 %	4 16.36 17.15 4 %	8 32.57 34.35 5 %	16 65.19 68.65 5 %	32 133.07 137.11 2 %	64 254.43 268.90 5 %	128 512.64 529.31 3 %	256 1013.21 1047.48 3 %	512 1959.28 2034.04 3 %	1K 3806.50 3888.42 2 %	2K 6495.99 6540.39 0 %	4K 7933.33 8009.98 0 %	8K 10118.37 10075.08 0 %
putbibw	1 7.32 7.18 -1 %	2 14.79 14.52 -1 %	4 29.56 29.05 -1 %	8 58.98 58.02 -1 %	16 117.73 116.10 -1 %	32 219.70 221.33 0 %	64 438.70 442.21 0 %	128 862.87 852.53 -1 %	256 1821.56 1795.21 -1 %	512 3505.96 3539.99 0 %	1K 6728.97 6714.72 0 %	2K 11697.93 11634.75 0 %	4K 15297.26 15186.84 0 %	8K 18873.10 18919.87 0 %
putbw	1 4.22 4.32 2 %	2 8.41 8.64 2 %	4 17.11 17.43 1 %	8 34.13 34.84 2 %	16 68.41 69.74 1 %	32 127.38 132.60 3 %	64 253.67 263.58 4 %	128 494.66 509.46 2 %	256 1051.14 1080.49 2 %	512 1987.26 2079.78 4 %	1K 3801.76 3978.30 4 %	2K 6357.54 6438.37 1 %	4K 8422.30 8478.65 0 %	8K 9972.75 10031.73 0 %
acclat	1 2.30 2.30 0 %	2 2.30 2.30 0 %	4 2.30 2.30 0 %	8 2.31 2.31 0 %	16 2.47 2.50 -1 %	32 2.51 2.53 0 %	64 2.91 2.93 0 %	128 2.97 2.99 0 %	256 3.09 3.11 0 %	512 3.33 3.35 0 %	1K 3.82 3.85 0 %	2K 4.79 4.81 0 %	4K 7.03 6.96 0 %	8K 10.76 10.79 0 %
getlat	1 1.96 1.96 0 %	2 1.97 1.96 0 %	4 1.96 1.95 0 %	8 1.97 1.95 1 %	16 1.97 1.95 1 %	32 1.97 1.95 1 %	64 1.97 1.95 1 %	128 2.01 2.01 0 %	256 2.07 2.07 0 %	512 2.11 2.13 0 %	1K 2.23 2.24 0 %	2K 2.51 2.50 0 %	4K 3.07 3.12 -1 %	8K 3.79 3.76 0 %
putlat	1 1.59 1.59 0 %	2 1.58 1.59 0 %	4 1.59 1.58 0 %	8 1.59 1.58 0 %	16 1.59 1.59 0 %	32 1.63 1.63 0 %	64 1.62 1.62 0 %	128 1.64 1.64 0 %	256 1.98 1.97 0 %	512 2.03 2.02 0 %	1K 2.12 2.11 0 %	2K 2.33 2.32 0 %	4K 2.85 2.87 0 %	8K 3.58 3.62 -1 %
lat	0 1.09 1.10 0 %	1 1.12 1.12 0 %	2 1.12 1.12 0 %	4 1.12 1.12 0 %	8 1.12 1.12 0 %	16 1.16 1.16 0 %	32 1.16 1.16 0 %	64 1.18 1.18 0 %	128 1.23 1.23 0 %	256 1.64 1.63 0 %	512 1.73 1.72 0 %	1K 1.89 1.90 0 %	2K 2.27 2.26 0 %	4K 3.30 3.16 4 %
bibw	1 0.01 6.02 99 %	2 8.84 12.41 28 %	4 17.67 24.90 29 %	8 35.26 50.43 30 %	16 69.95 102.34 31 %	32 139.10 209.77 33 %	64 272.80 441.25 38 %	128 527.61 906.82 41 %	256 1014.76 1649.39 38 %	512 1906.89 2984.43 36 %	1K 3512.06 5576.41 37 %	2K 5983.62 7606.62 21 %	4K 8808.00 2974.96 -196 %	8K 12102.21 11577.63 -4 %
bw	1 4.30 5.62 23 %	2 8.65 11.37 23 %	4 18.55 22.70 18 %	8 37.12 45.17 17 %	16 70.05 88.18 20 %	32 140.56 169.93 17 %	64 277.59 361.68 23 %	128 535.34 759.80 29 %	256 992.70 1471.40 32 %	512 1834.85 2313.06 26 %	1K 3403.25 4894.56 12 %	2K 5539.04 5642.22 1 %	4K 7134.90 7440.25 4 %	8K 9246.59 9152.05 -1 %
mbw_mr	1 4643958.72 5611922.55 17 %	2 4652819.64 5672494.10 17 %	4 4660101.58 5656039.25 17 %	8 4651994.45 5642174.30 17 %	16 4408316.86 5511754.90 20 %	32 4386219.34 5312786.53 17 %	64 4343210.34 5624117.68 22 %	128 4181690.38 5945467.49 29 %	256 3961422.35 5710061.67 30 %	512 3582486.69 4904271.86 26 %	1K 3358575.31 3815154.75 11 %	2K 2709562.74 2757052.57 1 %	4K 1751091.52 1816130.44 3 %	8K 1128687.30 1117241.92 -1 %

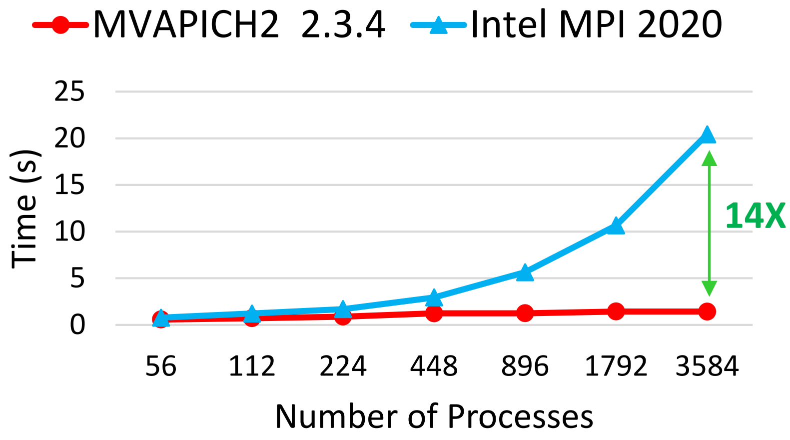
128 process collective tests

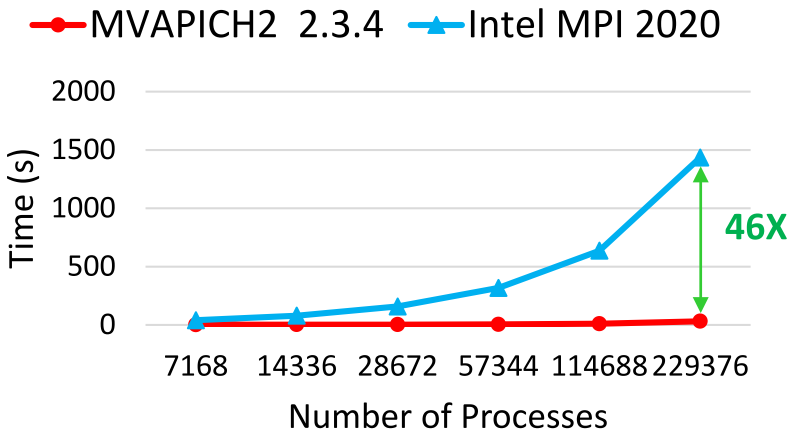
osu_allgather	1 21.83 17.34 20 %	2 23.72 16.16 31 %	4 22.11 17.28 21 %	8 24.40 17.87 26 %	16 28.29 19.89 29 %	32 38.99 21.65 44 %	64 28.70 26.24 8 %	128 47.85 43.90 8 %	256 66.45 63.03 5 %	512 105.64 102.18 3 %	1K 914.13 177.28 80 %	2K 2330.43 276.96 88 %	4K 726.91 502.60 30 %	8K 1197.84 1272.69 -6 %	16K 2460.79 2555.89 -3 %	32K 3996.50 3247.50 18 %	64K 6274.14 5516.63 12 %	128K 13385.82 10428.17 22 %	256K 20382.71 20449.20 0 %	512K 43029.04 43000.04 0 %	1M 102436.87 88526.72 13 %
osu_allgatherv	1 25.66 21.91 14 %	2 27.61 22.04 20 %	4 25.41 19.47 23 %	8 27.68 20.73 25 %	16 31.34 23.41 25 %	32 40.98 25.06 38 %	64 78.11 29.32 62 %	128 133.69 50.28 62 %	256 207.27 133.06 35 %	512 505.08 233.32 53 %	1K 529.16 239.41 54 %	2K 558.27 284.32 49 %	4K 689.62 525.36 23 %	8K 1176.09 1254.16 -6 %	16K 2461.61 2524.56 -2 %	32K 3898.46 3167.98 18 %	64K 6573.11 5468.24 16 %	128K 11721.43 10026.13 14 %	256K 21588.64 20060.80 7 %	512K 43244.00 43166.45 0 %	1M 102924.40 87885.28 14 %
osu_allreduce	4 36.82 19.08 48 %	8 32.90 22.28 32 %	16 32.66 19.37 40 %	32 33.15 19.37 41 %	64 35.42 21.29 39 %	128 36.90 27.06 26 %	256 41.44 22.96 44 %	512 51.83 29.01 44 %	1K 69.81 37.88 45 %	2K 107.22 60.36 43 %	4K 48.50 44.66 7 %	8K 51.63 48.42 6 %	16K 86.36 84.79 1 %	32K 120.66 109.31 9 %	64K 184.02 172.09 6 %	128K 296.64 267.65 9 %	256K 521.36 469.93 9 %	512K 1996.51 967.96 51 %	1M 3221.70 2649.74 17 %	2M 6665.53 6305.41 5 %	4M 9984.70 8972.08 10 %

MVAPICH2 Software Family (CPU-Based Deep Learning)

High-Performance Parallel Programming Libraries	
MVAPICH2	Support for InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE
MVAPICH2-X	Advanced MPI features, OSU INAM, PGAS (OpenSHMEM, UPC, UPC++, and CAF), and MPI+PGAS programming models with unified communication runtime
MVAPICH2-GDR	Optimized MPI for clusters with NVIDIA GPUs and for GPU-enabled Deep Learning Applications
MVAPICH2-Virt	High-performance and scalable MPI for hypervisor and container based HPC cloud
MVAPICH2-EA	Energy aware and High-performance MPI
MVAPICH2-MIC	Optimized MPI for clusters with Intel KNC
Microbenchmarks	
OMB	Microbenchmarks suite to evaluate MPI and PGAS (OpenSHMEM, UPC, and UPC++) libraries for CPUs and GPUs
Tools	
OSU INAM	Network monitoring, profiling, and analysis for clusters with MPI and scheduler integration
OEMT	Utility to measure the energy consumption of MPI applications

Startup Performance on TACC Frontera



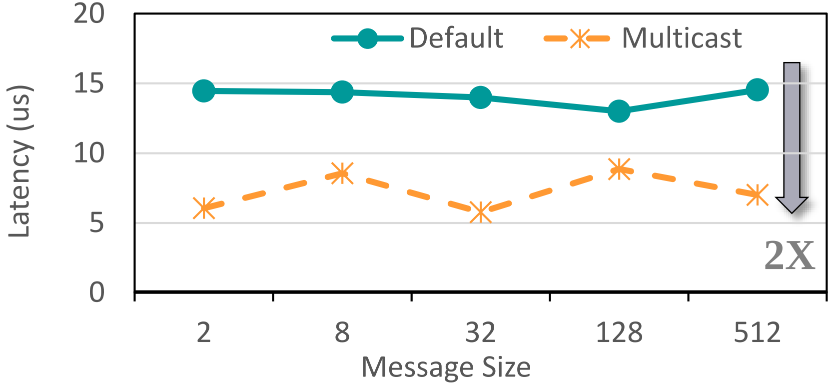


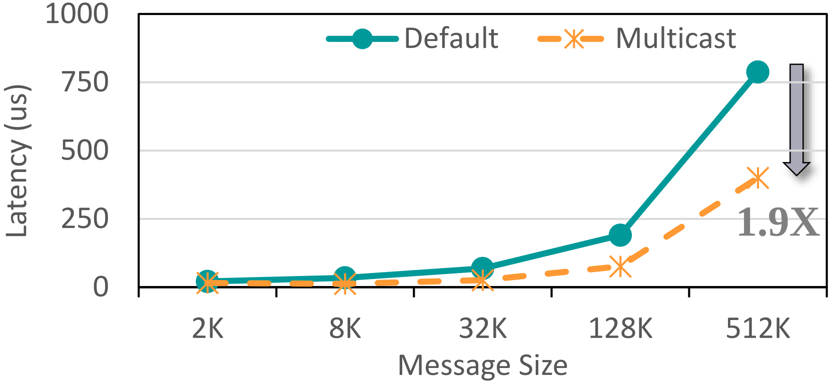
- MPI_Init takes 31 seconds on 229,376 processes on 4,096 nodes
- All numbers reported with 56 processes per node

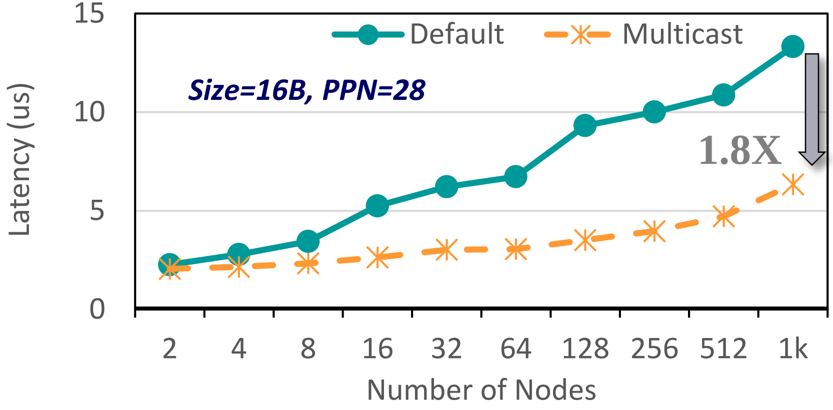
New designs available in MVAPICH2-2.3.4

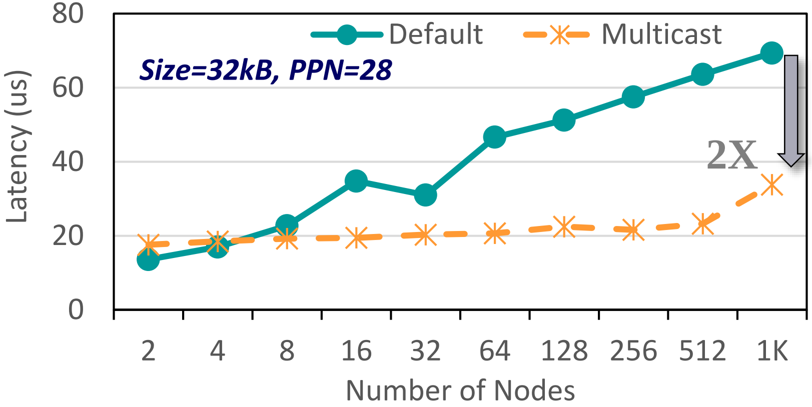
Hardware Multicast-aware MPI_Bcast on TACC Frontera

(Nodes=2K, PPN=28)



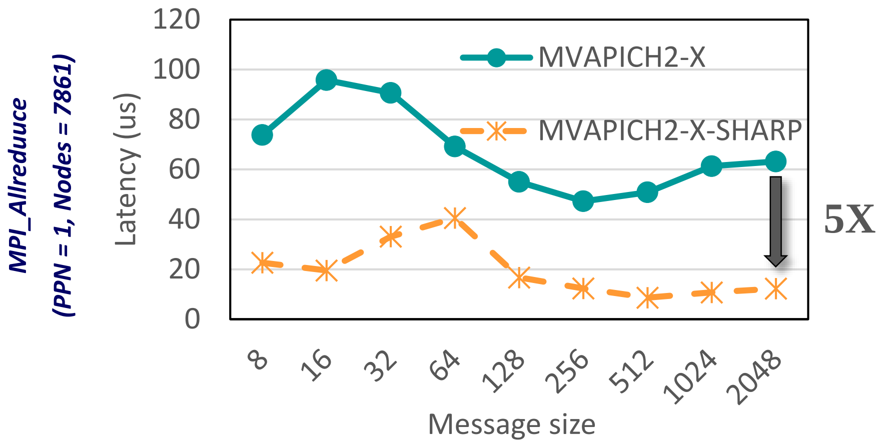


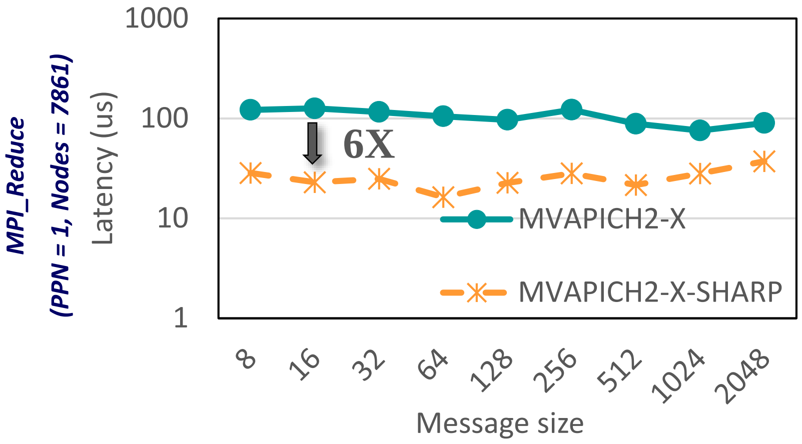


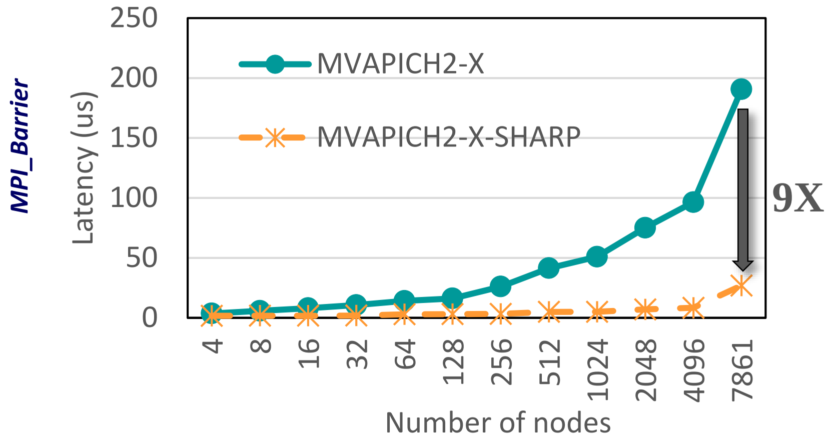


- MCAST-based designs improve latency of MPI_Bcast by up to **2X at 2,048 nodes**
- Use MV2_USE_MCAST=1 to enable MCAST-based designs

Performance of Collectives with SHARP on TACC Frontera







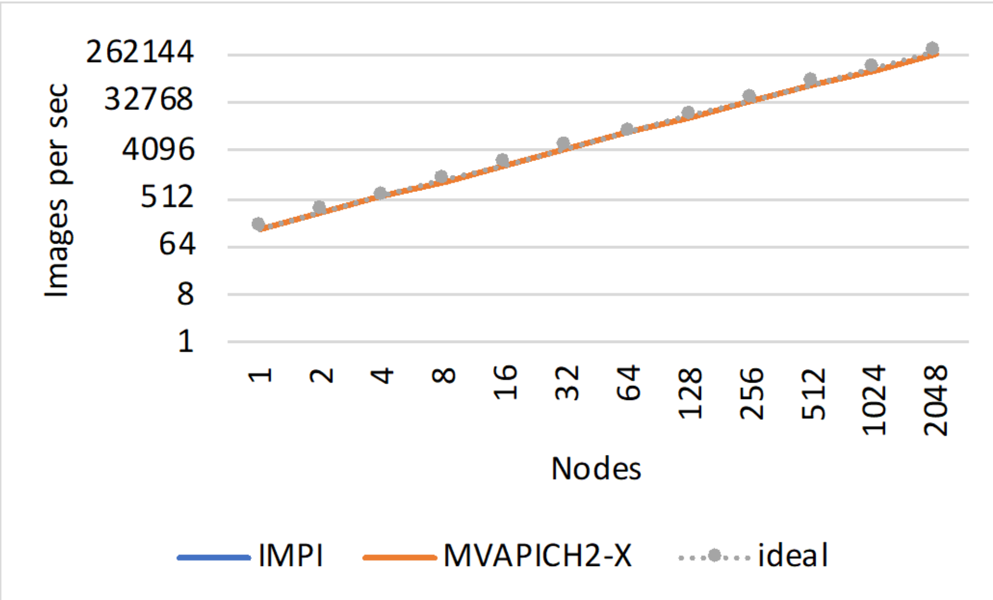
Optimized SHARP designs in MVAPICH2-X

– ***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

Distributed TensorFlow on TACC Frontera (2,048 CPU nodes)

- Scaled TensorFlow to 2048 nodes on Frontera using MVAPICH2 and IntelMPI
- MVAPICH2 and IntelMPI give similar performance for DNN training
- Report a peak of 260,000 images/sec on 2,048 nodes
- On 2048 nodes, ResNet-50 can be trained in 7 minutes!

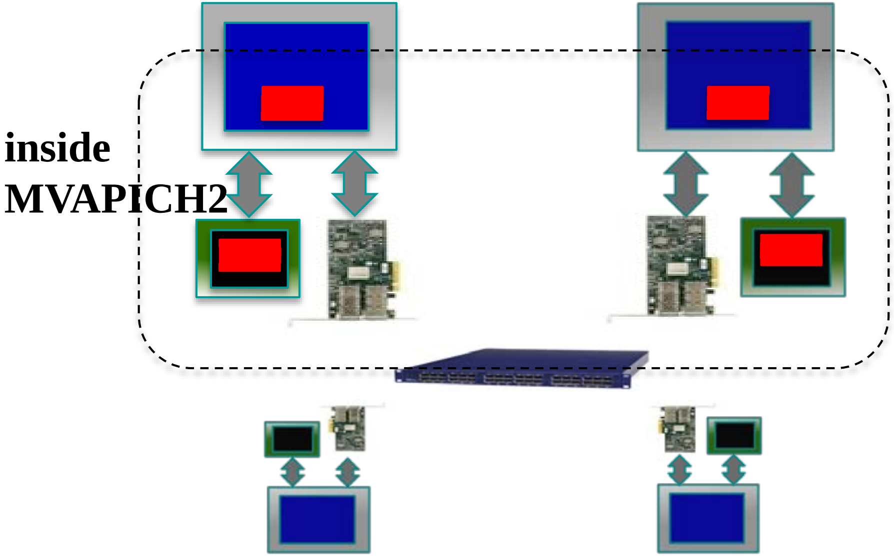


A. Jain, A. A. Awan, H. Subramoni, DK Panda, "Scaling TensorFlow, PyTorch, and MXNet using MVAPICH2 for High-Performance Deep Learning on Frontera", DLS '19 (SC '19 Workshop).

GPU-Aware (CUDA-Aware) MPI Library: MVAPICH2-GPU

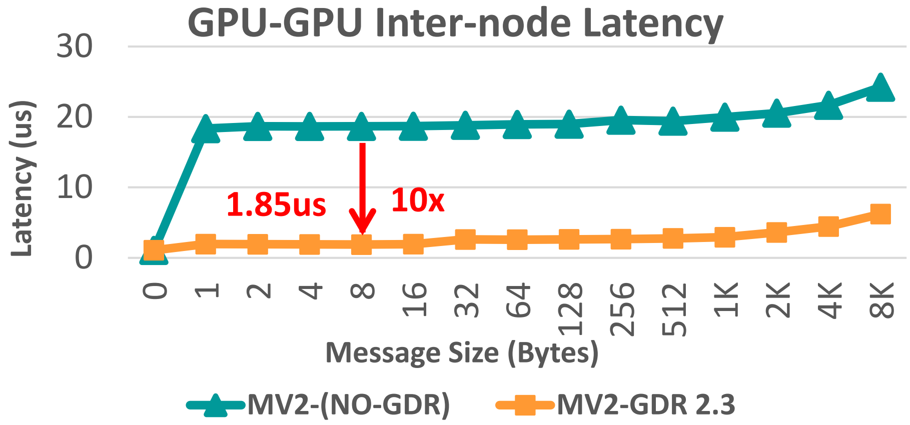
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

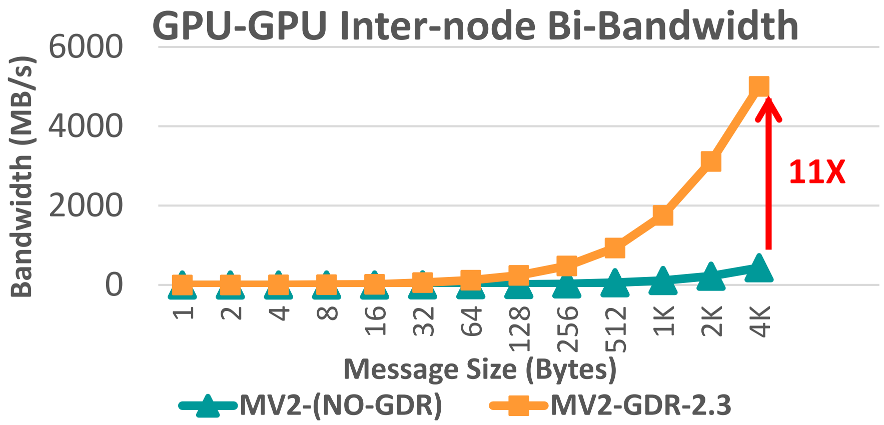
At Sender: MPI_Send(s_devbuf, size, …); **At Receiver:** MPI_Recv(r_devbuf, size, …);

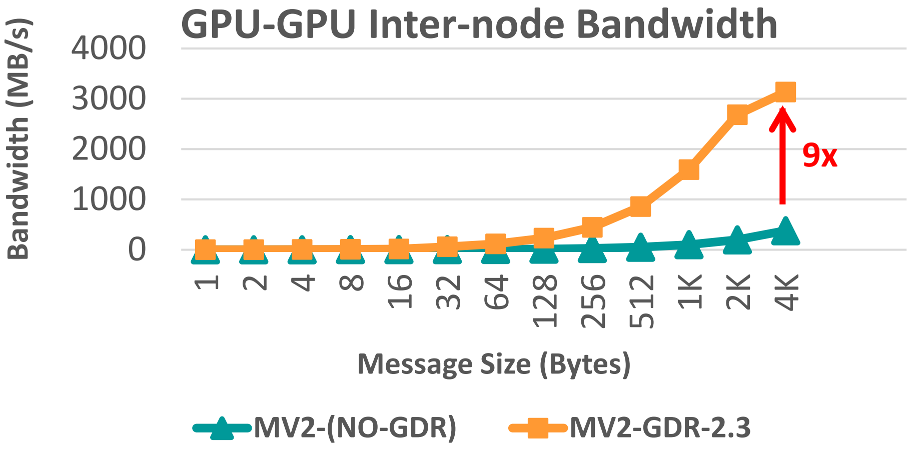


High Performance and High Productivity

Optimized MVAPICH2-GDR Design

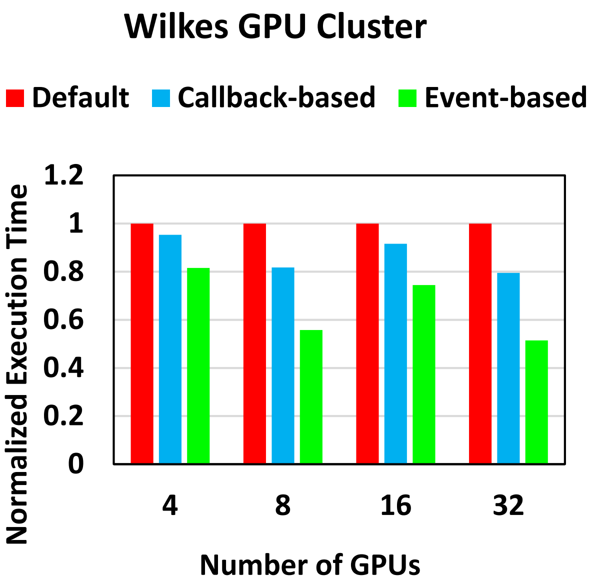


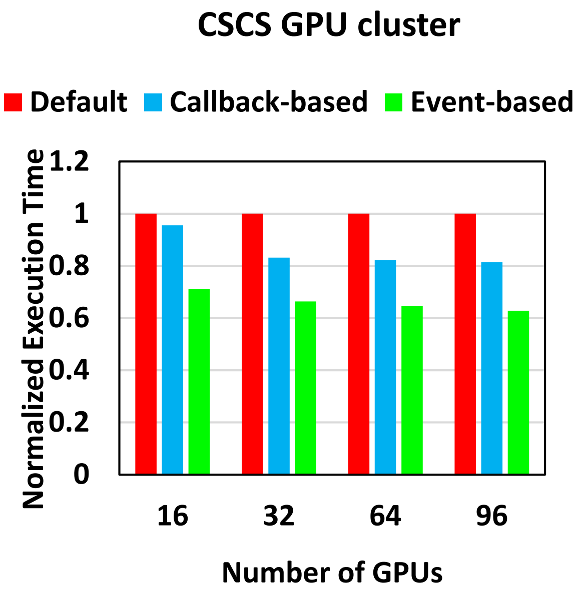


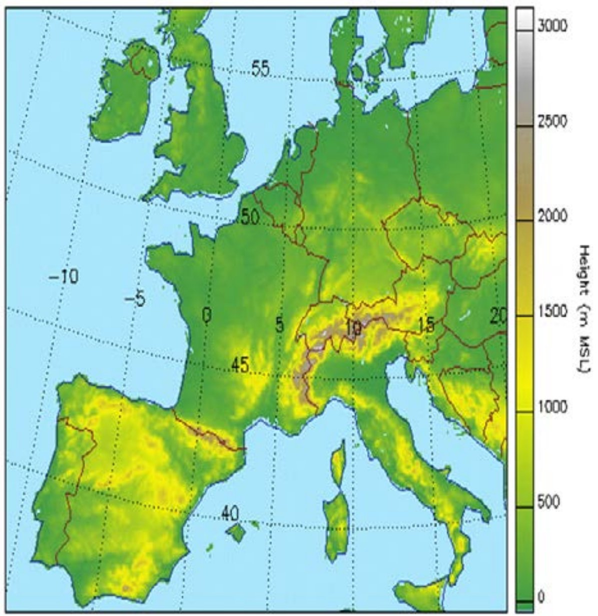


MVAPICH2-GDR-2.3
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

Application-Level Evaluation (Cosmo) and Weather Forecasting in Switzerland







- 2X improvement on 32 GPUs nodes
- 30% improvement on 96 GPU nodes (8 GPUs/node)

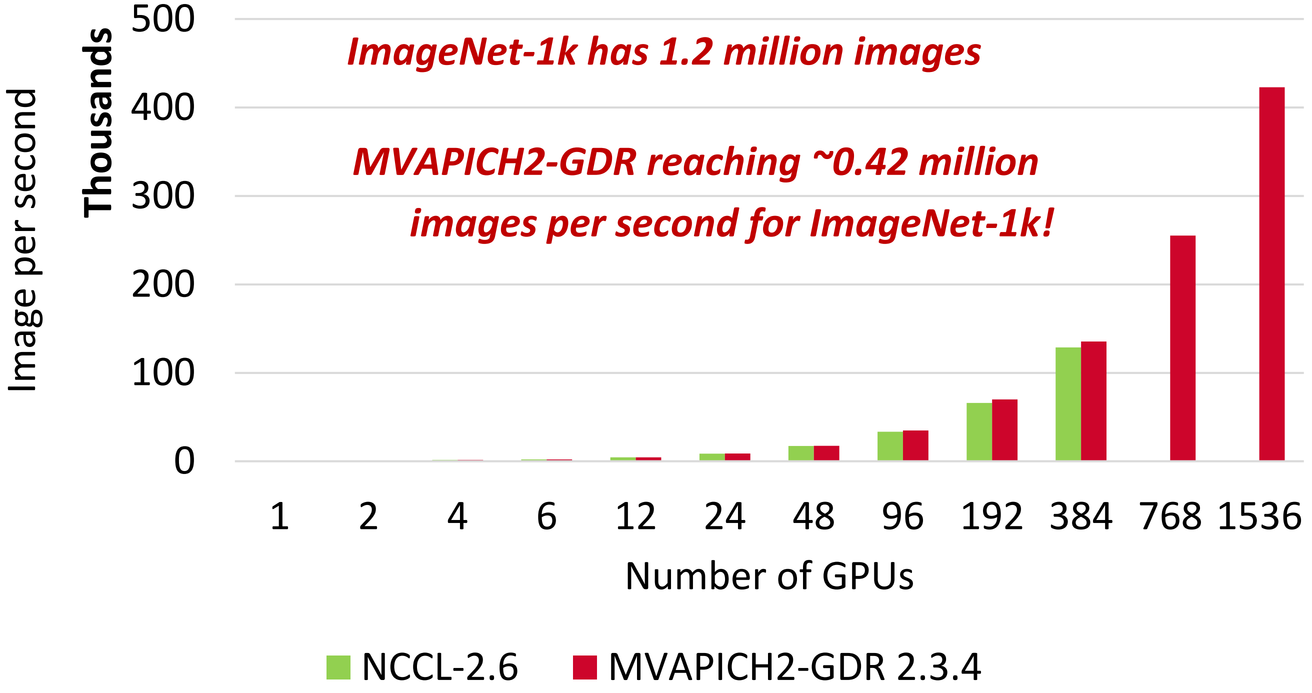
Cosmo model: http://www2.cosmo-model.org/content/tasks/operational/meteoSwiss/

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

C. Chu, K. Hamidouche, A. Venkatesh, D. Banerjee , H. Subramoni, and D. K. Panda, Exploiting Maximal Overlap for Non-Contiguous Data Movement Processing on Modern GPU-enabled Systems, IPDPS'16

Distributed TensorFlow on ORNL Summit (1,536 GPUs)

- ResNet-50 Training using TensorFlow benchmark on SUMMIT -- 1536 Volta GPUs!
- 1,281,167 (1.2 mil.) images
- Time/epoch = 3 seconds
- Total Time (90 epochs) = 3 x 90 = 270 seconds = **4.5 minutes!**

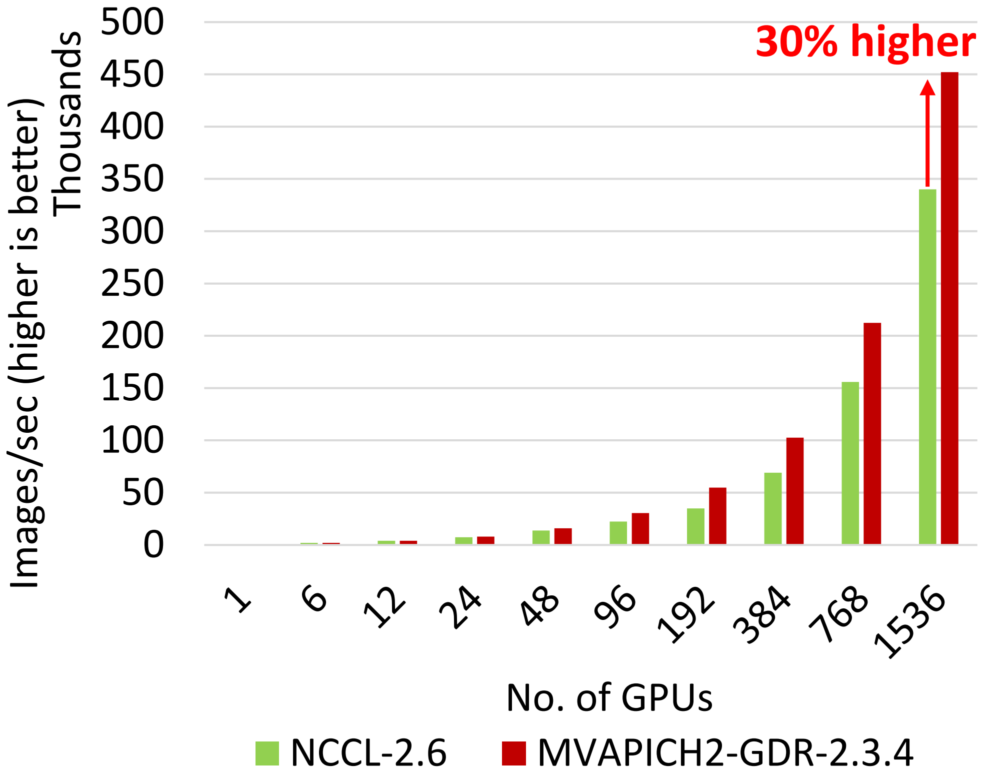


*We observed issues for NCCL2 beyond 384 GPUs

Platform: The Summit Supercomputer (#2 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 10.1

Scaling PyTorch on ORNL Summit using MVAPICH2-GDR

- ResNet-50 training using PyTorch + Horovod on Summit
  - Synthetic ImageNet dataset
  - Up to 256 nodes, 1536 GPUs
- MVAPICH2-GDR can outperform NCCL2
  - **Up to 30% higher throughput**
- CUDA 10.1 cuDNN 7.6.5
  PyTorch v1.5.0 Horovod v0.19.1



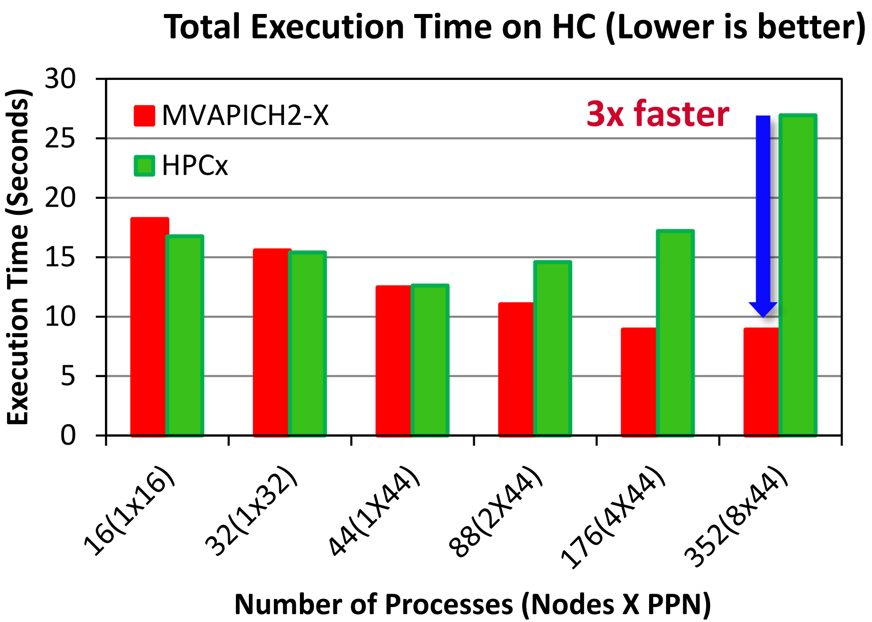
C.-H. Chu, P. Kousha, A. Awan, K. S. Khorassani, H. Subramoni and D. K. Panda, "NV-Group: Link-Efficient Reductions for Distributed Deep Learning on Modern Dense GPU Systems, " ICS-2020, June-July 2020.

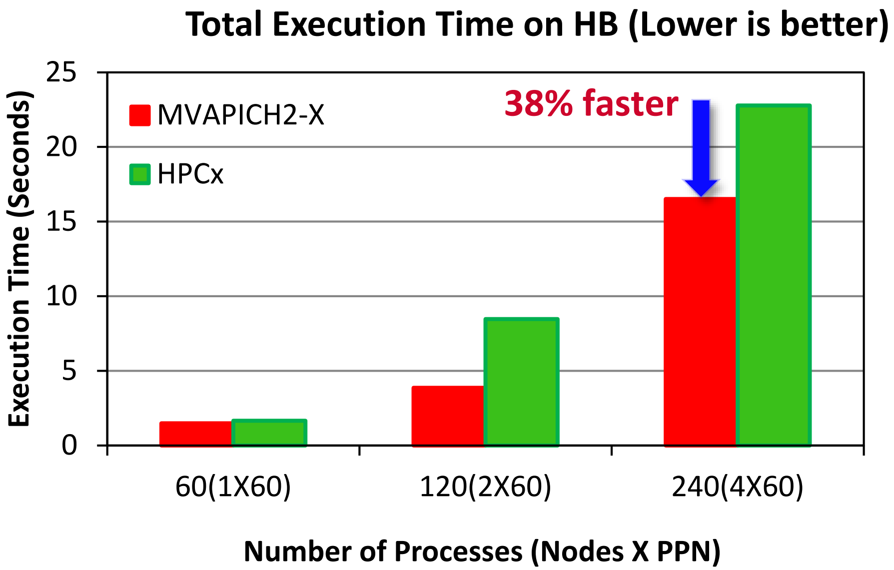
Platform: The Summit Supercomputer (#2 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 10.1

Presentation Overview

- MVAPICH Project
  - MPI and PGAS Library with CUDA-Awareness for HPC and DL
  - MVAPICH on Cloud
- Deployment Solutions
  - Public Cloud Deployment
  - RPM/Debian-based Deployment
  - Spack-based Deployment
- Conclusions

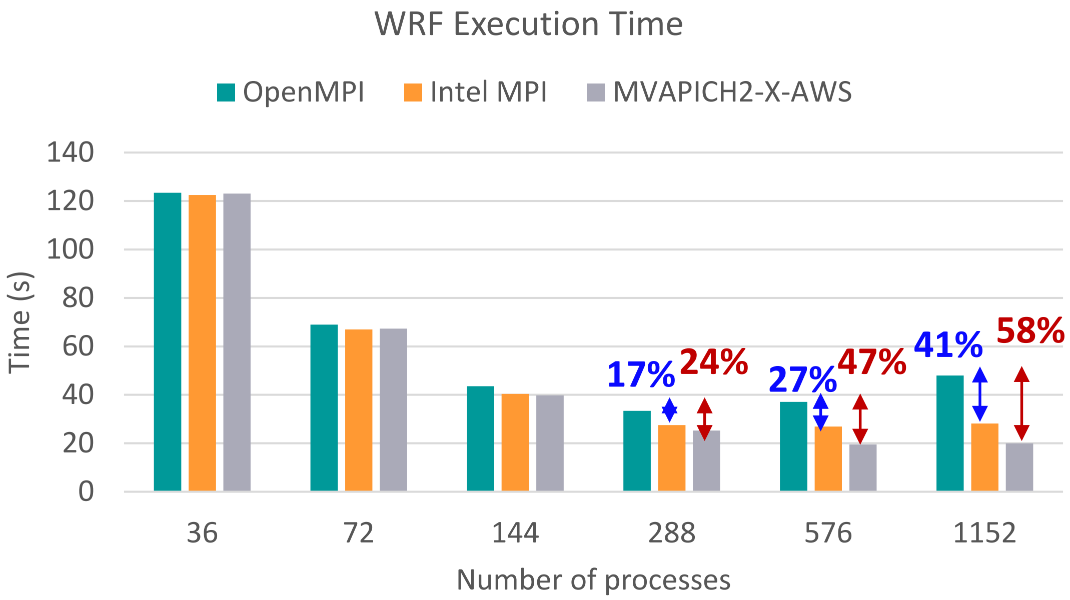
Performance of Radix on Azure





WRF Application Results

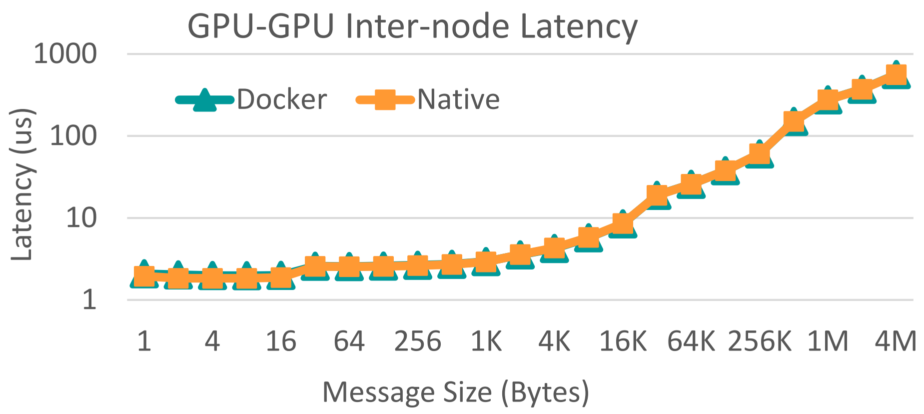
- Performance of WRF with Open MPI 4.0.3 vs Intel MPI 2019.7.217 vs MVAPICH2-X-AWS v2.3

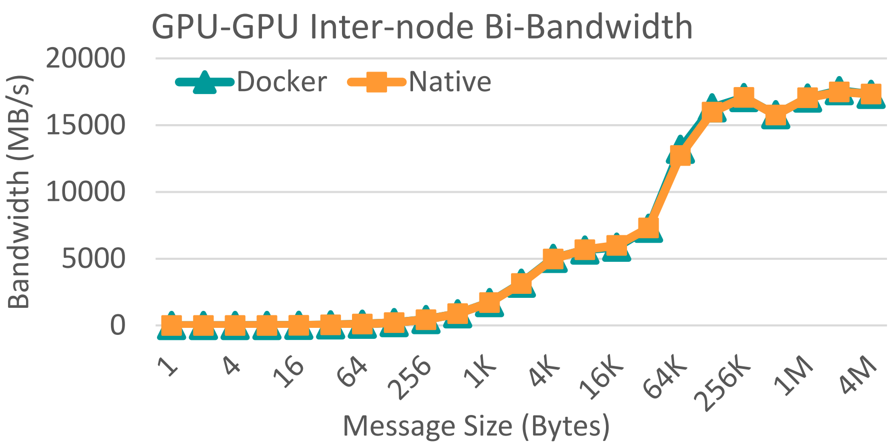


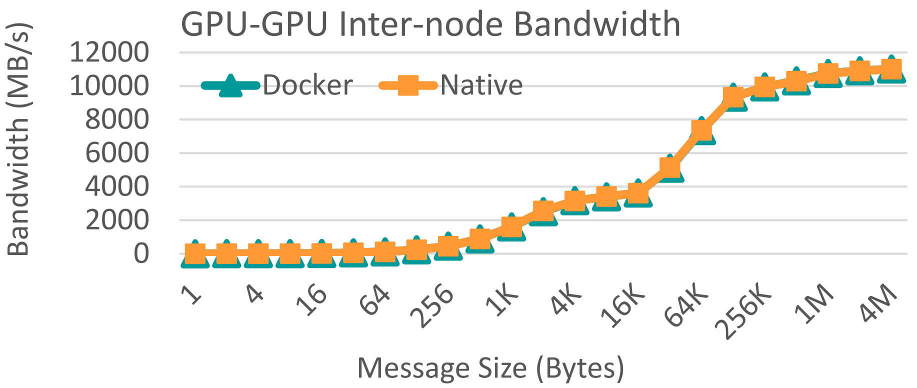
Benchmark Details

- WRF 3.6
  - https://github.com/hanschen/WRFV3
- Benchmark: 12km resolution case over the Continental U.S. (CONUS) domain
  - https://www2.mmm.ucar.edu/wrf/WG2/benchv3/#_Toc212961288
- Update io_form_history in namelist.input to 102
  - https://www2.mmm.ucar.edu/wrf/users/namelist_best_prac_wrf.html#io_form_history

MVAPICH2-GDR on Container with Negligible Overhead







MVAPICH2-GDR-2.3.2
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

Presentation Overview

- MVAPICH Project
  - MPI and PGAS Library with CUDA-Awareness for HPC and DL
  - MVAPICH on Cloud
- **Deployment Solutions**
  - **Public Cloud Deployment**
  - **RPM/Debian-based Deployment**
  - Spack-based Deployment
- Conclusions

MVAPICH2-Azure Deployment

- **Released on 05/20/2020**
- Integrated Azure CentOS HPC Images
  - **https://github.com/Azure/azhpc-images/releases/tag/centos-7.6-hpc-20200417**
- MVAPICH2 2.3.3
  - **CentOS Images (7.6, 7.7 and 8.1)**
  - **Tested with multiple VM instances**
- MVAPICH2-X 2.3.RC3
  - **CentOS Images (7.6, 7.7 and 8.1)**
  - **Tested with multiple VM instances**
- **More details from Azure Blog Post**
  - **https://techcommunity.microsoft.com/t5/azure-compute/mvapich2-on-azure-hpc-clusters/ba-p/1404305**

MVAPICH2-X-AWS 2.3

- **Released on 08/12/2019**
- Major Features and Enhancements
  - **Based on MVAPICH2-X 2.3**
  - **New design based on Amazon EFA adapter's Scalable Reliable Datagram (SRD) transport protocol**
  - **Support for XPMEM based intra-node communication for point-to-point and collectives**
  - **Enhanced tuning for point-to-point and collective operations**
  - **Targeted for AWS instances with Amazon Linux 2 AMI and EFA support**
  - **Tested with c5n.18xlarge instance**
- **New Release coming out today!!!!**

RPM and Debian Deployments

- Provide customized RPMs for different system requirements
  - ARM, Power8, Power9, x86 (Intel and AMD)
  - Different versions of Compilers (ICC, PGI, GCC, XLC, ARM), CUDA, OFED/Intel IFS

Deploy to Azure

MVAPICH2-GDR 2.3.2 Library

- The MVAPICH2-GDR library is distributed under the BSD License.
- OSU MVAPICH2-GDR 2.3.2 (08/08/2019), ABI compatible with MPICH-3.2.1
  - CHANGELOG for MVAPICH2-GDR 2.3.2.
- These RPMs contain the MVAPICH2-GDR software on the corresponding distro. **Please note that the RHEL RPMs are compatible with CentOS as well. For Debian/Ubuntu users, please follow the instructions in the install section in the userguide.**

OpenPOWER RPMs

	GNU 4.9.3	GNU 4.9.3 (w/ jsrun)	GNU 7.3.1	GNU 7.3.1 (w/ jsrun)	PGI 18.7	PGI 18.7 (w/ jsrun)	PGI 19.4	PGI 19.4 (w/ jsrun)
MLNX-OFED 4.3(Lassen/Sierra)	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	[CUDA 10.1]	[CUDA 10.1]

	GNU 4.8.5	GNU 6.4.0	GNU 7.4.0	PGI 18.7	PGI 19.4
MLNX-OFED 4.5(Summit)	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	[CUDA 10.1]

RHEL/CENTOS 7 RPMs

	GNU 4.8.5 (w/o SLURM)	GNU 4.8.5 (w/ SLURM)	GNU 4.8.5 (w/ PBS)	PGI (w/o SLURM)	PGI (w/ SLURM)	PGI (w/ PBS)
MLNX-OFED 4.X*	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2]	[CUDA 9.2] [CUDA 10.1]	[CUDA 9.2] [CUDA 10.1]	

*Note that the MOFED 3.X RPMs were built against MOFED 3.4 and the MOFED 4.X RPMs were built against MOFED 4.5
- However, these RPMs should work against the other MOFEDs with the same major MOFED version number
- e.g MOFED 4.X RPMs should work if you have MOFED 4.0, MOFED 4.2, MOFED 4.4, or MOFED 4.5
- Please email mvapich-help@cse.ohio-state.edu if you encounter any issues

MVAPICH2-X 2.3rc2 Library and User Guide

- The MVAPICH2-X 2.3rc2 library is distributed under the BSD License.
- OSU MVAPICH2-X 2.3rc2 (04/02/19), ABI compatible with MPICH-3.2.
  - CHANGELOG for MVAPICH2-X 2.3rc2
  - Patch to add PMI Extensions with SLURM 15
  - Patch to add PMI Extensions with SLURM 16
  - Patch to add PMI Extensions with SLURM 17
- MVAPICH2-X User Guide: A detailed user guide with instructions to install MVAPICH2-X and execute MPI/UPC/UPC++/OpenSHMEM/CAF/Hybrid programs is available. (HTML, PDF)
- **Installation Guide**
  - These tarballs contain the MVAPICH2-X software for Redhat and Debian based systems combined together in one combined package.
  - Running the install.sh script in the tarball will install the libraries.
  - These RPMs are relocatable and advanced users may skip the install.sh script to directly use alternate commands to install the desired RPMs.
- **Which RPM should I install?**
  - InfiniBand / RoCE System
  - Omni-Path System
- **Advanced Install Options**
  - Install library using a prefix other than the default of /opt/mvapich2/.

```
$ rpm --prefix /custom/install/prefix -Uvh --nodeps mvapich2-x-basic-mofed3.4-gnu4.8.5-2.3rc2-1.el7.centos.x86_64.rpm
```

  - If you do not have root permission or are on a system that does not use RPMs you can use rpm2cpio to extract the library.

```
$ rpm2cpio mvapich2-x-basic-mofed3.4-gnu4.8.5-2.3rc2-1.el7.centos.x86_64.rpm | cpio -id
```

  When using the rpm2cpio method, you will need to update the MPI compiler scripts, such as mpicc, in order to point to the correct path of where you place the library.
  - Tip: If you are using a Debian based system such as Ubuntu you can convert the rpm to a deb using a tool such as alien or follow the rpm2cpio instructions above.

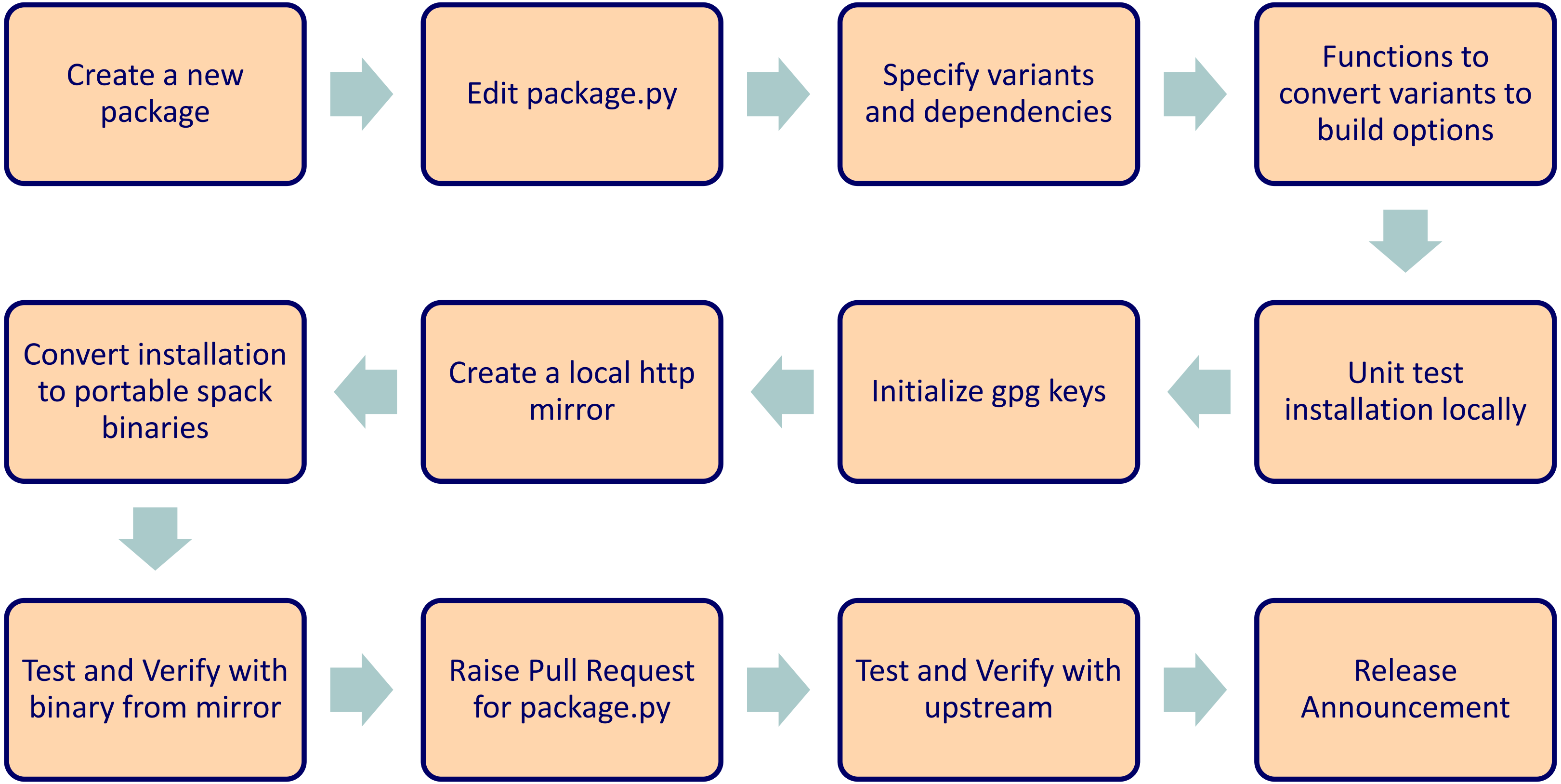
Combined Tarballs

	x86-64	OpenPOWER	ARM
Stock OFED	[GNU 4.8.5]	Coming Soon!	Coming Soon!
MLNX-OFED 3.X*	[GNU 4.8.5]	Coming Soon!	Coming Soon!
MLNX-OFED 4.X*	[GNU 4.8.5]	Coming Soon!	Coming Soon!
Intel IFS 10.6	[GNU 4.8.5]	N/A	N/A
Intel IFS 10.9	[GNU 4.8.5]	N/A	N/A

Presentation Overview

- MVAPICH Project
  - MPI and PGAS Library with CUDA-Awareness for HPC and DL
  - MVAPICH on Cloud
- **Deployment Solutions**
  - Public Cloud Deployment
  - RPM/Debian-based Deployment
  - **Spack-based Deployment**
    - **Maintainer View**
    - User View
- Conclusions

Workflow


Create a new package
Edit package.py
Specify variants and dependencies
Functions to convert variants to build options
Unit test installation locally
Initialize gpg keys
Create a local http mirror
Convert installation to portable spack binaries
Test and Verify with binary from mirror
Raise Pull Request for package.py
Test and Verify with upstream
Release Announcement

Package.py walkthrough – MVAPICH2-X & MVAPICH2-GDR

- Version and checksum of source tar ball with which the binaries were built

```
version('2.3', sha256='fc47070e2e9fac09b97022be2320200d732a0a4a820a2b51532b88f8ded14536', preferred=True)
version('2.3rc3', sha256='85a9f1ea1a837d487e356f021ef6f3a4661ad270a0c5f54777b362ee4d45166f')

provides('mpi')
provides('mpi@:3.1')
```

- Variants – MPI features, process_manager, OFED distribution, pmi_version
  - More details of features in download page
  - http://mvapich.cse.ohio-state.edu/downloads/

```
variant(
    'feature',
    description=('Feature descriptions are specified at: '
                 'https://mvapich.cse.ohio-state.edu/downloads/'),
    default='basic',
    values=('basic', 'basic-xpmem', 'advanced', 'advanced-xpmem'),
    multi=False
)
```

Package.py walkthrough - Dependencies

- MVAPICH2-X

```
depends_on('bison@3.4.2', type='build')
depends_on('libpciaccess@0.13.5', when=(sys.platform ≠ 'darwin'))
depends_on('libxml2@2.9.10')
depends_on('pmix@3.1.3', when='pmi_version=pmix')
```

- MVAPICH2-GDR

```
depends_on('bison@3.4.2', type='build')
depends_on('libpciaccess@0.13.5', when=(sys.platform ≠ 'darwin'))
depends_on('libxml2@2.9.10')
depends_on('cuda@9.2.88:10.2.89')
depends_on('pmix@3.1.3', when='pmi_version=pmix')
```

Package.py walkthrough – Build options

- Example of a function to convert the feature variant to build options

```
@property
def process_feature_options(self):
    spec = self.spec
    opts = []

    if 'feature=basic' in spec:
        opts = ['--enable-mcast', '--enable-hybrid', '--enable-mpit-tool',
                '--enable-mpit-pvars=mv2']
    elif 'feature=basic-xpmem' in spec:
        opts = ['--enable-mcast', '--enable-hybrid', '--enable-mpit-tool',
                '--enable-mpit-pvars=mv2', '--with-xpmem=/opt/xpmem/']
    elif 'feature=advanced' in spec:
        opts = ['--enable-mcast', '--enable-hybrid', '--enable-mpit-tool',
                '--enable-mpit-pvars=mv2', '--with-core-direct',
                '--enable-dc', '--enable-umr']
    elif 'feature=advanced-xpmem' in spec:
        opts = ['--enable-mcast', '--enable-hybrid', '--enable-mpit-tool',
                '--enable-mpit-pvars=mv2', '--with-core-direct',
                '--enable-dc', '--enable-umr', '--with-xpmem=/opt/xpmem/']
    return opts
```

- Issues and Wish List
- Issue #1
  - Spack will fetch latest version of dependencies and generate hash
  - Results in “no binary available” if version of dependency has changed
  - Would be good if Spack installs the version of dependency the binary was built with instead of the latest
- Follow up of Issue #1
  - User has no way to find out what version of dependencies the binary was created with
    - https://github.com/spack/spack/issues/17998

Package.py walkthrough – configure_args

- The start point of configuration

```
def configure_args(self):
    args = [
        '--enable-ucr',
        '--disable-static',
        '--enable-shared',
        '--disable-rdma-cm',
        '--without-hydra-ckpointlib'
    ]
    args.extend(self.process_manager_options)
    args.extend(self.distribution_options)
    args.append(self.construct_cflags)
    args.append(self.construct_ldflags)
    return args
```

Creating the Binaries

- Prerequisites
  - Mirror is setup
  - GPG keys are initialized
  - Package.py is complete
  - Packages were installed from source and are listed in $ spack find

```
$ spack buildcache create -f -m mymirror -k mv2_gpg --only package /jz6ofy
```

-f → force create (overwrite existing)

-m → specify the mirror to which the binary is installed

-k → the gpg key to use for signing the packages

--only package → only create binary for the package and not its dependencies

/jz6ofy → the hash of the installed package when you list using $ spack find -l

Raising the Pull request – github.com/spack

New packages Mvapich2x and Mvapich2-GDR #17883

Merged adamjstewart merged 8 commits into `spack:develop` from `unknown repository` on Aug 16

Conversation 35 | Commits 8 | Checks 12 | Files changed 2

Presentation Overview

- MVAPICH Project
  - MPI and PGAS Library with CUDA-Awareness for HPC and DL
  - MVAPICH on Cloud
- **Deployment Solutions**
  - Public Cloud Deployment
  - RPM/Debian-based Deployment
  - **Spack-based Deployment**
    - Maintainer View
    - **User View**
- Conclusions

Installation and Setup MVAPICH2 from Spack

Install Spack

$ git clone https://github.com/spack/spack.git

$ source ~/spack/share/spack/setup-env.sh

Installing MVAPICH2 (From Source)

$ spack info mvapich2

$ spack install mvapich2@2.3.4 %gcc@8.3.0

$ spack find -l -v -p mvapich2

Installation – MVAPICH-X or MVAPICH2-GDR

Currently only for gcc@4.8.5

```
$ spack compiler find
```

Add the required mirrors

$ spack mirror add mvapich2x http://mvapich.cse.ohio-state.edu/download/mvapich/spack-mirror/mvapich2x

$ spack mirror add mvapich2-gdr http://mvapich.cse.ohio-state.edu/download/mvapich/spack-mirror/mvapich2-gdr

Trust the public key used to sign the packages

$ wget http://mvapich.cse.ohio-state.edu/download/mvapich/spack-mirror/mvapich2x/build_cache/public.key

$ spack gpg trust public.key

Installation – MVAPICH-X or MVAPICH2-GDR from Spack

List the available binaries in the mirror

$ spack buildcache list -L -v -a

Install MVAPICH2-X and MVAPICH2-GDR

$ spack install mvapich2x@2.3%gcc@4.8.5 distribution=mofed4.6 feature=advanced-xpmem pmi_version=pmi1 process_managers=mpirun target=x86_64

$ spack install mvapich2-gdr@2.3.3~core_direct+mcast~openacc distribution=mofed4.5 pmi_version=pmi1 process_managers=mpirun ^cuda@9.2.88 target=x86_64

Supported CUDA Versions

– ^cuda@9.2.88, ^cuda@10.1.243, ^cuda@10.2.89

Run OSU Micro-Benchmarks

- Load the Spack binaries post installation, alternatively, export LD_LIBRARY_PATH

```
$ spack find -l -v -p mvapich2-gdr
-- linux-rhel7-ppc64le / gcc@4.8.5 ------------------------------
mkquayp mvapich2-gdr@2.3.3~core_direct+mcast~openacc cuda_version=10.2 distribution=mofed4.5 pmi_version=pmi1 process_m
anagers=mpirun  /home/user/spack/opt/spack/linux-centos7-x86_64/gcc-4.8.5/mvapich2-gdr-2.3.3-mkquaypgqutewy4yjrsx4bz7zph
wz7bb

# Note the hash of the required version - It's the first word of the previous command's output
$ spack load /mkquayp
$ which mpirun_rsh
```

- Run OSU Micro-Benchmarks

$ ./bin/mpirun_rsh -np 2 -hostfile ~/hostfile ./libexec/osu-micro-benchmarks/mpi/pt2pt/osu_latency

Useful Links

- Full Setup Instructions of MVAPICH2, MVAPICH2-X and MVAPICH2-GDR with Spack

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

- More information about Spack

  https://spack.io/

- Binary Packaging Reference

  https://archive.fosdem.org/2018/schedule/event/llnl_spack/attachments/slides/2663/export/events/attachments/llnl_spack/slides/2663/fosdem_spack_binary_packaging.pdf

Presentation Overview

- MVAPICH Project
  - MPI and PGAS Library with CUDA-Awareness for HPC and DL
  - MVAPICH on Cloud
- Deployment Solutions
  - Public Cloud Deployment
  - RPM/Debian-based Deployment
  - Spack-based Deployment
- **Conclusions**

Concluding Remarks

- Upcoming Exascale systems need to be designed with a holistic view of HPC, Deep Learning, and Cloud
- Presented an overview of designing convergent software stacks
- Presented solutions enable HPC and Deep Learning communities to take advantage of current and next-generation systems
- Presented solutions to deploy these solutions on traditional HPC and Cloud systems

8th Annual MVAPICH User Group (MUG) Meeting

- **Held August 24-26, 2020; Columbus, Ohio, USA**
- Keynote Talks, Invited Talks, Invited Tutorials by ARM, Mellanox, Contributed Presentations, Student Poster Presentations, Tutorial on MVAPICH2, MVAPICH2-X, MVAPICH2-GDR, OSU INAM as well as other optimization and tuning hints.

- **Keynote Speakers**
  - Brian van Essen, Lawrence Livermore National Laboratory (LLNL)
  - Michael L. Norman (San Diego Supercomputing Center)
- **Tutorials**
  - ARM
    - A tutorial from ARM on Performance Engineering using ARM's Scalable Vector Instructions (SVEs).
  - Mellanox
    - A tutorial from Mellanox focusing on the upcoming technologies being made available in next-generation Mellanox adapters and switches for Exascale systems.
  - Microsoft
    - A tutorial from Microsoft on various Microsoft Azure HPC offerings, best-practices and discussion on performance and scalability of using MVAPICH2 on real-world HPC applications.
  - OSU
    - Tutorials from OSU on advanced features of various MVAPICH2 libraries and InfiniBand Network Analysis and Monitoring (INAM)

- **Invited Speakers**
  - Devendar Bureddy, NVIDIA Mellanox Network Business Unit
  - John Cazes, Texas Advanced Computing Center
  - Donglai Dai, X-ScaleSolutions
  - Hyun-Wook Jin, Konkuk University, South Korea
  - Jithin Jose, Microsoft Azure
  - Minsik Kim, KISTI, South Korea
  - Pramod Kumbhar, Blue Brain Project, EPFL, Switzerland
  - John Linford, ARM
  - Heechang Na, Ohio Supercomputing Center
  - Raghunath Rajachandrasekar, AWS
  - Hemal Sah, Broadcom
  - Karen Schramm, Broadcom
  - Gilad Shainer, NVIDIA Mellanox Network Business Unit
  - Devesh Sharma, Broadcom
  - Sameer Shende, ParaTools and University of Oregon
  - Sayantan Sur, Intel
  - Alan Sussman, National Science Foundation (NSF)
  - Mahidhar Tatineni, San Diego Supercomputing Center (SDSC)
  - Karen Tomko, Ohio Supercomputing Center
  - Moshe Voloshin, Broadcom

More details and videos of the event available at: http://mug.mvapich.cse.ohio-state.edu

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 Supercomputing Centers

X-ScaleSolutions

Funding Acknowledgments

Funding Support by

Equipment Support by

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

Current Students (Graduate)

- Q. Anthony (Ph.D.)
- M. Bayatpour (Ph.D.)
- C.-H. Chu (Ph.D.)
- A. Jain (Ph.D.)
- M. Kedia (M.S.)
- K. S. Khorassani (Ph.D.)
- P. Kousha (Ph.D.)
- N. S. Kumar (M.S.)
- B. Ramesh (Ph.D.)
- K. K. Suresh (Ph.D.)
- N. Sarkauskas (Ph.D.)
- S. Srivastava (M.S.)
- S. Xu (Ph.D.)
- Q. Zhou (Ph.D.)

Current Research Scientists

- A. Shafi
- H. Subramoni

Current Senior Research Associate

- J. Hashmi

Current Software Engineer

- A. Reifsteck

Current Post-docs

- M. S. Ghazimeersaeed
- K. Manian

Current Research Specialist

- J. Smith

Past Students

- A. Awan (Ph.D.)
- 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.)
- J. Hashmi (Ph.D.)
- 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.)
- P. Lai (M.S.)
- J. Liu (Ph.D.)
- M. Luo (Ph.D.)
- A. Mamidala (Ph.D.)
- G. Marsh (M.S.)
- V. Meshram (M.S.)
- A. Moody (M.S.)
- S. Naravula (Ph.D.)
- R. Noronha (Ph.D.)
- X. Ouyang (Ph.D.)
- S. Pai (M.S.)
- S. Potluri (Ph.D.)
- K. Raj (M.S.)
- R. Rajachandrasekar (Ph.D.)
- D. Shankar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- N. Sarkauskas (B.S.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Sur (Ph.D.)
- H. Subramoni (Ph.D.)
- K. Vaidyanathan (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)
- J. Zhang (Ph.D.)

Past Post-Docs

- D. Banerjee
- X. Besseron
- H.-W. Jin
- J. Lin
- M. Luo
- E. Mancini
- S. Marcarelli
- A. Ruhela
- J. Vienne
- H. Wang

Past Research Scientists

- K. Hamidouche
- S. Sur
- X. Lu

Past Programmers

- D. Bureddy
- J. Perkins

Past Research Specialist

- M. Arnold

Thank You!

subramon@cse.ohio-state.edu

Network Based Computing Laboratory

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

MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

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

HiBD
High-Performance Big Data

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

HiDL
High-Performance Deep Learning

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