15th ANNUAL WORKSHOP 2019

ACCELERATING TENSORFLOW WITH RDMA FOR HIGH-PERFORMANCE DEEP LEARNING

Xiaoyi Lu, Dhabaleswar K. (DK) Panda

The Ohio State University

[ March 19, 2019 ]

E-mail: {luxi, panda}@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~luxi
http://www.cse.ohio-state.edu/˜panda
Deep Learning is a sub-set of Machine Learning

- But, it is perhaps the most radical and revolutionary subset

Deep Learning is going through a resurgence

- Model: Excellent accuracy for deep/convolutional neural networks
- Data: Public availability of versatile datasets like MNIST, CIFAR, and ImageNet
- Capability: Unprecedented computing and communication capabilities: Multi-/Many-Core, GPGPUs, Xeon Phi, InfiniBand, RoCE, etc.

Big Data has become one of the most important elements in business analytics

- Increasing demand for getting Big Value out of Big Data to drive the revenue continuously growing
APPLICATION EXAMPLE: FLICKR’S MAGIC VIEW PHOTO FILTERING

- Image recognition to divide pictures into surprisingly accurate categories
- Magic of AI/DL: Generate accurate tags for billions of pictures
EXAMPLES OF DEEP LEARNING STACKS

- TensorFlow
- Caffe/Caffe2
- Torch
- SparkNet
- TensorFlow
- DeepLearning4J
- BigDL
- CNTK
- mmlspark
- Many others…
TRENDS OF DEEP LEARNING STACKS

- Google TensorFlow
- Microsoft CNTK
- Facebook Caffe2 and PyTorch

- Google Search Trend (March, 2019)
INCREASING USAGE OF HPC, BIG DATA AND DEEP LEARNING

HPC
(MPI, RDMA, Lustre, etc.)

Big Data
(Hadoop, Spark, HBase, Memcached, etc.)

Deep Learning
(Caffe, TensorFlow, BigDL, etc.)

Convergence of HPC, Big Data, and Deep Learning!!!
HIGHER-TECHNICAL UNDERLYING LIBRARIES WITH HPC TECHNOLOGIES

- **BLAS Libraries** – the heart of math operations
  - Atlas/OpenBLAS
  - NVIDIA cuBlas
  - Intel Math Kernel Library (MKL)

- **DNN Libraries** – the heart of Convolutions!
  - NVIDIA cuDNN (already reached its 7th iteration – cudnn-v7)
  - Intel MKL-DNN (MKL 2017) – recent but a very promising development

- **Communication Libraries** – the heart of model parameter updating
  - RDMA
  - GPUDirect RDMA

---

OUTLINE

- Overview of TensorFlow and gRPC
  - Accelerating gRPC and TensorFlow with RDMA
  - Benchmarking gRPC and TensorFlow
  - Performance Evaluation
  - Conclusion
ARCHITECTURE OVERVIEW OF GOOGLE TENSORFLOW

Key Features:

- Widely used for Deep Learning
- Open source software library for numerical computation using data flow graphs
- Graph edges represent the multidimensional data arrays
- Nodes in the graph represent mathematical operations
- Flexible architecture allows to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API
- Used by Google, Airbnb, Dropbox, Snapchat, Twitter
- Communication and Computation intensive

Source: https://www.tensorflow.org/
ARCHITECTURE OVERVIEW OF GRPC

- Key Features:
  - Simple service definition
  - Works across languages and platforms
    - C++, Java, Python, Android Java etc
    - Linux, Mac, Windows
  - Start quickly and scale
  - Bi-directional streaming and integrated authentication
  - Used by Google (several of Google’s cloud products and Google externally facing APIs, TensorFlow), NetFlix, Docker, Cisco, Juniper Networks etc.
  - Uses sockets for communication!

Large-scale distributed systems composed of micro services

Source: http://www.grpc.io/
Worker services communicate among each other using gRPC, or gRPC+X!
THE HIGH-PERFORMANCE BIG DATA (HIBD) PROJECT

- RDMA for Apache Spark
- RDMA for Apache Hadoop 3.x (RDMA-Hadoop-3.x)
- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
  - Plugins for Apache, Hortonworks (HDP) and Cloudera (CDH) Hadoop distributions
- RDMA for Apache Kafka
- RDMA for Apache HBase
- RDMA for Memcached (RDMA-Memcached)
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- OSU HiBD-Benchmarks (OHB)
  - HDFS, Memcached, HBase, and Spark Micro-benchmarks
- http://hibd.cse.ohio-state.edu
- Users Base: 300 organizations from 35 countries
- More than 29,350 downloads from the project site

Available for InfiniBand and RoCE
Also run on Ethernet

Available for x86 and OpenPOWER
Support for Singularity and Docker
Can similar designs be done for gRPC and TensorFlow to achieve significant performance benefits by taking advantage of native RDMA support?

How do we benchmark gRPC and TensorFlow for both deep learning and system researchers?

What kind of performance benefits we can get through native RDMA-based designs in gRPC and TensorFlow?
OUTLINE

- Overview of TensorFlow and gRPC
  - Accelerating gRPC and TensorFlow with RDMA
  - Benchmarking gRPC and TensorFlow
  - Performance Evaluation
  - Conclusion
Tensor communication over gRPC channel

- **Rendezvous protocol**
  - TensorFlow worker (tensor receiving process) actively requests for tensors to the parameter server (tensor sending process)

- **Worker issues Tensor RPC request that to Parameter Server (PS)**

- **PS finds the requested tensor, and responds to the worker**

- **gRPC core uses recvmsg and sendmsg primitives for receiving and sending payloads**

- **Tensor Transmission uses iovec structures**

---

HIGH PERFORMANCE TENSOR COMMUNICATION CHANNEL

- **gRPC + Verbs**
  - Dedicated verbs channel for tensor communication
  - gRPC channel for administrative task communication

- **gRPC + MPI**
  - Dedicated MPI channel for tensor communication
  - gRPC channel for administrative task communication

- **Uber Horovod**
  - Uber’s approach of MPI based distributed TensorFlow

- **Baidu Tensorflow-Allreduce**
  - Baidu’s approach of MPI based distributed TensorFlow
- Small, Medium and Large indicate buffers of few Bytes, KBytes and MBytes of length

- gRPC payload may contain a uniform distribution of such Small buffers

- A lot of Large buffers and a few Small buffers may create a skew distribution of such buffers in one gRPC payload

---

OSU AR-GRPC AND AR-GRPC ENHANCED TENSORFLOW

- Adaptive RDMA gRPC

- Features
  - Hybrid Communication engine
  - Adaptive protocol selection between eager and rendezvous

- Message pipelining and coalescing
  - Adaptive chunking and accumulation
  - Intelligent threshold detection
  - Zero copy transmission
  - Zero copy send/recv

Overview of TensorFlow and gRPC
Accelerating gRPC and TensorFlow with RDMA
Benchmarking gRPC and TensorFlow
Performance Evaluation
Conclusion
# AVAILABLE BENCHMARKS, MODELS, AND DATASETS

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>CIFAR-10</th>
<th>ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Digit Classification</td>
<td>Object Classification</td>
<td>Object Classification</td>
</tr>
<tr>
<td>Resolution</td>
<td>28 × 28 B&amp;W</td>
<td>32 × 32 Color</td>
<td>256 × 256 Color</td>
</tr>
<tr>
<td>Classes</td>
<td>10</td>
<td>10</td>
<td>1000</td>
</tr>
<tr>
<td>Training Images</td>
<td>60 K</td>
<td>50 K</td>
<td>1.2 M</td>
</tr>
<tr>
<td>Testing Images</td>
<td>10 K</td>
<td>10 K</td>
<td>100 K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Layers (Conv. / Full-connected)</th>
<th>Dataset</th>
<th>Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet</td>
<td>2 / 2</td>
<td>MNIST</td>
<td>TensorFlow, CaffeOnSpark, TensorFlowOnSpark</td>
</tr>
<tr>
<td>SoftMax Regression</td>
<td>NA / NA</td>
<td>MNIST</td>
<td>TensorFlow, TensorFlowOnSpark</td>
</tr>
<tr>
<td>CIFAR-10 Quick</td>
<td>3 / 1</td>
<td>CIFAR-10</td>
<td>CaffeOnSpark, TensorFlowOnSpark, MMLSpark</td>
</tr>
<tr>
<td>VGG-16</td>
<td>13 / 3</td>
<td>CIFAR-10</td>
<td>TensorFlow, BigDL</td>
</tr>
<tr>
<td>AlexNet</td>
<td>5 / 3</td>
<td>ImageNet</td>
<td>TensorFlow, CaffeOnSpark</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>22 / 0</td>
<td>ImageNet</td>
<td>TensorFlow, CaffeOnSpark</td>
</tr>
<tr>
<td>Resnet-50</td>
<td>53/1</td>
<td>Synthetic</td>
<td>TensorFlow</td>
</tr>
</tbody>
</table>
ARE CURRENT BENCHMARKS SUFFICIENT?

• Current DL models and benchmarks are deep learning research oriented
  • Example: Facebook caffe2 takes 1 hour to train ImageNet data

• However, many system researchers are focused on improving the communication engine of deep learning frameworks
  • A fast benchmark that models deep learning characteristics is highly desirable

TENSORFLOW DL MICRO-BENCHMARKS FOR GRPC

TF-gRPC-Bench Design Considerations

- Number of Parameter Servers and Workers
- Resource Utilization
- Distribution of iovec buffers
- Size and Number of iovec Buffers
- Mode

User Parameters
Deploy Cluster

Parameter Server
- Spawn Server
- Handle RPC
- Generate Payload
- Respond
- Generate Results

Worker
- Spawn Client
- Generate Payload
- Issue RPC
- Handle Response
- Generate Results

Gather Result
Performance Results

OUTLINE

- Overview of TensorFlow and gRPC
- Accelerating gRPC and TensorFlow with RDMA
- Benchmarking gRPC and TensorFlow
- Performance Evaluation
- Conclusion
• AR-gRPC (OSU design) Latency on SDSC-Comet-FDR
  – Up to 2.7x performance speedup over Default gRPC (IPoIB) for Latency for small messages.
  – Up to 2.8x performance speedup over Default gRPC (IPoIB) for Latency for medium messages.
  – Up to 2.5x performance speedup over Default gRPC (IPoIB) for Latency for large messages.

• OSU-RI2-IB-EDR: AR-gRPC (RDMA) reduces latency by 59% and 56% compared to Default gRPC over 40G Ethernet and IPoIB
• SDSC-Comet-IB-FDR: AR-gRPC (RDMA) reduces 78% latency compared to 10G (Default gRPC) Ethernet and 69% compared to IPoIB (Default gRPC)
OSU-R12-IB-EDR

• OSU-R12-IB-EDR: AR-gRPC (RDMA) gRPC achieves a 3.4x speedup compared to Default gRPC over IPoIB for uniform scheme

SDSC-Comet-IB-FDR

• SDSC-Comet-IB-FDR: AR-gRPC (RDMA) achieves 3.6x bandwidth compared to Default gRPC over IPoIB for uniform scheme
PERFORMANCE BENEFITS FOR AR-GRPC WITH TENSORFLOW MIMIC TEST

- **AR-gRPC (OSU design)** TensorFlow Mimic test on SDSC-Comet-FDR
  - Up to 60% reduction in average latency over Default gRPC (IPOIB)
  - Up to 2.68x performance speedup over Default gRPC (IPOIB)
EVALUATION OF TENSORFLOW: GOOGLENET & ALEXNET

GoogleNet & AlexNet Evaluation on OSU-RI2-IB-EDR (Higher Better); TotalBatchSize = (BatchSize/GPU) × NUMofGPUs

- GoogleNet has only 5 Million parameters, whereas AlexNet has about 60 Million parameters
- AR-gRPC scales better as we go from 4 nodes to 8 nodes
- For large batch size (32/GPU, total 224) the GoogleNet improvement is about 15% (597 vs 517)
  - GoogleNet results in less network intensive gradient updates
- However, AR-gRPC shows 89% (124 vs 65) performance improvement for Alexnet compared to default gRPC
EVALUATION OF TENSORFLOW: INCEPTION-V4

Inception4 Evaluation on Cluster A (Higher Better); TotalBatchSize = (BatchSize/GPU) × NUMofGPUs

- AR-gRPC improves TensorFlow performance by a maximum of 29%, 80%, and 144% compared to default gRPC on 4, 8, and 12 nodes, respectively
  - For example: Improvement of 80% (93 vs 51 images) for batch size 16/GPU (total 176) on 12 nodes
- AR-gRPC process a maximum of 27%, 12%, and 31% more images than Verbs channel
- AR-gRPC outperforms MPI channel by a maximum of 29%, 151%, and 228% for 4, 8, and 12 nodes
EVALUATION OF TENSORFLOW: RESNET152

Resnet152 Evaluation on Cluster A (Higher Better); TotalBatchSize = (BatchSize/GPU)×NUMofGPUs

- AR-gRPC accelerates TensorFlow by 62% (batch size 8/GPU) more compared to default gRPC on 4 nodes
- AR-gRPC improves Resnet152 performance by 32% (batch size 32/GPU) to 147% on 8 nodes
- AR-gRPC incurs a maximum speedup of 3x (55 vs 18 images) compared to default gRPC 12 nodes
  - Even for higher batch size of 32/GPU (total 352) AR-gRPC improves TensorFlow performance by 82% 12 nodes
- AR-gRPC processes a maximum of 40%, 35%, and 30% more images, on 4, 8, and 12 nodes, respectively, than Verbs
- AR-gRPC achieves a maximum speedup of 1.61x, 3.3x and 4.5x compared to MPI channel on 4, 8, and 12 nodes, respectively
AR-GRPC SPEEDUP COMPARED TO DEFAULT GRPC

![Speedup Comparison Graph]

- CNNs: AlexNet, GoogleNet, VGG16, Resnet50, Resnet152, Inception4
- Speedup comparison bar chart showing relative performance improvements for different CNN models.
OSU RDMA-TENSORFLOW DISTRIBUTION

- **High-Performance Design of TensorFlow over RDMA-enabled Interconnects**
  - High performance RDMA-enhanced design with native InfiniBand support at the verbs-level for gRPC and TensorFlow
  - RDMA-based data communication
  - Adaptive communication protocols
  - Dynamic message chunking and accumulation
  - Support for RDMA device selection
  - Easily configurable for different protocols (native InfiniBand and IPoIB)

- **Current release: 0.9.1**
  - Based on Google TensorFlow 1.3.0
  - Tested with
    - Mellanox InfiniBand adapters (e.g., EDR)
    - NVIDIA GPGPU K80
    - Tested with CUDA 8.0 and CUDNN 5.0
  - [http://hidl.cse.ohio-state.edu](http://hidl.cse.ohio-state.edu)
OUTLINE

- Overview of TensorFlow and gRPC
- Accelerating gRPC and TensorFlow with RDMA
- Benchmarking gRPC and TensorFlow
- Performance Evaluation
- Conclusion
CONCLUSION

- Present architecture overview of TensorFlow and gRPC
- Discuss challenges in accelerating and benchmarking TensorFlow and gRPC
- RDMA can benefit DL workloads as showed by our AR-gRPC and the corresponding enhanced TensorFlow
  - Unified high-performance communication runtime throughout the TensorFlow stack
    - Up to 4.1x speedup compared to the default gRPC
    - Up to 3x performance improvement on TensorFlow when using AR-gRPC compared to default gRPC channel
    - Significant improvement over Verbs and MPI channel
    - Consistently good performance for different CNNs
- Plan to explore TensorFlow runtime to find more bottlenecks
- Our work is publicly available: http://hidl.cse.ohio-state.edu/
15th ANNUAL WORKSHOP 2019

THANK YOU

Xiaoyi Lu, Dhabaleswar K. (DK) Panda

The Ohio State University

E-mail: {luxi, panda}@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~luxi
http://www.cse.ohio-state.edu/~panda