The MVAPICH2 Project: Latest Developments and Plans Towards Exascale Computing

Presentation at OSU Booth  (SC ‘18)

by

Hari Subramoni
The Ohio State University
E-mail: subramon@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~subramon
Drivers of Modern HPC Cluster Architectures

- Multi-core/many-core technologies
- Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
- Solid State Drives (SSDs), Non-Volatile Random-Access Memory (NVRAM), NVMe-SSD
- Accelerators (NVIDIA GPGPUs and Intel Xeon Phi)
- Available on HPC Clouds, e.g., Amazon EC2, NSF Chameleon, Microsoft Azure, etc.

**Multi-core Processors**
- <1 usec latency, 100Gbps Bandwidth>

**High Performance Interconnects - InfiniBand**
- <1 usec latency, 100Gbps Bandwidth>

**Accelerators / Coprocessors**
- high compute density, high performance/watt
- >1 TFlop DP on a chip

**SSD, NVMe-SSD, NVRAM**

*Summit*  
*Sunway TaihuLight*  
*Sierra*  
*K - Computer*
Parallel Programming Models Overview

- Programming models provide abstract machine models
- Models can be mapped on different types of systems
  - e.g. Distributed Shared Memory (DSM), MPI within a node, etc.
- PGAS models and Hybrid MPI+PGAS models are gradually receiving importance

Shared Memory Model
SHMEM, DSM

Distributed Memory Model
MPI (Message Passing Interface)

Partitioned Global Address Space (PGAS)
Global Arrays, UPC, Chapel, X10, CAF, ...
Designing Communication Libraries for Multi-Petaflop and Exaflop Systems: Challenges

Application Kernels/Applications

Middleware

Programming Models
MPI, PGAS (UPC, Global Arrays, OpenSHMEM), CUDA, OpenMP, OpenACC, Cilk, Hadoop (MapReduce), Spark (RDD, DAG), 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, 40/100GigE, Aries, and OmniPath)

Multi/Many-core Architectures

Accelerators (GPU and FPGA)

Co-Design Opportunities and Challenges across Various Layers
Performance
Scalability
Fault-Resilience
MPI+X Programming model: Broad Challenges 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
• 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 GPGPUs and FPGAs
• Fault-tolerance/resiliency
• QoS support for communication and I/O
• Support for Hybrid MPI+PGAS programming (MPI + OpenMP, MPI + UPC, MPI+UPC++, MPI + OpenSHMEM, CAF, ...)
• Virtualization
• Energy-Awareness
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 2,950 organizations in 86 countries
  – More than 505,000 (> 0.5 million) downloads from the OSU site directly
  – Empowering many TOP500 clusters (Jul ‘18 ranking)
    • 2nd ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    • 12th, 556,104 cores (Oakforest-PACS) in Japan
    • 15th, 367,024 cores (Stampede2) at TACC
    • 24th, 241,108-core (Pleiades) at NASA and many others
  – Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  – http://mvapich.cse.ohio-state.edu Partner in the upcoming TACC Frontera System
  – Empowering Top500 systems for over a decade
MVAPICH2 Release Timeline and Downloads

Number of Downloads

Timeline

Architecture of MVAPICH2 Software Family

High Performance Parallel Programming Models

- Message Passing Interface (MPI)
- PGAS (UPC, OpenSHMEM, CAF, UPC++)
- Hybrid --- MPI + X (MPI + PGAS + OpenMP/Cilk)

High Performance and Scalable Communication Runtime

Diverse APIs and Mechanisms

- Point-to-point Primitives
- Collectives Algorithms
- Job Startup
- Energy-Awareness
- Remote Memory Access
- I/O and File Systems
- Fault Tolerance
- Virtualization
- Active Messages
- Introspection & Analysis

Support for Modern Networking Technology (InfiniBand, iWARP, RoCE, Omni-Path)

- 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, ARM, NVIDIA GPGPU)

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

* Upcoming
## MVAPICH2 Software Family

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

- **MVAPICH2**
  - Basic MPI support for IB, iWARP and RoCE
- **MVAPICH2-X**
  - Advanced MPI features and support for INAM
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
- **MVAPICH2-Virt**
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- **MVAPICH2-EA**
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- **OSU Micro-Benchmarks (OMB)**
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- **OSU INAM**
  - InfiniBand Network Analysis and Monitoring Tool
- **MVAPICH2-GDR and Deep Learning** (Will be presented on Thursday at 10:30am)
MVAPICH2 2.3-GA

- Released on 07/23/2018
- Major Features and Enhancements
  - Based on MPICH v3.2.1
  - Introduce basic support for executing MPI jobs in Singularity
  - Improve performance for MPI-3 RMA operations
  - Enhancements for Job Startup
    - Improved job startup time for OFA-IB-CH3, PSM-CH3, and PSM2-CH3
    - On-demand connection management for PSM-CH3 and PSM2-CH3 channels
    - Enhance PSM-CH3 and PSM2-CH3 job startup to use non-blocking PMI calls
    - Introduce capability to run MPI jobs across multiple InfiniBand subnets
  - Enhancements to point-to-point operations
    - Enhance performance of point-to-point operations for CH3-Gen2 (InfiniBand), CH3-PSM, and CH3-PSM2 (Omni-Path) channels
    - Improve performance for Intra- and Inter-node communication for OpenPOWER architecture
    - Enhanced tuning for OpenPOWER, Intel Skylake and Cavium ARM (ThunderX) systems
    - Improve performance for host-based transfers when CUDA is enabled
    - Improve support for large processes per node and hugepages on SMP systems
  - Enhancements to collective operations
    - Enhanced performance for Allreduce, Reduce_scatter_block, Allgatherv
      - Thanks to Danielle Sikich and Adam Moody @ LLNL for the patch
    - Add support for non-blocking Allreduce using Mellanox SHARP
      - Enhance tuning framework for Allreduce using SHArP
    - Enhanced collective tuning for IBM POWER8, IBM POWER9, Intel Skylake, Intel KNL, Intel Broadwell
  - Enhancements to process mapping strategies and automatic architecture/network detection
    - Improve performance of architecture detection on high core-count systems
    - Enhanced architecture detection for OpenPOWER, Intel Skylake and Cavium ARM (ThunderX) systems
    - New environment variable MV2_THREADS_BINDING_POLICY for multi-threaded MPI and MPI+OpenMP applications
    - Support 'spread', 'bunch', 'scatter', 'linear' and 'compact' placement of threads
      - Warn user if oversubscription of core is detected
    - Enhance MV2_SHOW_CPU_BINDING to enable display of CPU bindings on all nodes
    - Added support for MV2_SHOW_CPU_BINDING to display number of OMP threads
    - Added logic to detect heterogeneous CPU/HFI configurations in PSM-CH3 and PSM2-CH3 channels
      - Thanks to Matias Cabral@Intel for the report
    - Enhanced HFI selection logic for systems with multiple Omni-Path HFIs
    - Introduce run time parameter MV2_SHOW_HCA_BINDING to show process to HCA bindings
  - Miscellaneous enhancements and improved debugging and tools support
    - Enhance support for MPI_T_PVARs and CVARs
    - Enhance debugging support for PSM-CH3 and PSM2-CH3 channels
    - Update to hwloc version 1.11.9
    - Tested with CLANG v5.0.0
One-way Latency: MPI over IB with MVAPICH2

Small Message Latency

Latency (us)

Message Size (bytes)

0 4 8 16 32 64 128 256 512 1k

1.19
1.11
1.15
1.01
1.04

TrueScale-QDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
ConnectX-4-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch
Omni-Path - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with Omni-Path switch

Large Message Latency

Latency (us)

Message Size (bytes)

2k 4k 8k 16k 32k 64k 128k 256k

TrueScale-QDR
ConnectX-3-FDR
ConnectIB-DualFDR
ConnectX-4-EDR
Omni-Path
Bandwidth: MPI over IB with MVAPICH2

Unidirectional Bandwidth

Bidirectional Bandwidth

TrueScale-QDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
ConnectIB-DualFDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
ConnectX-4-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 IB switch
Omni-Path - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with Omni-Path switch
• MPI_Init takes 22 seconds on 231,936 processes on 3,624 KNL nodes (Stampede2 – Full scale)
• At 64K processes, MPI_Init and Hello World takes 5.8s and 21s respectively (Oakforest-PACS)
• All numbers reported with 64 processes per node, MVAPICH2-2.3a
• Designs integrated with mpirun_rsh, available for srun (SLURM launcher) as well
Benefits of SHARP Allreduce at Application Level

![Graph showing latency comparison between MVAPICH2 and other configurations.]

**SHARP support available since MVAPICH2 2.3a**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV2_ENABLE_SHARP=1</td>
<td>Enables SHARP-based collectives</td>
<td>Disabled</td>
</tr>
<tr>
<td>--enable-sharp</td>
<td>Configure flag to enable SHARP</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

- Refer to [Running Collectives with Hardware based SHARP support](http://mvapich.cse.ohio-state.edu/static/media/mvapich/mvapich2-2.3-userguide.html#x1-990006.26) section of MVAPICH2 user guide for more information
Intra-node Point-to-Point Performance on OpenPower

**Intra-Socket Small Message Latency**
- MVAPICH2-2.3
- SpectrumMPI-10.1.0.2
- OpenMPI-3.0.0

**Intra-Socket Large Message Latency**
- MVAPICH2-2.3
- SpectrumMPI-10.1.0.2
- OpenMPI-3.0.0

**Intra-Socket Bandwidth**
- MVAPICH2-2.3
- SpectrumMPI-10.1.0.2
- OpenMPI-3.0.0

**Intra-Socket Bi-directional Bandwidth**
- MVAPICH2-2.3
- SpectrumMPI-10.1.0.2
- OpenMPI-3.0.0

*Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA*
Intra-node Point-to-point Performance on ARM Cortex-A72

**Small Message Latency**

- **Latency (us)**
  - MVAPICH2-2.3
  - 0.27 micro-second (1 bytes)

**Large Message Latency**

- **Latency (us)**
  - MVAPICH2-2.3

**Bandwidth**

- **Bandwidth (MB/s)**
  - MVAPICH2-2.3

**Bi-directional Bandwidth**

- **Bidirectional Bandwidth**
  - MVAPICH2-2.3

Platform: ARM Cortex A72 (aarch64) MIPS processor with 64 cores dual-socket CPU. Each socket contains 32 cores.
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
Dynamic and Adaptive Tag Matching

Challenge
Tag matching is a significant overhead for receivers
Existing Solutions are
- Static and do not adapt dynamically to communication pattern
- Do not consider memory overhead

Solution
A new tag matching design
- Dynamically adapt to communication patterns
- Use different strategies for different ranks
- Decisions are based on the number of request object that must be traversed before hitting on the required one

Results
Better performance than other state-of-the-art tag-matching schemes
Minimum memory consumption
Will be available in future MVAPICH2 releases

Dynamic and Adaptive MPI Point-to-point Communication Protocols

**Desired Eager Threshold**

<table>
<thead>
<tr>
<th>Process Pair</th>
<th>Eager Threshold (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 – 4</td>
<td>32</td>
</tr>
<tr>
<td>1 – 5</td>
<td>64</td>
</tr>
<tr>
<td>2 – 6</td>
<td>128</td>
</tr>
<tr>
<td>3 – 7</td>
<td>32</td>
</tr>
</tbody>
</table>

**Eager Threshold for Example Communication Pattern with Different Designs**

- **Default**: 0 – 4 (32 KB), 1 – 5 (64 KB), 2 – 6 (128 KB), 3 – 7 (32 KB)
- **Manually Tuned**: 0 – 4 (128 KB), 1 – 5 (128 KB), 2 – 6 (128 KB), 3 – 7 (32 KB)
- **Dynamic + Adaptive**: 0 – 4 (32 KB), 1 – 5 (64 KB), 2 – 6 (128 KB), 3 – 7 (32 KB)

**Execution Time of Amber**

- Default: Poor overlap; Low memory requirement
- Manually Tuned: Good overlap; High memory requirement
- Dynamic + Adaptive: Good overlap; Optimal memory requirement

**Relative Memory Consumption of Amber**

- Default: Low Performance; High Productivity
- Manually Tuned: High Performance; Low Productivity
- Dynamic + Adaptive: High Performance; High Productivity

Application Scalability on Skylake and KNL (Stampeede2)

MiniFE (1300x1300x1300 ~ 910 GB)

NEURON (YuEtAl2012)

Cloverleaf (bm64) MPI+OpenMP, NUM_OMP_THREADS = 2

<table>
<thead>
<tr>
<th>No. of Processes (KNL: 64ppn)</th>
<th>Execution Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2048</td>
<td>4</td>
</tr>
<tr>
<td>4096</td>
<td>2</td>
</tr>
<tr>
<td>8192</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of Processes (Skylake: 48ppn)</th>
<th>Execution Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>500</td>
</tr>
<tr>
<td>96</td>
<td>1000</td>
</tr>
<tr>
<td>192</td>
<td>1500</td>
</tr>
<tr>
<td>384</td>
<td>2000</td>
</tr>
<tr>
<td>768</td>
<td>2500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of Processes (KNL: 68ppn)</th>
<th>Execution Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
<td>500</td>
</tr>
<tr>
<td>136</td>
<td>1000</td>
</tr>
<tr>
<td>272</td>
<td>1500</td>
</tr>
<tr>
<td>544</td>
<td>2000</td>
</tr>
<tr>
<td>1088</td>
<td>2500</td>
</tr>
<tr>
<td>2176</td>
<td>3000</td>
</tr>
<tr>
<td>4352</td>
<td>3500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. of Processes (Skylake: 48ppn)</th>
<th>Execution Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>500</td>
</tr>
<tr>
<td>96</td>
<td>1000</td>
</tr>
<tr>
<td>192</td>
<td>1500</td>
</tr>
<tr>
<td>384</td>
<td>2000</td>
</tr>
<tr>
<td>768</td>
<td>2500</td>
</tr>
<tr>
<td>1536</td>
<td>3000</td>
</tr>
<tr>
<td>3072</td>
<td>3500</td>
</tr>
</tbody>
</table>

Runtime parameters: MV2_SMPI_LENGTH_QUEUE=524288 PSM2_MQ_RNDV_SHM_THRESH=128K PSM2_MQ_RNDV_HFI_THRESH=128K

Courtesy: Mahidhar Tatineni @SDSC, Dong Ju (DJ) Choi@SDSC, and Samuel Khuvis@OSC ---- Testbed: TACC Stampede2 using MVAPICH2-2.3b
MVAPICH2 Distributions

- MVAPICH2
  - Basic MPI support for IB, iWARP and RoCE

- MVAPICH2-X
  - Advanced MPI features and support for INAM
  - MPI, PGAS and Hybrid MPI+PGAS support for IB

- MVAPICH2-Virt
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity

- MVAPICH2-EA
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)

- OSU Micro-Benchmarks (OMB)
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC

- OSU INAM
  - InfiniBand Network Analysis and Monitoring Tool

- MVAPICH2-GDR and Deep Learning (Will be presented on Thursday at 10:30am)
### MVAPICH2-X for Hybrid MPI + PGAS Applications

#### High Performance Parallel Programming Models

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MPI</strong></td>
<td>Message Passing Interface</td>
<td></td>
</tr>
<tr>
<td><strong>PGAS</strong></td>
<td>(UPC, OpenSHMEM, CAF, UPC++)</td>
<td></td>
</tr>
<tr>
<td><strong>Hybrid --- MPI + X</strong></td>
<td>(MPI + PGAS + OpenMP/Cilk)</td>
<td></td>
</tr>
</tbody>
</table>

#### High Performance and Scalable Unified Communication Runtime

<table>
<thead>
<tr>
<th><strong>Diverse APIs and Mechanisms</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimized Point-to-point Primitives</td>
</tr>
<tr>
<td>Remote Memory Access</td>
</tr>
<tr>
<td>Active Messages</td>
</tr>
<tr>
<td>Collectives Algorithms (Blocking and Non-Blocking)</td>
</tr>
<tr>
<td>Scalable Job Startup</td>
</tr>
<tr>
<td>Fault Tolerance</td>
</tr>
<tr>
<td>Introspection &amp; Analysis with OSU INAM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Support</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for Efficient Intra-node Communication (POSIX SHMEM, CMA, LiMIC, XPMEM...)</td>
</tr>
<tr>
<td>Support for Modern Networking Technologies (InfiniBand, iWARP, RoCE, Omni-Path...)</td>
</tr>
<tr>
<td>Support for Modern Multi-/Many-core Architectures (Intel-Xeon, GPGPUS, OpenPOWER, ARM...)</td>
</tr>
</tbody>
</table>

- **Current Model** – Separate Runtimes for OpenSHMEM/UPC/UPC++/CAF and MPI
  - Possible deadlock if both runtimes are not progressed
  - Consumes more network resource

- **Unified communication runtime for MPI, UPC, UPC++, OpenSHMEM, CAF**
  - Available with since 2012 (starting with MVAPICH2-X 1.9)
  - [http://mvapich.cse.ohio-state.edu](http://mvapich.cse.ohio-state.edu)
Minimizing Memory Footprint by Direct Connect (DC) Transport

- Constant connection cost (*One QP for any peer*)
- Full Feature Set (RDMA, Atomics etc)
- Separate objects for send (DC Initiator) and receive (DC Target)
  - DC Target identified by “DCT Number”
  - Messages routed with (DCT Number, LID)
  - Requires same “DC Key” to enable communication
- Available since MVAPICH2-X 2.2a
User-mode Memory Registration (UMR)

- Introduced by Mellanox to support direct local and remote noncontiguous memory access
- Avoid packing at sender and unpacking at receiver
- Available since MVAPICH2-X 2.2b

### Small & Medium Message Latency

![Small & Medium Message Latency](image)

### Large Message Latency

![Large Message Latency](image)

Connect-IB (54 Gbps): 2.8 GHz Dual Ten-core (IvyBridge) Intel PCI Gen3 with Mellanox IB FDR switch

On-Demand Paging (ODP)

- Applications no longer need to pin down underlying physical pages
- Memory Region (MR) are **NEVER** pinned by the OS
  - Paged in by the HCA when needed
  - Paged out by the OS when reclaimed
- ODP can be divided into two classes
  - Explicit ODP
    - Applications still register memory buffers for communication, but this operation is used to define access control for IO rather than pin-down the pages
  - Implicit ODP
    - Applications are provided with a special memory key that represents their complete address space, does not need to register any virtual address range
- Advantages
  - Simplifies programming
  - Unlimited MR sizes
  - Physical memory optimization


Available since MVAPICH2-X 2.3b
For MPI_Allreduce latency with 32K bytes, MVAPICH2-OPT can reduce the latency by **2.4X**.


Available since MVAPICH2-X 2.3b
Optimized CMA-based Collectives for Large Messages

Performance of MPI_Gather on KNL nodes (64PPN)

- Significant improvement over existing implementation for Scatter/Gather with 1MB messages (up to 4x on KNL, 2x on Broadwell, 14x on OpenPower)
- New two-level algorithms for better scalability
- Improved performance for other collectives (Bcast, Allgather, and Alltoall)


Available since MVAPICH2-X 2.3b
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

Available in MVAPICH2-X 2.3rc1


[Graph showing latency vs. message size for different systems and improvements marked]
Benefits of the New Asynchronous Progress Design: Broadwell + InfiniBand

Up to 44% performance improvement in P3DFFT application with 448 processes
Up to 19% and 9% performance improvement in HPL application with 448 and 896 processes

A. Ruhela, H. Subramoni, S. Chakraborty, M. Bayatpour, P. Kousha, and D.K. Panda, Efficient Asynchronous Communication Progress for MPI without Dedicated Resources, EuroMPI 2018

Available in MVAPICH2-X 2.3rc1
SPEC MPI 2007 Benchmarks: Broadwell + InfiniBand

- MVAPICH2-X outperforms Intel MPI by up to 31%

Configuration: 448 processes on 16 Intel E5-2680v4 (Broadwell) nodes having 28 PPN and interconnected with 100Gbps Mellanox MT4115 EDR ConnectX-4 HCA
Hybrid (MPI+PGAS) Programming

• Application sub-kernels can be re-written in MPI/PGAS based on communication characteristics

• Benefits:
  – Best of Distributed Computing Model
  – Best of Shared Memory Computing Model
Application Level Performance with Graph500 and Sort

Performance of Hybrid (MPI+OpenSHMEM) Graph500 Design

- 8,192 processes
  - 2.4X improvement over MPI-CSR
  - 7.6X improvement over MPI-Simple
- 16,384 processes
  - 1.5X improvement over MPI-CSR
  - 13X improvement over MPI-Simple

Performance of Hybrid (MPI+OpenSHMEM) Sort Application

- 4,096 processes, 4 TB Input Size
  - MPI – 2408 sec; 0.16 TB/min
  - Hybrid – 1172 sec; 0.36 TB/min
  - 51% improvement over MPI-design


J. Jose, S. Potluri, K. Tomko and D. K. Panda, Designing Scalable Graph500 Benchmark with Hybrid MPI+OpenSHMEM Programming Models, International Supercomputing Conference (ISC'13), June 2013

J. Jose, K. Kandalla, M. Luo and D. K. Panda, Supporting Hybrid MPI and OpenSHMEM over InfiniBand: Design and Performance Evaluation, Int'l Conference on Parallel Processing (ICPP '12), September 2012
MVAPICH2 Distributions

- MVAPICH2
  - Basic MPI support for IB, iWARP and RoCE
- MVAPICH2-X
  - Advanced MPI features and support for INAM
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
- MVAPICH2-Virt
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- MVAPICH2-EA
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- OSU Micro-Benchmarks (OMB)
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
  - InfiniBand Network Analysis and Monitoring Tool
- MVAPICH2-GDR and Deep Learning (Will be presented on Thursday at 10:30am)
Can HPC and Virtualization be Combined?

• Virtualization has many benefits
  – Fault-tolerance
  – Job migration
  – Compaction

• Have not been very popular in HPC due to overhead associated with Virtualization

• New SR-IOV (Single Root – IO Virtualization) support available with Mellanox InfiniBand adapters changes the field

• Enhanced MVAPICH2 support for SR-IOV

• MVAPICH2-Virt 2.2 supports:
  – OpenStack, Docker, and singularity

J. Zhang, X. Lu, J. Jose, R. Shi and D. K. Panda, Can Inter-VM Shmem Benefit MPI Applications on SR-IOV based Virtualized InfiniBand Clusters? EuroPar’14
J. Zhang, X. Lu, J. Jose, M. Li, R. Shi and D.K. Panda, High Performance MPI Libray over SR-IOV enabled InfiniBand Clusters, HiPC’14
Application-Level Performance on Chameleon

- 32 VMs, 6 Core/VM
- Compared to Native, 2-5% overhead for Graph500 with 128 Procs
- Compared to Native, 1-9.5% overhead for SPEC MPI2007 with 128 Procs

**Graph500**

- MV2-SR-IOV-Def
- MV2-SR-IOV-Opt
- MV2-Native

**SPEC MPI2007**

- MV2-SR-IOV-Def
- MV2-SR-IOV-Opt
- MV2-Native
• 64 Containers across 16 nodes, pining 4 Cores per Container
• Compared to Container-Def, up to 11% and 73% of execution time reduction for NAS and Graph 500
• Compared to Native, less than 9% and 5% overhead for NAS and Graph 500
Application-Level Performance on Singularity with MVAPICH2

- 512 Processes across 32 nodes
- Less than 7% and 6% overhead for NPB and Graph500, respectively

J. Zhang, X. Lu and D. K. Panda, Is Singularity-based Container Technology Ready for Running MPI Applications on HPC Clouds?, UCC ’17, Best Student Paper Award
MVAPICH2 Distributions

- MVAPICH2
  - Basic MPI support for IB, iWARP and RoCE
- MVAPICH2-X
  - Advanced MPI features and support for INAM
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
- MVAPICH2-Virt
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- MVAPICH2-EA
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- OSU Micro-Benchmarks (OMB)
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- OSU INAM
  - InfiniBand Network Analysis and Monitoring Tool
- MVAPICH2-GDR and Deep Learning (Will be presented on Thursday at 10:30am)
Designing Energy-Aware (EA) MPI Runtime

Overall application

Energy Expenditure

- Energy Spent in Communication Routines
- Energy Spent in Computation Routines

- Point-to-point Routines
- Collective Routines
- RMA Routines

MVAPICH2-EA Designs

- MPI Two-sided and collectives (ex: MVAPICH2)
- One-sided runtimes (ex: ComEx)
- Other PGAS Implementations (ex: OSHMPI)

Impact

MPI-3 RMA Implementations (ex: MVAPICH2)
MVAPICH2-EA: Application Oblivious Energy-Aware-MPI (EAM)

- An energy efficient runtime that provides energy savings without application knowledge
- Uses automatically and transparently the best energy lever
- Provides guarantees on maximum degradation with 5-41% savings at <= 5% degradation
- Pessimistic MPI applies energy reduction lever to each MPI call
- Released (together with OEMT) since Aug’15

A Case for Application-Oblivious Energy-Efficient MPI Runtime A. Venkatesh, A. Vishnu, K. Hamidouche, N. Tallent, D.

MVAPICH2 Distributions

- **MVAPICH2**
  - Basic MPI support for IB, iWARP and RoCE
- **MVAPICH2-X**
  - Advanced MPI features and support for INAM
  - MPI, PGAS and Hybrid MPI+PGAS support for IB
- **MVAPICH2-Virt**
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity
- **MVAPICH2-EA**
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)
- **OSU Micro-Benchmarks (OMB)**
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC
- **OSU INAM**
  - InfiniBand Network Analysis and Monitoring Tool
- **MVAPICH2-GDR and Deep Learning** (Will be presented on Thursday at 10:30am)
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++)
    • 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

• Please visit the following link for more information
  – http://mvapich.cse.ohio-state.edu/benchmarks/
MVAPICH2 Distributions

- **MVAPICH2**
  - Basic MPI support for IB, iWARP and RoCE

- **MVAPICH2-X**
  - Advanced MPI features and support for INAM
  - MPI, PGAS and Hybrid MPI+PGAS support for IB

- **MVAPICH2-Virt**
  - Optimized for HPC Clouds with IB and SR-IOV virtualization
  - Support for OpenStack, Docker, and Singularity

- **MVAPICH2-EA**
  - Energy Efficient Support for point-to-point and collective operations
  - Compatible with OSU Energy Monitoring Tool (OEMT-0.8)

- **OSU Micro-Benchmarks (OMB)**
  - MPI (including CUDA-aware MPI), OpenSHMEM and UPC

- **OSU INAM**
  - InfiniBand Network Analysis and Monitoring Tool

- **MVAPICH2-GDR and Deep Learning** (Will be presented on Thursday at 10:30am)
Overview of OSU INAM

- A network monitoring and analysis tool that is capable of analyzing traffic on the InfiniBand network with inputs from the MPI runtime
  - [http://mvapich.cse.ohio-state.edu/tools/osu-inam/](http://mvapich.cse.ohio-state.edu/tools/osu-inam/)
- Monitors IB clusters in real time by querying various subnet management entities and gathering input from the MPI runtimes
- Capability to analyze and profile node-level, job-level and process-level activities for MPI communication
  - Point-to-Point, Collectives and RMA
- Ability to filter data based on type of counters using “drop down” list
- Remotely monitor various metrics of MPI processes at user specified granularity
- "Job Page" to display jobs in ascending/descending order of various performance metrics in conjunction with MVAPICH2-X
- Visualize the data transfer happening in a “live” or “historical” fashion for entire network, job or set of nodes
- OSU INAM 0.9.4 released on 11/10/2018
  - Enhanced performance for fabric discovery using optimized OpenMP-based multi-threaded designs
  - Ability to gather InfiniBand performance counters at sub-second granularity for very large (>2000 nodes) clusters
  - Redesign database layout to reduce database size
  - Enhanced fault tolerance for database operations
    - Thanks to Trey Dockendorf @ OSC for the feedback
  - OpenMP-based multi-threaded designs to handle database purge, read, and insert operations simultaneously
  - Improved database purging time by using bulk deletes
  - Tune database timeouts to handle very long database operations
  - Improved debugging support by introducing several debugging levels
OSU INAM Features

Comet@SDSC --- Clustered View

Finding Routes Between Nodes

(1,879 nodes, 212 switches, 4,377 network links)

- Show network topology of large clusters
- Visualize traffic pattern on different links
- Quickly identify congested links/links in error state
- See the history unfold – play back historical state of the network
OSU INAM Features (Cont.)

Visualizing a Job (5 Nodes)

- **Job level view**
  - Show different network metrics (load, error, etc.) for any live job
  - Play back historical data for completed jobs to identify bottlenecks

- **Node level view - details per process or per node**
  - CPU utilization for each rank/node
  - Bytes sent/received for MPI operations (pt-to-pt, collective, RMA)
  - Network metrics (e.g. XmitDiscard, RcvError) per rank/node

Estimated Process Level Link Utilization

- **Estimated Link Utilization view**
  - Classify data flowing over a network link at different granularity in conjunction with MVAPICH2-X 2.2rc1
    - Job level and
    - Process level
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.
MVAPICH Team Part of New NSF Tier-1 System

• TACC and MVAPICH partner to win new NSF Tier-1 System

• The MVAPICH team will be an integral part of the effort

• An overview of the recently awarded NSF Tier 1 System at TACC will be presented by Dan Stanzione
  – The presentation will also include discussion on MVAPICH collaboration on past systems and this upcoming system at TACC
Commercial Support for MVAPICH2 Libraries

• Supported through X-ScaleSolutions (http://x-scalesolutions.com)
• Benefits:
  – Help and guidance with installation of the library
  – Platform-specific optimizations and tuning
  – Timely support for operational issues encountered with the library
  – Web portal interface to submit issues and tracking their progress
  – Advanced debugging techniques
  – Application-specific optimizations and tuning
  – Obtaining guidelines on best practices
  – Periodic information on major fixes and updates
  – Information on major releases
  – Help with upgrading to the latest release
  – Flexible Service Level Agreements
• Support provided to Lawrence Livermore National Laboratory (LLNL) this year
MVAPICH2 – Plans for Exascale

- Performance and Memory scalability toward 1M-10M cores
- Hybrid programming (MPI + OpenSHMEM, MPI + UPC, MPI + CAF ...)
  - MPI + Task*
- Enhanced Optimization for GPUs and FPGAs*
- Taking advantage of advanced features of Mellanox InfiniBand
  - Tag Matching*
  - Adapter Memory*
- Enhanced communication schemes for upcoming architectures
  - NVLINK*
  - 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
One More Presentation

- Thursday (11/15/18) at 10:30am

MVAPICH2-GDR: Pushing the Frontier of HPC and Deep Learning
Join us at the OSU Booth

• Join the OSU Team at SC’18 at Booth #4404 and grab a free T-Shirt!
• More details about various events available at the following link
• http://mvapich.cse.ohio-state.edu/conference/735/talks/
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/

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

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