Designing a Deep-Learning Aware MPI Library: An MVAPICH2 Approach

Ammar Ahmad Awan, Jahanzeb Maqbool Hashmi, Ching-Hsiang Chu, Hari Subramoni, and Dhabaleswar K. (DK) Panda

The Ohio State University

http://nowlab.cse.ohio-state.edu
WHAT IS DEEP LEARNING?

• Deep Learning (DL)
  – A subset of Machine Learning that uses Deep Neural Networks (DNNs)
  – Perhaps, the most revolutionary subset!
• Based on learning data representation
• Examples Convolutional Neural Networks, Recurrent Neural Networks, Hybrid Networks
• Data Scientist or Developer Perspective
  1. Identify DL as solution to a problem
  2. Determine Data Set
  3. Select Deep Learning Algorithm to Use
  4. Use a large data set to train an algorithm


DL-Aware MPI - OFAWS 2020
DEEP LEARNING AND HIGH-PERFORMANCE ARCHITECTURES

- NVIDIA GPUs are the main driving force for faster training of DL models
  - The ImageNet Challenge - (ILSVRC) -- 90% of the teams used GPUs (2014)
  - Deep Neural Networks (DNNs) like ResNet(s) and Inception

- However, High Performance Architectures for DL and HPC are evolving
  - 135/500 Top HPC systems use NVIDIA GPUs (Nov ’19)
  - DGX-1 (Pascal) and DGX-2 (Volta)
    - Dedicated DL supercomputers
  - Cascade-Lake Xeon CPUs have 28 cores/socket (TACC Frontera – #5 on Top500)
  - AMD EPYC (Rome) CPUs have 64 cores/socket (Upcoming DOE Clusters)
  - AMD GPUs will be powering the Frontier – DOE’s Exascale System at ORNL
  - Domain Specific Accelerators for DNNs are also emerging

*https://blogs.nvidia.com/blog/2014/09/07/imagenet/
How to efficiently scale-out Deep Learning (DL) workloads by better exploiting High Performance Computing (HPC) resources like Multi-/Many-core CPUs and GPUs?
HIGH-PERFORMANCE DISTRIBUTED DATA PARALLEL TRAINING WITH TENSORFLOW

- **gRPC**
  - Officially available and supported
  - Open-source – can be enhanced by others
  - Accelerated gRPC (add RDMA to gRPC)

- **gRPC+X**
  - Use gRPC for bootstrap and rendezvous
  - *Actual communication is in “X”*
  - X → MPI, Verbs, GPUDirect RDMA (GDR), etc.

- **No-gRPC**
  - Baidu – the first one to use MPI Collectives for TF
  - Horovod – Use NCCL, or MPI, or any other future library (e.g. IBM DDL support recently added)

---


---
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
- 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 HPCIB 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 3,090 organizations in 89 countries
- More than 761,000 (> 0.76 million) downloads from the OSU site directly
- Empowering many TOP500 clusters (Nov ‘19 ranking)
  - 3rd, 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China
  - 5th, 448, 448 cores (Frontera) at TACC
  - 8th, 391,680 cores (ABCI) in Japan
  - 14th, 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 5th ranked TACC Frontera system
- Empowering Top500 systems for more than 15 years
MVAPICH2 (MPI)-DRIVEN INFRASTRUCTURE FOR ML/DL TRAINING

More details from http://hidl.cse.ohio-state.edu
CPU-based Deep Learning

- Using MVAPICH2-X

GPU-based Deep Learning

- Using MVAPICH2-GDR
ResNet-152 Training performance

- BS=64, 4ppn is better; BS=32, 8ppn is slightly better
- However, keeping effective batch size (EBS) low is more important! – Why? (DNN does not converge to SOTA when batch size is large)

ResNet-152: Single Process (SP) vs. Multi-Process (MP)

- MP is better for all effective batch sizes
- Up to 1.35X better performance for MP compared to SP for BS=64.

Skylake-3 (48 cores, 96 threads)

- Scale—32 nodes
- MP-Tuned—up to 1.5X better than SP
- MP-Tuned—10% better than MP-Default
- Why MP-Tuned is better?
  - Uses the best possible number of inter-op and intra-op threads

<table>
<thead>
<tr>
<th>Model</th>
<th>IMGS/SEC</th>
<th>MP-Tuned</th>
<th>MP-Default</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESNET-50</td>
<td>3086</td>
<td>2900</td>
<td>152</td>
<td>1213</td>
</tr>
<tr>
<td>RESNET-101</td>
<td>1766</td>
<td>1718</td>
<td>1292</td>
<td>919</td>
</tr>
<tr>
<td>RESNET-152</td>
<td>1213</td>
<td>1197</td>
<td>1729</td>
<td>1245</td>
</tr>
<tr>
<td>INCEPTION-V3</td>
<td>2478</td>
<td>2271</td>
<td>1729</td>
<td>1283</td>
</tr>
<tr>
<td>INCEPTION-V4</td>
<td>832</td>
<td>1245</td>
<td>1128</td>
<td></td>
</tr>
</tbody>
</table>

• Scaled TensorFlow to 2,048 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 2048 nodes

• On 2048 nodes, ResNet-50 can be trained in 7 minutes!

• On single node, TensorFlow (TF) is 8% better than MXNet
• TF (tf_cnn_benchmark) is 2.13x better than PyTorch
• TensorFlow is 1.7x better than MXNet
• TF (Keras) gives better performance compared to PyTorch and MXNet.
PERFORMANCE OF CNTK WITH MVAPICH2-X ON CPU-BASED DEEP LEARNING

- CPU-based training of AlexNet neural network using ImageNet ILSVRC2012 dataset
- Advanced XPMEM-based designs show up to 20% benefits over Intel MPI (IMPI) for CNTK DNN training using All_Reduce
- The proposed designs show good scalability with increasing system size

Available since MVAPICH2-X 2.3rc1 release

Designing Efficient Shared Address Space Reduction Collectives for Multi-/Many-cores, J. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, and DK Panda, 32nd IEEE International Parallel & Distributed Processing Symposium (IPDPS '18), May 2018
BENCHMARKING HYPAR-FLOW ON STAMPEDE2

- CPU based **Hybrid-Parallel** (Data Parallelism and Model Parallelism) training on Stampede2
- Benchmark developed for various configuration
  - Batch sizes
  - No. of model partitions
  - No. of model replicas
- Evaluation on a very deep model
  - ResNet-1000 (a 1,000-layer model)


110x speedup on 128 Intel Xeon Skylake nodes (TACC Stampede2 Cluster)
- **ResNet-1001 with variable batch size**

- **Approach:**
  - 48 model-partitions for 56 cores
  - 512 model-replicas for 512 nodes
  - Total cores: 48 x 512 = 24,576

- **Speedup**
  - **253X** on 256 nodes
  - **481X** on 512 nodes

- **Scaling Efficiency**
  - **98%** up to 256 nodes
  - **93.9%** for 512 nodes

---

HIGH-PERFORMANCE DEEP LEARNING

- CPU-based Deep Learning
  - Using MVAPICH2-X

- GPU-based Deep Learning
  - Using MVAPICH2-GDR
OPTIMIZED MVAPICH2-GDR (GPUDIRECT RDMA) DESIGN

**GPU-GPU Inter-node Latency**
- **MV2-(NO-GDR)**
- **MV2-GDR 2.3.4**

Latency (us) vs. Message Size (Bytes)

1.85us, 10x

**GPU-GPU Inter-node Bandwidth**
- **MV2-(NO-GDR)**
- **MV2-GDR 2.3.4**

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

9x, 11x

**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
Optimized designs in MVAPICH2-GDR offer better/comparable performance for most cases

MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) on 1 DGX-2 node (16 Volta GPUs)

Platform: Nvidia DGX-2 system (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 10.1

~2.5X better

~4.7X better

**MVAPICH2-GDR VS. NCCL2 – ALLREDUCE ON GPU SYSTEMS (ABCI)**

- Optimized designs in upcoming MVAPICH2-GDR offer better performance for most cases
- MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) up to 128 GPUs (32 nodes on ABCI)

**ABCI Platform**: Dual-socket Intel Xeon Gold, 4 NVIDIA Volta V100 GPUs with NVLink, and two InfiniBand EDR Interconnect

---

**Small Messages - Latency on 128 GPUs**

- MVAPICH2-GDR-2.3.4 vs. NCCL 2.4
- Up to 8X better performance for small messages

**Large Messages - Latency on 128 GPUs**

- MVAPICH2-GDR-2.3.4 vs. NCCL 2.4
- Up to 1.6X better performance for large messages
Optimized designs in MVAPICH2-GDR offer better performance for most cases.

**MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) up to 1,536 GPUs**

Platform: Dual-socket IBM POWER9 CPU, 6 NVIDIA Volta V100 GPUs, and 2-port InfiniBand EDR Interconnect

SCALABLE TENSORFLOW USING HOROVOD AND MVAPICH2-GDR

- ResNet-50 Training using TensorFlow benchmark on 1 DGX-2 node (16 Volta GPUs)

Platform: Nvidia DGX-2 system, CUDA 10.1

![Image showing comparison between NCCL-2.5 and MVAPICH2-GDR-2.3.4]

9% higher

Scaling Efficiency = \( \frac{\text{Actual throughput}}{\text{Ideal throughput at scale}} \times 100\% \)

• ResNet-50 Training using TensorFlow benchmark on SUMMIT -- 1536 Volta GPUs!

• 1,281,167 (1.2 mil.) images

• Time/epoch = 3.6 seconds

• Total Time (90 epochs) = 3.6 x 90 = 332 seconds = 5.5 minutes!

Platform: The Summit Supercomputer (#1 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 9.2

*We observed errors for NCCL2 beyond 96 GPUs
Scalable PyTorch on ORNL/Summit using MVAPICH2-GDR

- PyTorch is becoming a very important DL framework
- Scaling PyTorch models with Horovod is simple
- MVAPICH2-GDR provides better performance and scalability compared to NCCL2

Platform: The Summit Supercomputer (#1 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 10.1

- RTX 5000 are NVIDIA’s GPUs targeted for data centers
- Different from GTX series
  - Supports GPUDirect RDMA (GDR)
  - Supports GDRCOPY
- MVAPICH2-GDR offers good performance and reasonable scaling
- Scaling is not as good as Lassen because
  - Nodes are connected with IB FDR
  - No NVLink between GPUs (only PCIe)

**Platform:** TACC Frontera-RTX – 4 NVIDIA RTX 5000 GPUs per node, CUDA 10.1
Scalable distributed training is getting important

Requires high-performance middleware designs while exploiting modern interconnects

Provided a set of MPI (MVAPICH2)-driven solutions to achieve scalable distributed training for TensorFlow, PyTorch and MXNet

More details on these solutions and usage are available from: http://hidl.cse.ohio-state.edu and http://mvapich.cse.ohio-state.edu

The proposed solutions will continue to enable the DL community to achieve scalability and high-performance for their distributed training
THANK YOU!

{awan.10, hashmi.29, chu.368}@osu.edu
subramon@cse.ohio-state.edu, 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 Deep Learning Project
http://hidl.cse.ohio-state.edu/