Achieving High Performance on AWS HPC Cloud using MVAPICH2-AWS

Presenter: Dhabaleswar K (DK) Panda

The Ohio State University
panda@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~panda

11/19/19
© 2019
Agenda

• Overview of the MVAPICH2 Project
• Overview of Amazon Elastic Fabric Adapter (EFA)
• Designing MVAPICH2 for EFA
• Performance Results
• Software Release and Future Plans
Overview of the MVAPICH2 MPI Library 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 (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 (June ‘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
  - 15th, 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](http://mvapich.cse.ohio-state.edu)

Empowering Top500 systems for over a decade  

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

Event:
- MV 0.9.4
- MV2 0.9.0
- MV2 0.9.8
- MV2 1.0
- MV 1.0
- MV2 1.0.3
- MV 1.1
- MV2 1.1.5
- MV2 1.1.6
- MV2 1.1.7
- MV2 1.2
- MV2 1.3
- MV2 1.4
- MV2 1.5
- MV2 1.6
- MV2 1.7
- MV2 1.8
- MV2 1.9
- MV2-GDR 2.0b
- MV2-MIC 2.0
- OSU INAM 0.9.3
- MV2-GDR 2.3.2
- MV2-X 2.3 rc2
- MV2 Virt 2.2
- MV2 Virt 2.2.1
- MV2-GDR 2.3.2
- MV2-GDR 2.3.2
- MV2-GDR 2.3.2
- MV2-GDR 2.3.2
- MV2-GDR 2.3.2
- MV2-AWS 2.3
## 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, EFA)*
- 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 Protocols: CMA, IVSHMEM, XPMEM
- Modern Features: NVLink, CAPI*
Agenda

• Overview of the MVAPICH2 Project
• **Overview of Amazon Elastic Fabric Adapter (EFA)**
• Designing MVAPICH2 for EFA
• Performance Results
• Software Release and Future Plans

© 2019
Amazon Elastic Fabric Adapter (EFA)

- Enhanced version of Elastic Network Adapter (ENA)
- Allows OS bypass, up to 100 Gbps bandwidth
- Network aware multi-path routing
- Exposed through libibverbs and libfabric interfaces
- Introduces new Queue-Pair (QP) type
  - Scalable Reliable Datagram (SRD)
  - Also supports Unreliable Datagram (UD)
  - No support for Reliable Connected (RC)
## Comparison with IB Transport Types and Trade-offs

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reliable Connection</th>
<th>Reliable Datagram</th>
<th>Dynamic Connected</th>
<th>Scalable Reliable Datagram (SRD)</th>
<th>Unreliable Connection</th>
<th>Unreliable Datagram</th>
<th>Raw Datagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability (M processes, N nodes)</td>
<td>M^2N QPs per HCA</td>
<td>M QPs per HCA</td>
<td>M QPs per HCA</td>
<td>M QPs per HCA</td>
<td>M^2N QPs per HCA</td>
<td>M QPs per HCA</td>
<td>1 QP per HCA</td>
</tr>
<tr>
<td>Corrupt data detected</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Delivery Guarantee</td>
<td>Data delivered exactly once</td>
<td></td>
<td></td>
<td>No guarantees</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Order Guarantees</td>
<td>Per connection</td>
<td>One source to multiple destinations</td>
<td>Per connection</td>
<td>No</td>
<td>Unordered, duplicate data detected</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Data Loss Detected</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error Recovery</td>
<td>Errors (retransmissions, alternate path, etc.) handled by transport layer. Client only involved in handling fatal errors (links broken, protection violation, etc.)</td>
<td></td>
<td></td>
<td>Errors are reported to responder</td>
<td></td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
Scalable Reliable Datagrams (SRD): Features & Limitations

<table>
<thead>
<tr>
<th>Feature</th>
<th>UD</th>
<th>SRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send/Recv</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Send w/ Immediate</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>RDMA</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Read/Write/Atomic</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Scatter Gather Lists</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Shared Receive Queue</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Reliable Delivery</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Ordering</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Inline Sends</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Global Routing Header</td>
<td>✔️</td>
<td>✗</td>
</tr>
<tr>
<td>Max Message Size</td>
<td>4KB</td>
<td>8KB</td>
</tr>
</tbody>
</table>

- **Similar to IB Reliable Datagram**
  - No limit on number of outstanding messages per context

- **Out of order delivery**
  - No head-of-line blocking
  - Bad fit for MPI, can suit other workloads

- **Packet spraying over multiple ECMP paths**
  - No hotspots
  - Fast and transparent recovery from network failures

- **Congestion control designed for large scale**
  - Minimize jitter and tail latency
Verbs level evaluation of EFA performance

- SRD adds 8-10% overhead compared to UD
- Due to hardware based acks used for reliability

- Instance type: c5n.18xlarge
- CPU: Intel Xeon Platinum 8124M @ 3.00GHz
Agenda

• Overview of the MVAPICH2 Project
• Overview of Amazon Elastic Fabric Adapter (EFA)
• **Designing MVAPICH2 for EFA**
• Performance Results
• Software Release and Future Plans
Designing MPI libraries for EFA

- **Shared Memory Model**
  - SHMEM, DSM

- **Distributed Memory Model**
  - MPI (Message Passing Interface)
  - Partitioned Global Address Space (PGAS)
    - OpenSHMEM, UPC, UPC++, CAF …

- MPI offer various communication primitives
  - Point-to-point, Collective, Remote Memory Access
  - Provides strict guarantees about reliability and ordering
  - Allows message sizes much larger than allowed by the network

- How to address these semantic mismatches between the network and programming model in a scalable and high-performance manner?
Handled three Major Challenges

- Reliable and in-order delivery
- Zero-copy transmission of large messages
- Handling out-of-order packets for zero-copy transfers

S. Chakraborty, S. Xu, H. Subramoni and D. K. Panda, Designing Scalable and High-Performance MPI Libraries on Amazon Elastic Adapter, Hot Interconnect, 2019
Agenda

• Overview of the MVAPICH2 Project
• Overview of Amazon Elastic Fabric Adapter (EFA)
• Designing MVAPICH2 for EFA
• **Performance Results**
• Software Release and Future Plans
Experimental Setup

• Instance type: c5n.18xlarge
• CPU: Intel Xeon Platinum 8124M @ 3.00GHz
• Cores: 2 Sockets, 18 cores / socket
• KVM Hypervisor, 192 GB RAM, One EFA adapter / node
• MVAPICH2 version: Latest MVAPICH2-X + SRD support
• OpenMPI version: Open MPI v4.0.2 with libfabric 1.8
• OMB version: OSU Micro-Benchmarks 5.6.2
Point-to-Point Performance

- Up to 1.8x better than OpenMPI on large message sizes
Collectives (4-Node-128-Processes)

- Up to 26x lower all-reduce latency with MVAPICH2-X than with OpenMPI
- Up to 15x lower bcast latency with MVAPICH2-X than with OpenMPI
Collectives (4-Node-128-Processes) Cont’d

- Up to 38x lower reduce latency with MVAPICH2-X than with OpenMPI
- Up to 18x lower scatter latency with MVAPICH2-X than with OpenMPI
Collectives (16-Node-256-Processes)

- Up to 87x lower allreduce latency with MVAPICH2-X than with OpenMPI
- Up to 5.5x lower bcast latency with MVAPICH2-X than with OpenMPI
Collectives (16-Node-256-Processes) Cont’d

- Up to 37x lower reduce latency with MVAPICH2-X than with OpenMPI
- Up to 83x lower scatter latency with MVAPICH2-X than with OpenMPI
• Up to 23% better performance with 3D-stencil on 16 nodes
• Up to 10% performance improvement for Cloverleaf on 16 nodes

S. Chakraborty, S. Xu, H. Subramoni and D. K. Panda, Designing Scalable and High-Performance MPI Libraries on Amazon Elastic Adapter, Hot Interconnect, 2019
Agenda

• Overview of the MVAPICH2 Project
• Overview of Amazon Elastic Fabric Adapter (EFA)
• Designing MVAPICH2 for EFA
• Performance Results

• **Software Release and Future Plans**
Software Release and Future Plans

• MVAPICH2-X for AWS 2.3 released on 04/12/2019
  • Includes support for SRD and XPMEM based transports
  • Available for download from http://mvapich.cse.ohio-state.edu/downloads/
  • Detailed User Guide: http://mvapich.cse.ohio-state.edu/userguide/mv2x-aws/

• Working on
  • Additional optimizations and tuning, a new version will be released soon
  • Making it available in an integrated manner in the AWS portal
  • Making it available through AWS Market Place

• Commercial Support available for End-Users, ISVs, and Organizations
  • Through X-ScaleSolutions (http://x-scalesolutions.com)
Thank You!

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

Follow us on
https://twitter.com/mvapich

The MVAPICH2 Project
http://mvapich.cse.ohio-state.edu/