The MVAPICH2 Project
Latest Status and Future Plans

Presentation at MPICH BoF (SC ‘19)
by

Hari Subramoni
The Ohio State University
E-mail: subramon@cse.ohio-state.edu

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

• A long time ago, in a galaxy far, far away…. (actually 20 years ago), there existed...
• MPICH
  – High performance and widely portable implementation of MPI standard
  – From ANL
• MVICH
  – Implementation of MPICH ADI-2 for VIA
  – VIA – Virtual Interface Architecture (precursor to InfiniBand)
  – From LBL
• VAPI
  – Verbs level API
  – Initial InfiniBand API from IB Vendors (older version of OFED/IB verbs)

\[ \text{MPICH} + \text{MVICH} + \text{VAPI} = \text{MVAPICH} \]
Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
  - Based on MPICH2 3.2.1
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.1), Started in 2001, First version available in 2002 (SC ‘02)
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
  - Used by more than 3,050 organizations in 89 countries
  - More than 615,000 (> 0.6 million) downloads from the OSU site directly
  - Empowering many TOP500 clusters (Nov ‘19 ranking)
    - 3rd, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China
    - 5th, 448, 448 cores (Frontera) at TACC
    - 8th, 391,680 cores (ABCI) in Japan
    - 14th, 570,020 cores (Neurion) in South Korea and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  - http://mvapich.cse.ohio-state.edu

- Empowering Top500 systems for over a decade

Partner in the #5th TACC Frontera System
MVAPICH2 Release Timeline and Downloads

Number of Downloads

Timeline

MV 0.9.4
MV2 0.9.0
MV2 0.9.8
MV 1.0
MV2 1.0.3
MV 1.1
MV2 1.4
MV 1.5
MV2 1.6
MV 1.7
MV2 1.8
MV 1.9
MV2-GDR 2.0b
MV2-MIC 2.0
MV2 Virt 2.2
MV2-Azure 2.3
MV2 2.3.2
OSU INAM 0.9.3
MV2-GDR 2.3.2
MV2-MIC 2.3
MV2-X 2.3 rc2
MV 2.3.2
MV2 Virt 2.3
MV2-AWS 2.3
MV2-Azure 2.3

Network Based Computing Laboratory

MPICH BoF (SC’19)
Architecture of MVAPICH2 Software Family (HPC and DL)

High Performance Parallel Programming Models

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

High Performance and Scalable Communication Runtime

**Diverse APIs and Mechanisms**

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

**Support for Modern Networking Technology** (InfiniBand, iWARP, RoCE, Omni-Path, Elastic Fabric Adapter)

- **Transport Protocols**
  - RC
  - SRD
  - UD
  - DC

- **Modern Features**
  - UMR
  - ODP
  - SR-IOV
  - Multi Rail

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

- **Transport Mechanisms**
  - Shared Memory
  - CMA
  - IVSHMEMP
  - XPMEM

- **Modern Features**
  - Optane*
  - NVLink
  - CAPI*

* Upcoming
## MVAPICH2 Software Family

### High-Performance Parallel Programming Libraries

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVAPICH2</td>
<td>Support for InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE</td>
</tr>
<tr>
<td>MVAPICH2-X</td>
<td>Advanced MPI features, OSU INAM, PGAS (OpenSHMEM, UPC, UPC++, and CAF), and MPI+PGAS programming models with unified communication runtime</td>
</tr>
<tr>
<td>MVAPICH2-GDR</td>
<td>Optimized MPI for clusters with NVIDIA GPUs</td>
</tr>
<tr>
<td>MVAPICH2-Virt</td>
<td>High-performance and scalable MPI for hypervisor and container based HPC cloud</td>
</tr>
<tr>
<td>MVAPICH2-EA</td>
<td>Energy aware and High-performance MPI</td>
</tr>
<tr>
<td>MVAPICH2-MIC</td>
<td>Optimized MPI for clusters with Intel KNC</td>
</tr>
</tbody>
</table>

### Microbenchmarks

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMB</td>
<td>Microbenchmarks suite to evaluate MPI and PGAS (OpenSHMEM, UPC, and UPC++) libraries for CPUs and GPUs</td>
</tr>
</tbody>
</table>

### Tools

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSU INAM</td>
<td>Network monitoring, profiling, and analysis for clusters with MPI and scheduler integration</td>
</tr>
<tr>
<td>OEMT</td>
<td>Utility to measure the energy consumption of MPI applications</td>
</tr>
</tbody>
</table>
MVAPICH2 – Basic MPI

Fast Startup on Emerging Many-Cores

**MPI_Init on Frontera**

<table>
<thead>
<tr>
<th>Number of Processes</th>
<th>Time Taken (Milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>65</td>
</tr>
<tr>
<td>224</td>
<td>100</td>
</tr>
<tr>
<td>896</td>
<td>2000</td>
</tr>
<tr>
<td>3584</td>
<td>3500</td>
</tr>
<tr>
<td>14336</td>
<td>4000</td>
</tr>
<tr>
<td>57344</td>
<td>4500</td>
</tr>
</tbody>
</table>

**Intel MPI 2019**

- Time Taken: 4.5s

**MVAPICH2 2.3.2**

- Time Taken: 3.9s

Enhanced Intra-node Performance for ARM

**MVAPICH2-2.3.2**

- Latency: 0.25us

Enhanced Intra-node Performance for OpenPOWER

**MVAPICH2 2.3.2**

- Latency: 0.25us

**SpectrumMPI-10.3.0.01**

- Latency: 4.0us

Advanced Allreduce with SHARP

<table>
<thead>
<tr>
<th>(Number of Nodes, PPN)</th>
<th>Latency (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4,28)</td>
<td>0.25</td>
</tr>
<tr>
<td>(8,28)</td>
<td>0.3</td>
</tr>
<tr>
<td>(16,28)</td>
<td>0.4</td>
</tr>
</tbody>
</table>

13% Higher is Better

Advanced Non-Blocking Allreduce with SHARP

<table>
<thead>
<tr>
<th>(Number of Nodes, PPN)</th>
<th>Communication-Computation Overlap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 PPN, 8 Nodes</td>
<td>MVAPICH2</td>
</tr>
<tr>
<td></td>
<td>MVAPICH2-SHArP</td>
</tr>
</tbody>
</table>

MPI_Bcast using RDMA_CM-based Multicast

<table>
<thead>
<tr>
<th>No. of Nodes</th>
<th>Latency (us)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>32</td>
<td>10</td>
</tr>
<tr>
<td>128</td>
<td>100</td>
</tr>
<tr>
<td>512</td>
<td>1000</td>
</tr>
</tbody>
</table>

- 16: 1000us, 256: 100us, 4K: 10us, 64K: 1us, 512K: 1us
MVAPICH2-X – Advanced MPI + PGAS + Tools

**CMA-Aware MPI_Bcast**

- **Power8 (160 Processes)**
  - Use CMA
  - Use SHMEM
  - MVAPICH2-X 2.3rc1

**Shared Address Space (XPMEM)-based Collectives Design**

- **OSU_Allreduce (Broadwell 256 procs)**
  - MVAPICH2-2.3b
  - IMPI-2017v1.132
  - 1.8X

- **OSU_Reduce (Broadwell 256 procs)**
  - MVAPICH2-2.3b
  - IMPI-2017v1.132
  - MVAPICH2-X 2.3rc1
  - 4X

**Performance of P3DFFT Optimized Async Progress**

- **Time per loop in seconds**
  - MVAPICH2-X Async
  - MVAPICH2-X Default
  - Intel MPI 18.1.163
  - 33%
  - 44%

- **Neuron with YuEtAl2012**
  - MVAPICH2
  - MVAPICH2-X
  - 10%
  - 39%
  - 76%

**Overhead of RC protocol for connection establishment and communication**
MVAPICH2-GDR – Optimized MPI for clusters with NVIDIA GPUs

Best Performance for GPU-based Transfers

TensorFlow Training with MVAPICH2-GDR on Summit

GPU-Based MPI_Allreduce on Summit

Enhanced Kernel-based Datatype Processing for Cosmo Weather Prediction Model on x86

Enhanced Kernel-based Datatype Processing for COMB Application Kernel on POWER9

16 GPUs on POWER9 system (test Comm mpi Mesh cuda Device Buffers mpi_type)

<table>
<thead>
<tr>
<th></th>
<th>pre-comm</th>
<th>post-rev</th>
<th>post-send</th>
<th>wait-rev</th>
<th>wait-send</th>
<th>post-comm</th>
<th>start-up</th>
<th>test-comm</th>
<th>bench-comm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectrum MPI 10.3</td>
<td>0.0001</td>
<td>0.0000</td>
<td>1.6021</td>
<td>1.7204</td>
<td>0.0112</td>
<td>0.0001</td>
<td>7.7383</td>
<td>83.6229</td>
<td></td>
</tr>
<tr>
<td>MVAPICH2-GDR 2.3.2</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0862</td>
<td>0.0871</td>
<td>0.0018</td>
<td>0.0001</td>
<td>0.0009</td>
<td>0.3558</td>
<td>4.4396</td>
</tr>
<tr>
<td>MVAPICH2-GDR 2.3.3 (Upcoming)</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0030</td>
<td>0.0032</td>
<td>0.0001</td>
<td>0.0001</td>
<td>0.0009</td>
<td>0.0133</td>
<td>0.1602</td>
</tr>
</tbody>
</table>
MVAPICH2-X Advanced Support for HPC-Clouds

Performance on Amazon EFA

<table>
<thead>
<tr>
<th>Processes (Nodes X PPN)</th>
<th>Execution Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>72(2x36)</td>
<td>72X</td>
</tr>
<tr>
<td>144(4x36)</td>
<td>144X</td>
</tr>
<tr>
<td>288(8x36)</td>
<td>288X</td>
</tr>
</tbody>
</table>

- **miniGhost**: 10% better
- **CloverLeaf**: 27.5% better

Instance type: c5n.18xlarge
CPU: Intel Xeon Platinum 8124M @ 3.00GHz
MVAPICH2 version: MVAPICH2-X 2.3rc2 + SRD support
OpenMPI version: Open MPI v3.1.3 with libfabric 1.7

Performance of Radix on Microsoft Azure

<table>
<thead>
<tr>
<th>Processes (Nodes X PPN)</th>
<th>Execution Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>60(1X60)</td>
<td>60X</td>
</tr>
<tr>
<td>120(2X60)</td>
<td>120X</td>
</tr>
<tr>
<td>240(4X60)</td>
<td>240X</td>
</tr>
</tbody>
</table>

- **Total Execution Time on HC (Lower is better)**: MVAPICH2-X 3x faster
- **Total Execution Time on HB (Lower is better)**: MVAPICH2-X 38% faster

- **MVAPICH2-X-AWS 2.3**
  - Released on 08/12/2019
  - Major Features and Enhancements
    - Based on MVAPICH2-X 2.3
    - Support for Amazon EFA adapter's Scalable Reliable Datagram (SRD)
    - 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

- **MVAPICH2-Azure 2.3.2**
  - Released on 08/16/2019
  - Major Features and Enhancements
    - Based on MVAPICH2-2.3.2
    - Enhanced tuning for point-to-point and collective operations
    - Targeted for Azure HB & HC virtual machine instances
    - Flexibility for ‘one-click’ deployment
    - Tested with Azure HB & HC VM instances
    - Available for download from [http://mvapich.cse.ohio-state.edu/downloads/](http://mvapich.cse.ohio-state.edu/downloads/)
    - Detailed User Guide: [http://mvapich.cse.ohio-state.edu/userguide/mv2-azure/](http://mvapich.cse.ohio-state.edu/userguide/mv2-azure/)
MVAPICH2 – Future Roadmap and Plans for Exascale

- Update to MPICH 3.3.2 CH3 channel
  - 2020
- Initial support for the CH4 channel
  - 2020/2021
- Making CH4 channel default
  - 2021/2022
- 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
Thank You!

subramoni.1@osu.edu
http://web.cse.ohio-state.edu/~subramon

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

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

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