HiBD
High-Performance Big Data

Characterizing and Benchmarking Deep Learning Systems on Modern Data Center Architectures

Talk at Bench 2018

by

Xiaoyi Lu

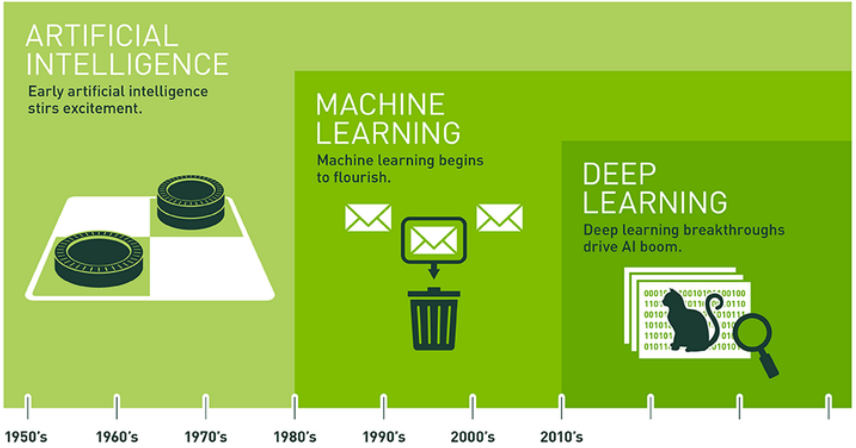
The Ohio State University

E-mail: luxi@cse.ohio-state.edu

http://www.cse.ohio-state.edu/~luxi

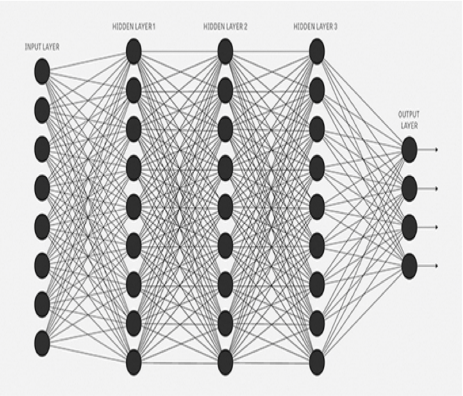
Overview of Deep Learning

- **Deep Learning** is a sub-set of Machine Learning
  – 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.



Courtesy: http://www.zdnet.com/article/caffe2-deep-learning-wide-ambitions-flexibility-scalability-and-advocacy/

MNIST handwritten digits

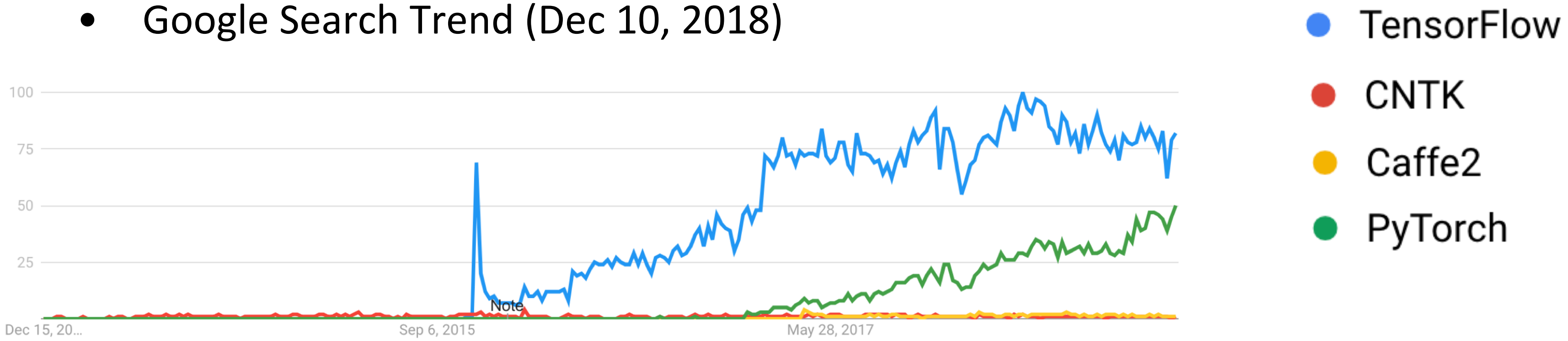


Deep Neural Network

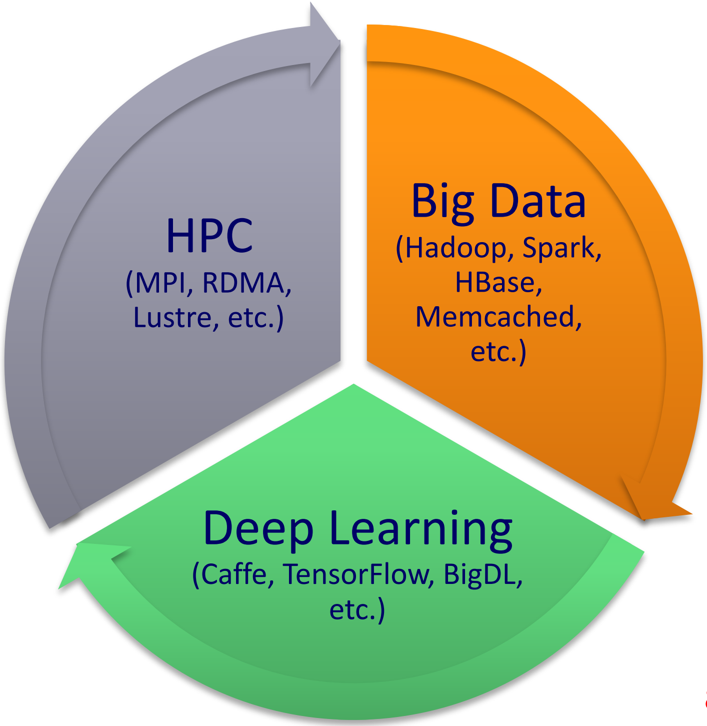
Trends of Deep Learning Systems

- Google TensorFlow
- Microsoft CNTK
- Facebook Caffe2
- PyTorch

- Google Search Trend (Dec 10, 2018)



Increasing Usage of HPC, Big Data and Deep Learning on Modern Datacenters



Convergence of HPC, Big Data, and Deep Learning!

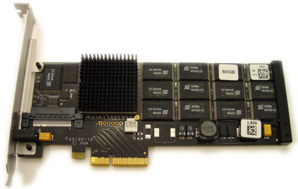
Increasing Need to Run these applications on the Cloud!!

Drivers of Modern Data Center Architecture

Multi-/Many-core Processors

High Performance Interconnects – InfiniBand (with SR-IOV)
<1usec latency, 200Gbps Bandwidth>

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



SSD, NVMe-SSD, NVRAM

- Multi-core/many-core technologies
- **Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand, iWARP, RoCE, and Omni-Path)**
- Single Root I/O Virtualization (SR-IOV)
- Solid State Drives (SSDs), NVMe/NVMf, Parallel Filesystems, Object Storage Clusters
- Accelerators (NVIDIA GPGPUs and FPGAs)

SDSC Comet

TACC Stampede

Microsoft Azure

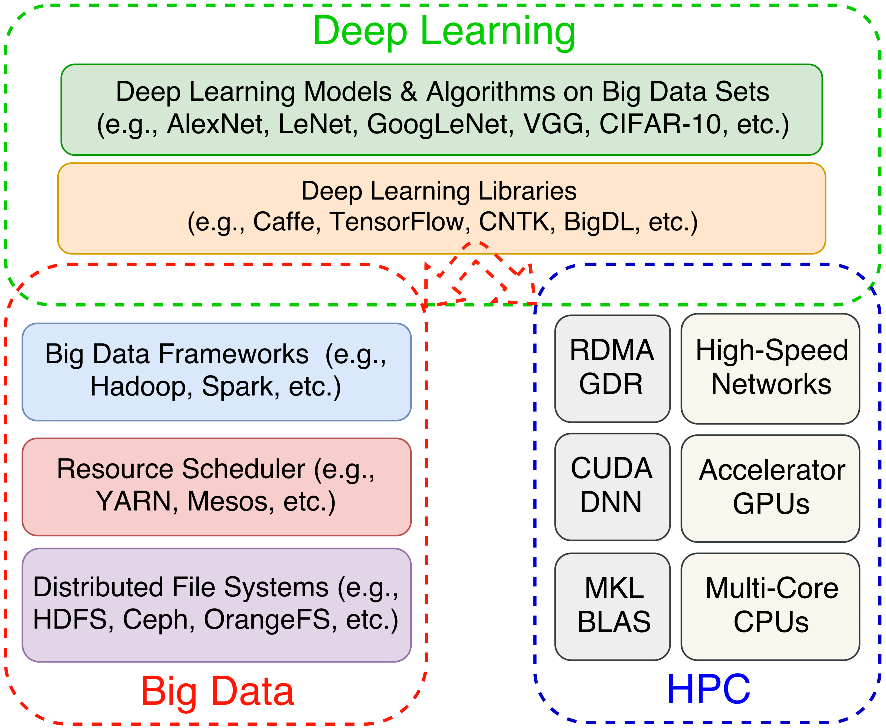
amazon web services EC2

ORACLE Cloud

Chameleon Cloud

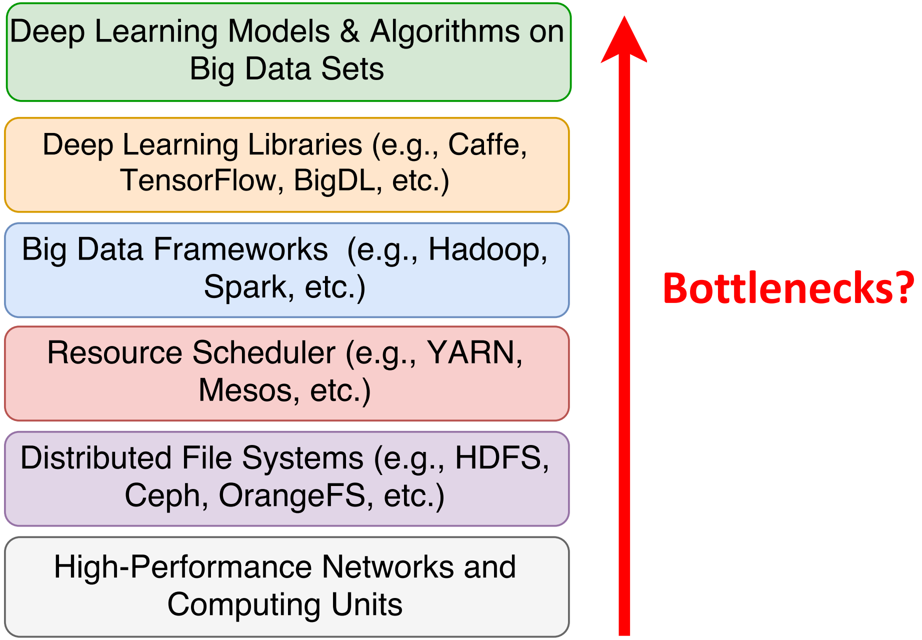
Modern Deep Learning System Architecture

- BLAS Libraries – the heart of math operations
  - Atlas/OpenBLAS
  - NVIDIA cuBlas
  - Intel Math Kernel Library (MKL)
- DNN Libraries – the heart of Convolutions!
  - NVIDIA cuDNN
  - Intel MKL-DNN
- Communication Libraries – the heart of model parameter updating
  - MPI
  - gRPC
  - RDMA / GPUDirect RDMA



Example: Overview of DLoBD Stacks

- Layers of DLoBD Stacks
  - Deep learning application layer
  - Deep learning library layer
  - Big data analytics framework layer
  - Resource scheduler layer
  - Distributed file system layer
  - Hardware resource layer
- Where are the bottlenecks for deep learning jobs?



A Quick Survey on Current Deep Learning Bechmarks

- **Stanford DAWNBench**
  - An open-source benchmark and competition for end-to-end deep learning training and inference
  - End-to-end training time, cost, accuracy
  - Support TensorFlow and PyTorch
  - Various types of hardware, like GPU, CPU
  - https://dawn.cs.stanford.edu/benchmark/
- **Baidu DeepBench**
  - An open-source benchmark covering both training and inference
  - Performance of basic operations in neural network libraries
  - Determining the most suitable hardware for specific operations, and communicating requirements to hardware manufacturers
  - Various types of hardware, like GPU, CPU, mobile devices
  - https://github.com/baidu-research/DeepBench

A Quick Survey on Current Deep Learning Benchmarks (*Cont.*)

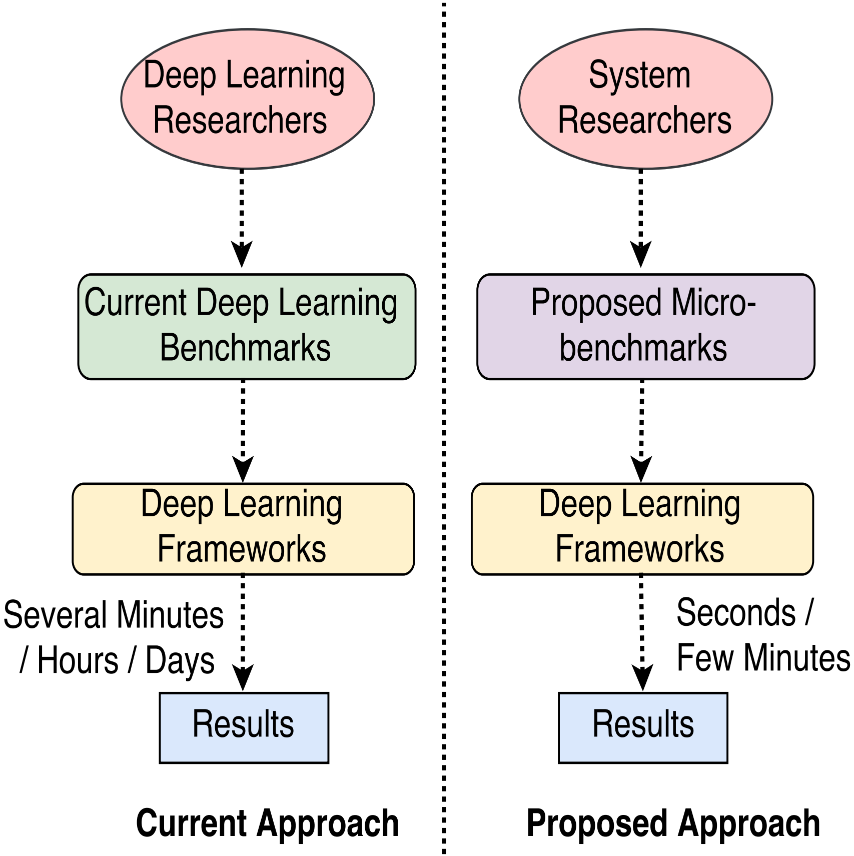
- **Facebook AI Performance Evaluation Platform**
  - Compare Machine Learning or Deep Learning inferencing performance metrics on a set of models over different backends
  - Total execution time, error rate, and power consumption
  - Support Caffe2 and TFLite
  - Various types of hardware, like GPU, CPU, DSP, mobile devices
  - https://github.com/facebook/FAI-PEP
- **ICT BigDataBench 4.0**
  - A comprehensive Big Data and AI benchmark suite
  - *Data motifs*, which considers any Big Data and AI workload as a pipeline of one or more classes of computation units performed on different input data sets
  - Eight data motifs, including Matrix, Sampling, Logic, Transform, Set, Graph, Sort, and Statistic computation
  - Support TensorFlow and Caffe
  - http://prof.ict.ac.cn/

Many Other Benchmarks

- **MLPerf**: a synthetic benchmark suite for measuring the performance of software frameworks, hardware accelerators, and cloud platforms for machine learning
- **Fathom**: a set of reference implementations of state-of-the-art deep learning models and has the ability to provide a quantitative analysis of the fundamental computational characteristics of these workloads
- **TensorFlow Benchmark**: a selection of image classification models across multiple platforms
- **CortexSuite**: a Synthetic Brain Benchmark Suite which classifies and identifies benchmarks by analogy to the human neural processing functions
- **BenchNN:** a hardware-based neural network accelerator can be compatible with many of the emerging benchmarks for high-performance micro-architectures
- **DjiNN:** an open infrastructure for providing Deep Neural Networks (DNN) as a service
- **Tonic**: provides image, speech, and natural language processing applications that can have a common DNN backend

Motivation

- Current DL models and benchmarks are deep learning research oriented.
  - Example: Facebook caffe2 takes 1 hour to train ImageNet data[1]
- System researchers just focus on improving the computation and communication engine of deep learning systems
  - A fast benchmark that models deep learning characteristics is highly desirable
  - Understanding the cross-layer activities



1. Goyal, Priya, et al. "Accurate, large minibatch SGD: training imagenet in 1 hour." arXiv preprint arXiv:1706.02677 (2017).

Case Studies - Characterizing and Benchmarking TensorFlow

- Standalone TensorFlow
- TensorFlow on Spark

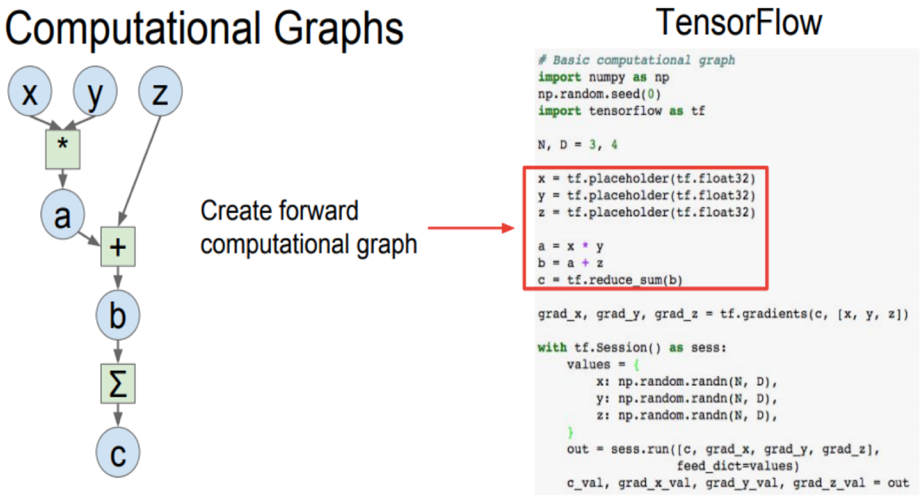
Overview of TensorFlow

Image courtesy: http://cs231n.stanford.edu/

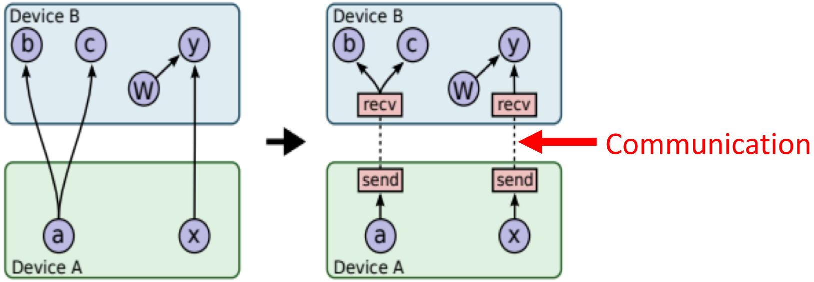
Key Features:

- Widely used for Deep Learning
- Open source software library for numerical computation using data flow graphs
- Nodes in the graph represent mathematical operations
- Graph edges represent the multidimensional data arrays

- Flexible architecture allows to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device
- Used by Google, Airbnb, DropBox, Snapchat, Twitter, and many other companies
- Communication and Computation intensive

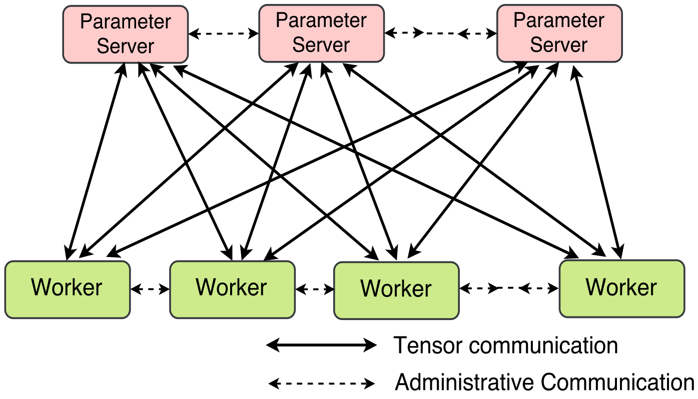


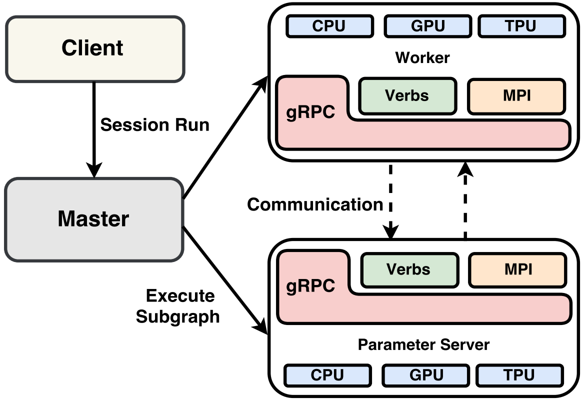
M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., "TensorFlow: A System for Large-Scale Machine Learning." in OSDI, vol. 16, 2016, pp. 265–283.



Before and After Usage of Distributed TF

Overview of Distributed Execution

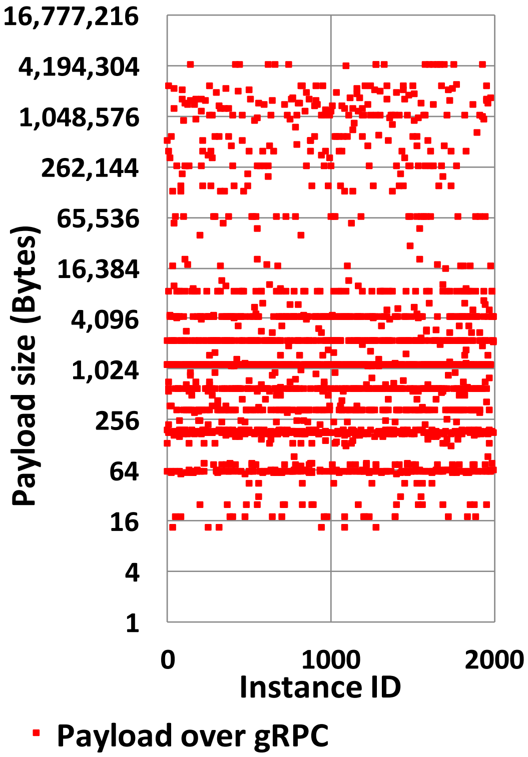




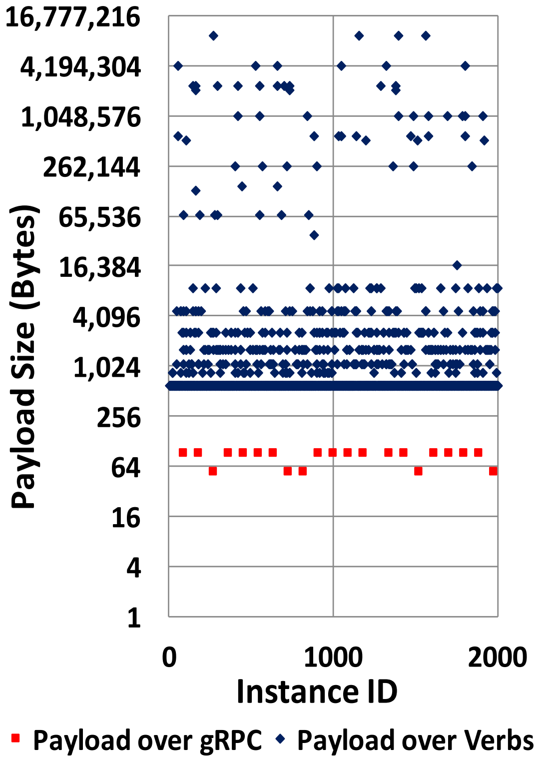
TensorFlow PS Architecture

- Training variables are updated using aggregated gradients and deltas, represented as tensors
- Widely used approach for managing the training variables is Parameter Server
- Parameter Server (PS) owns the master copies of the variables
- Workers request for those variables when needed
- Workers compute (such as gradient updates) a new value of a variable, it sends an update to the PS
- Variable updates (tensor updates) are communication intensive

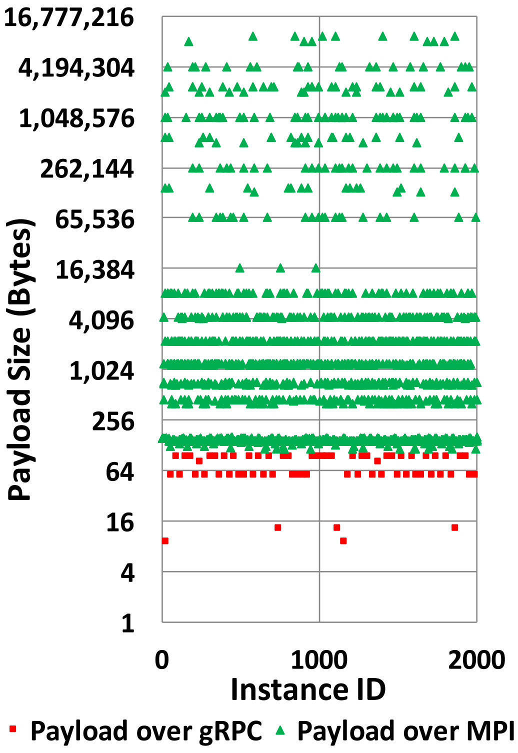
Payload Distribution



gRPC



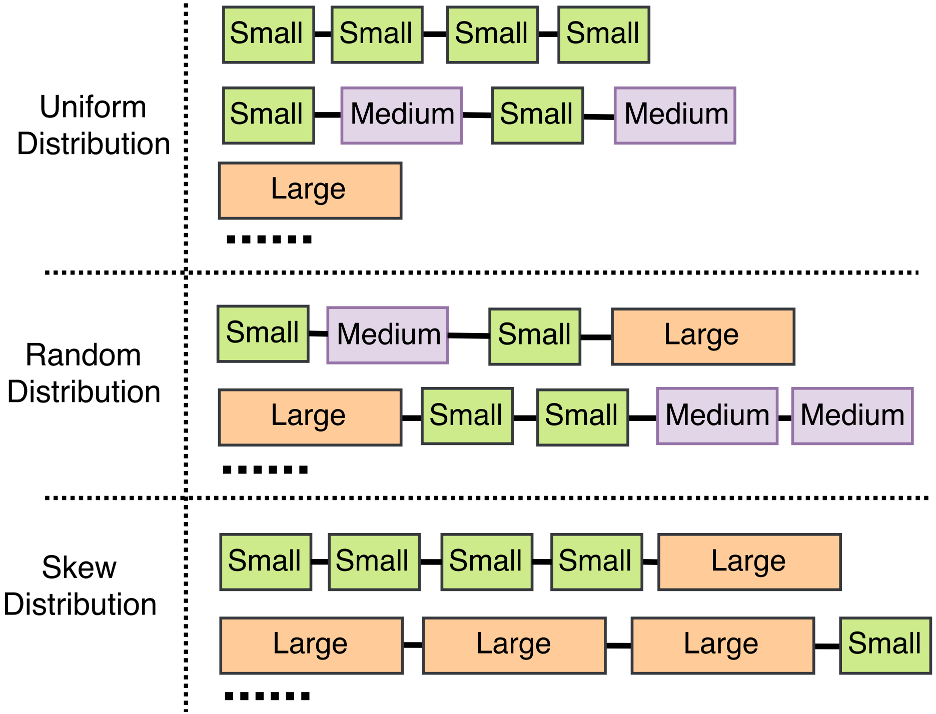
gRPC+Verbs



gRPC+MPI

Payload Distribution

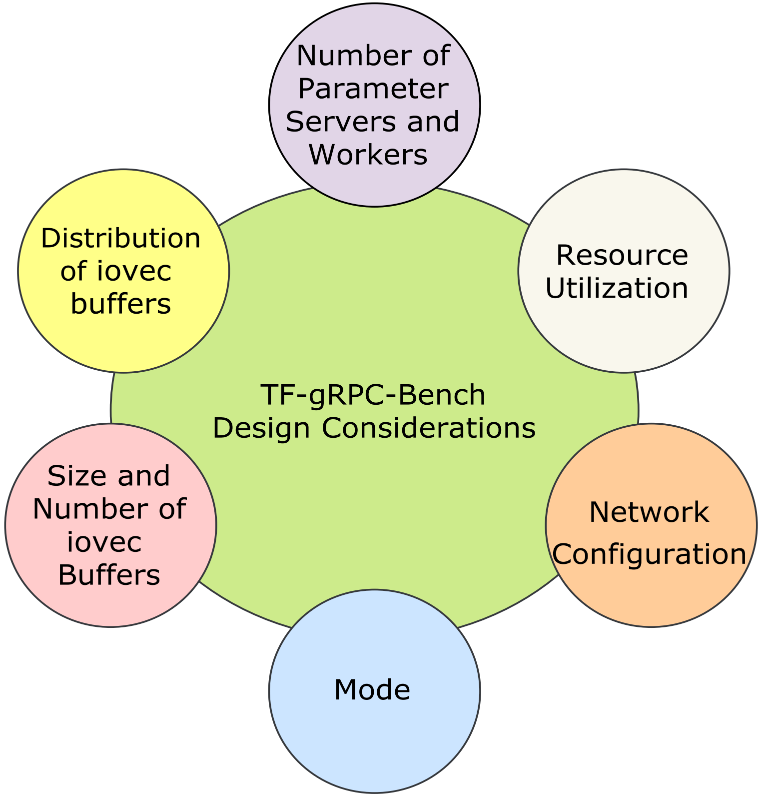
- Profiled different CNNs
- Small, Medium and Large indicate buffers of few Bytes, KBytes and MBytes of length, respectively
- gRPC payload may contain a uniform distribution of such Small buffers
- A lot of Large buffers and a few Small buffers may create a skew distribution of such buffers in one gRPC payload



iovec Buffer Distribution Observed for TensorFlow training over gRPC

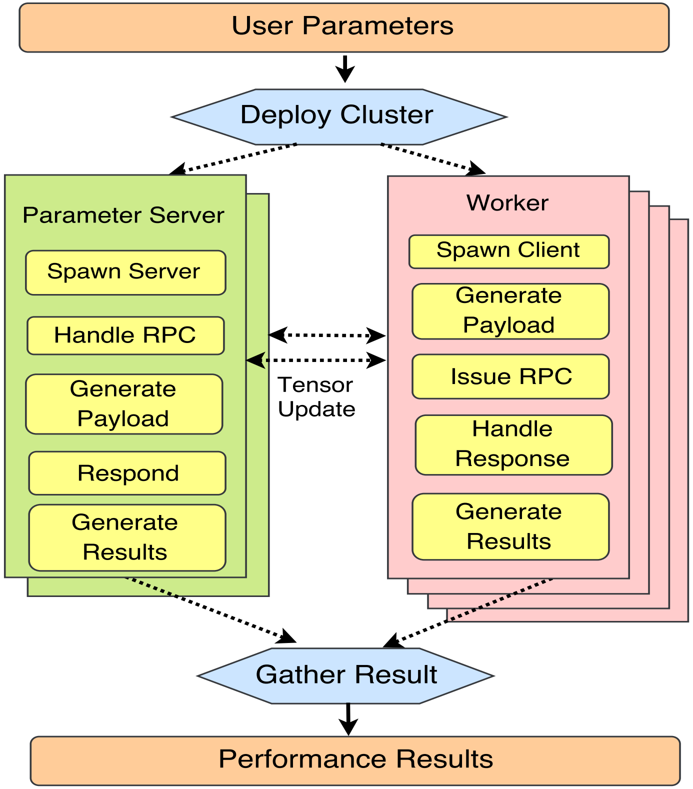
TensorFlow DL Micro-benchmarks for gRPC

Design Considerations for TF-gRPC-Bench Micro-benchmark



R. Biswas, X. Lu, and D. K. Panda, Designing a MicroBenchmark Suite to Evaluate gRPC for TensorFlow: Early Experiences, BPOE-9, 2018.

Design of TF-gRPC-Bench Micro-benchmark Suite



TF-gRPC-Bench Deployment

- Deploys in Parameter Server architecture to exactly model the distributed TensorFlow communication pattern
- Three different benchmarks to measure –
  - Point-to-Point latency
  - Point-to-Point Bandwidth
  - Parameter Server Throughput
- Supports both serialized and non-serialized mode of payload transfer
- Written using gRPC's C++ language binding API's
- Uses gRPC's core C APIs directly to avoid any serialization overhead
- Payload generation Schemes:
  - Uniform
  - Random
  - Skew

Experimental Setup

- We have used two different clusters

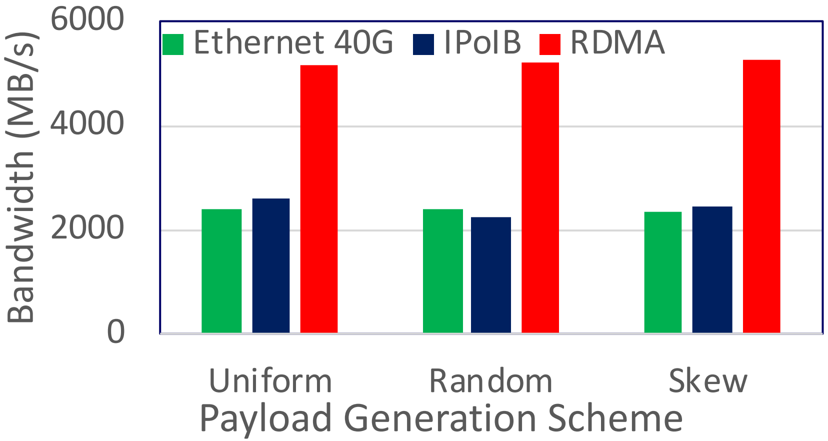
A: OSU-RI2-IB-EDR	B: SDSC-Comet-IB-FDR
❑Intel Broadwell, dual fourteen-core processors	❑Intel Haswell, dual twelve-core processors
❑512 GB RAM	❑128 GB RAM
❑370 GB Local NVMe-SSD	❑320 GB Local SSD
❑InfiniBand EDR	❑InfiniBand FDR

- Software Stack used

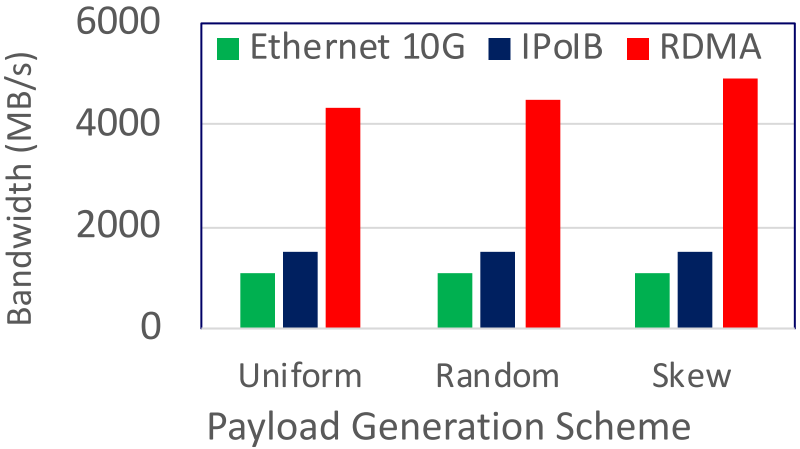
Stack	Version	Cluster
gRPC	1.5.0	A, B
AR-gRPC (OSU RDMA gRPC)[1]	Based on 1.5.0	A, B
TensorFlow	1.4 , Python 2.7	A

1. R. Biswas, X. Lu, and D. K. Panda, Accelerating TensorFlow with Adaptive RDMA-based gRPC. HiPC'18.

TF-gRPC-P2P-Bandwidth



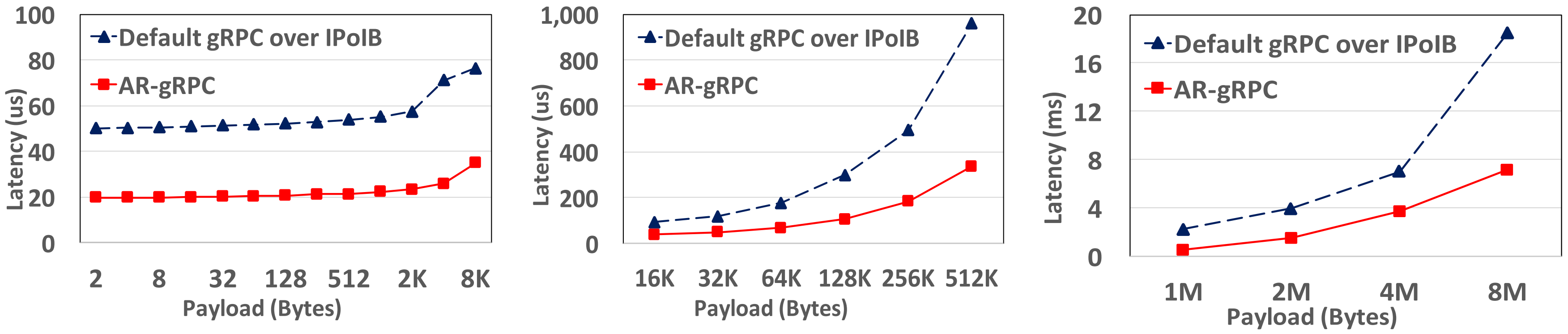
Cluster A



Cluster B

- Cluster A: RDMA gRPC achieves a 2.14x bandwidth increase compared to IPoIB and Ethernet.
- Cluster B: RDMA achieves 3.2x bandwidth compared to IPoIB for skewed data.

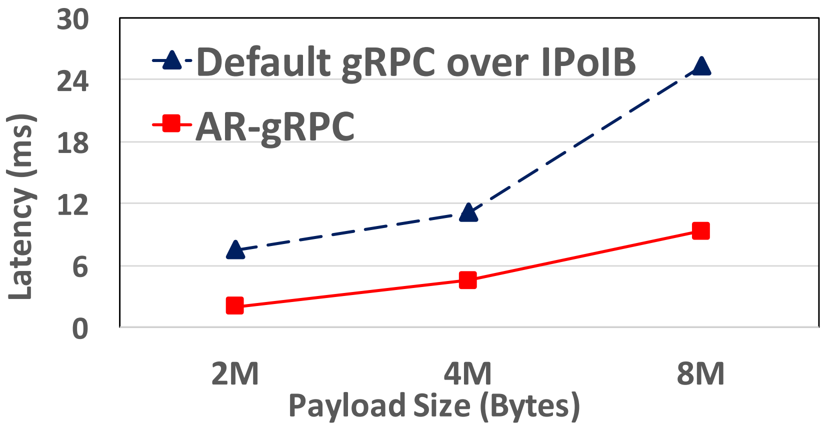
Point-to-Point Latency

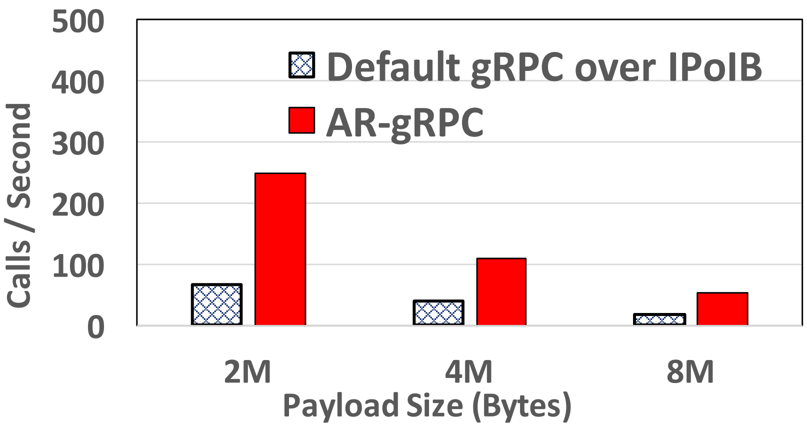


gRPC Point-to-Point Latency Evaluation on Cluster B

- AR-gRPC reduces 32 Bytes latency by 60%
- Shows a speedup of about 2.5x and 4.1x for 64 KBytes and 1 MBytes payload, respectively

Performance Comparison in Fully-Connected Architecture

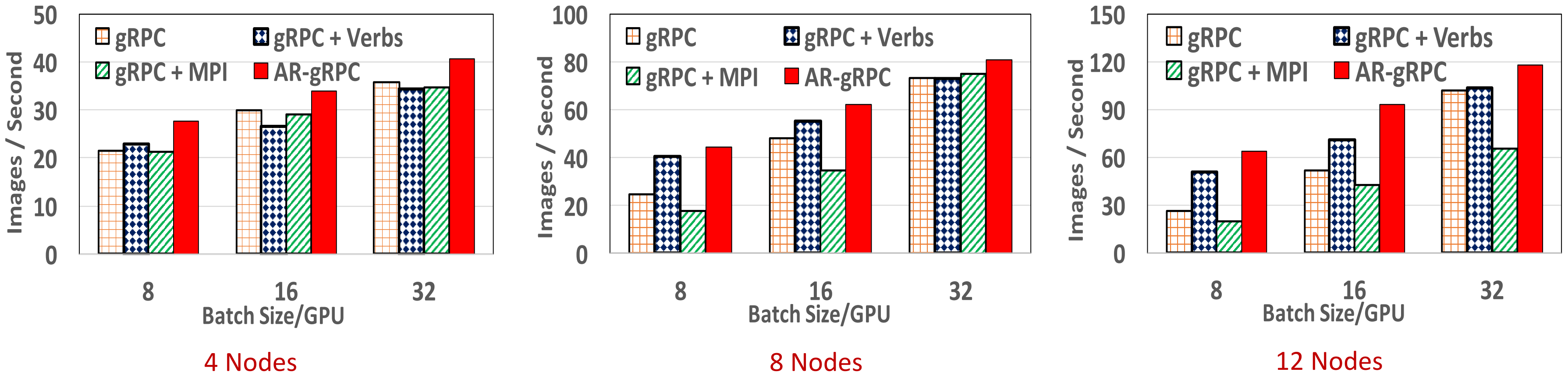




Performance Comparison in Fully-Connected Architecture of gRPC on Cluster B

- AR-gRPC achieves 60% reduction in average latency.
- Obtains throughput speedup of about 2.68x for 4 Mbytes payload.

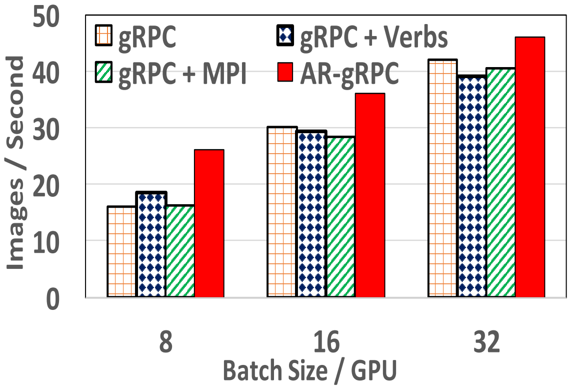
Evaluation of TensorFlow: Inception4



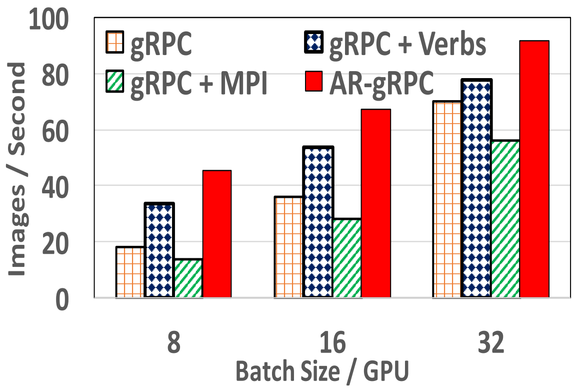
Inception4 Evaluation on Cluster A (Higher Better); *TotalBatchSize* = (*BatchSize/GPU*)×*NUMofGPUs*

- 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

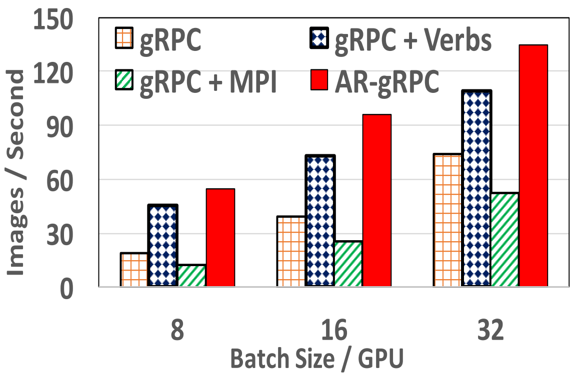
Evaluation of TensorFlow: Resnet152



4 Nodes



8 Nodes



12 Nodes

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

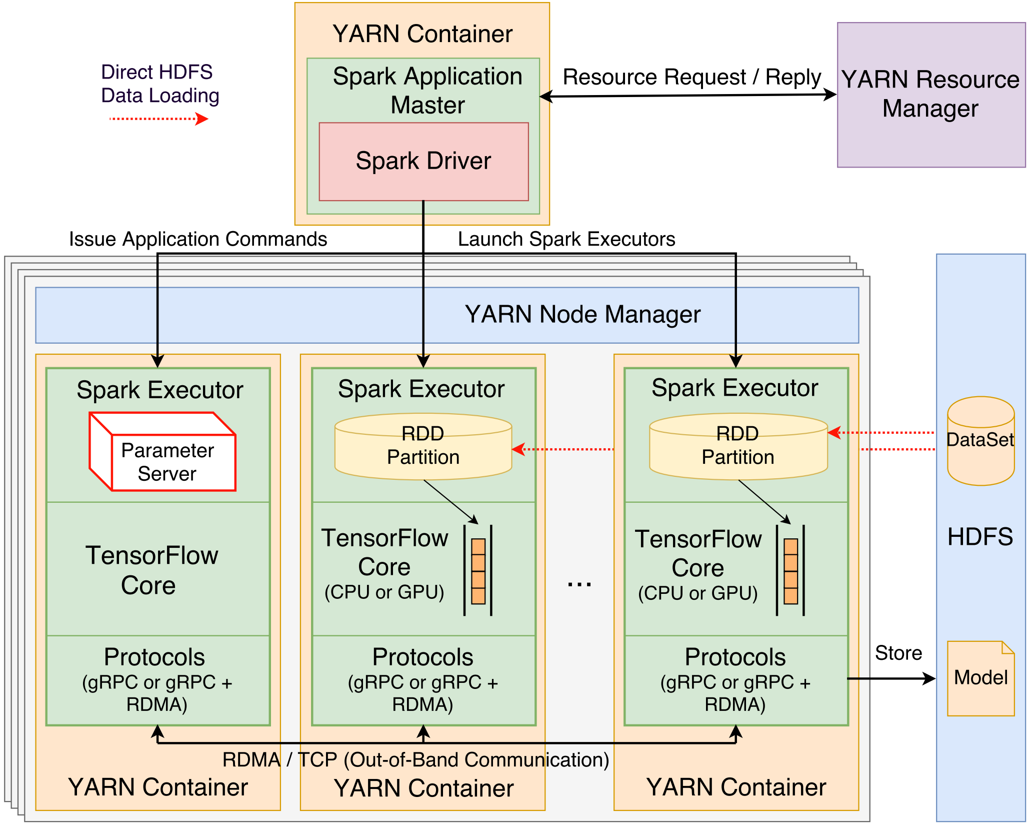
- AR-gRPC accelerates TensorFlow by 62% (batch size 8/GPU) more compared to default gRPC on 4 nodes
- AR-gRPC improves Resnet152 performance by 32% (batch size 32/GPU) to 147% on 8 nodes
- AR-gRPC incurs a maximum speedup of 3x (55 vs 18 images) compared to default gRPC 12 nodes
  - Even for higher batch 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

Case Studies - Characterizing and Benchmarking TensorFlow

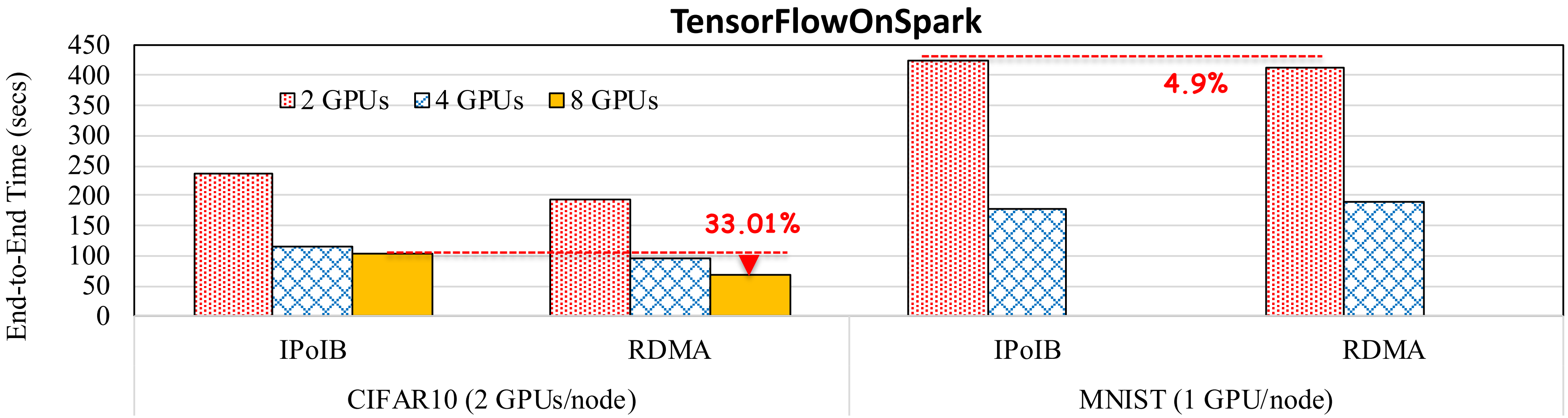
- Standalone TensorFlow
- TensorFlow on Spark

Overview of TensorFlowOnSpark

- Spark Executors acting as containers used to run TensorFlow code
- Two different modes to ingesting data
  – Read data directly from HDFS using built-in TensorFlow modules
  – Feeding data from Spark RDDs to Spark executors (TensorFlow core)
- Scalable and Communication intensive
  – Parameter Server-based approach
  – Embedded inside one Spark executor and talk to other workers over gRPC or gPRC with RDMA
  – Out-of-band communication



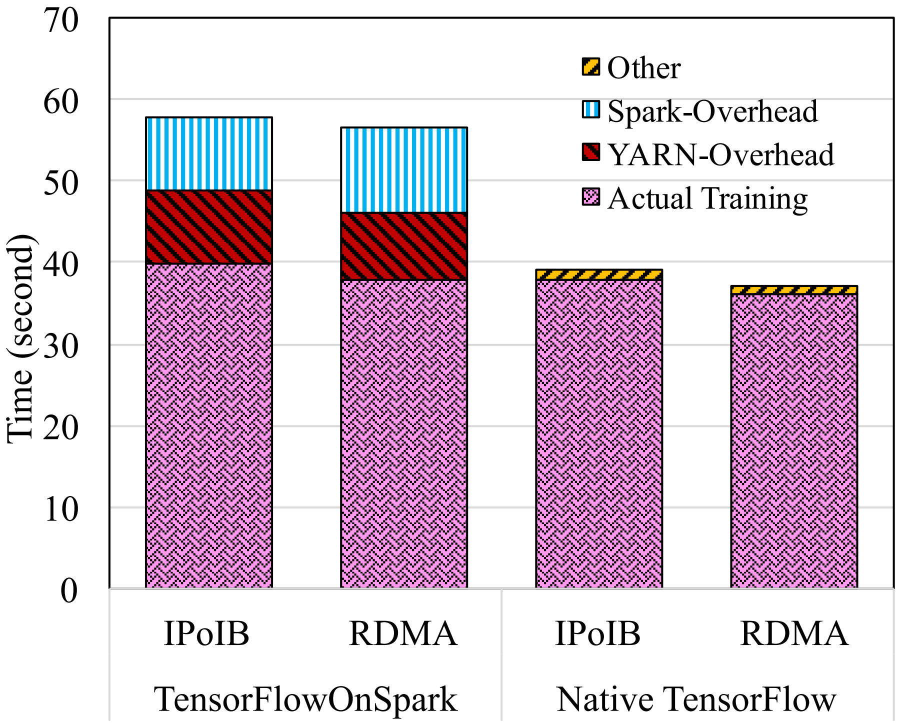
Performance Characterization for IPoIB and RDMA with TensorFlowOnSpark (IB EDR)



- RDMA outperforms IPoIB by **33%** for 8 GPUs in training CIFAR-10 model. However, in training MNIST, RDMA is **4.9%** faster for 2 GPUs and **worse** than IPoIB for 4 GPUs
- The default RDMA design in TensorFlowOnSpark is not fully optimized yet. For MNIST tests, RDMA is not showing obvious benefits

Performance Overhead across Layers in DLoBD Stacks

- SoftMax Regression model, over MNIST dataset
- Up to 15.5% time in Apache Hadoop YARN scheduler layer
- Up to 18.1% execution time in Spark job execution layer
- Data size is small, so we do not count the time spent on accessing HDFS layer.
- Need more effort to reduce the overhead across different layers of DLoBD stacks
- Maybe amortized in long-running deep learning jobs



X. Lu, H. Shi, R. Biswas, M. H. Javed, and D. K. Panda, DLoBD: A Comprehensive Study of Deep Learning over Big Data Stacks on HPC Clusters. TMSCS'18.

Concluding Remarks and Future Work

- Deep Learning community needs system perspective benchmarks to understand the complex executions in deep learning stacks, like TF and TF-on-Spark
- Such benchmarks should also be able to help system design and optimization
- Early experience with TF-gRPC-Bench
  - Measures Point-to-Point latency, Point-to-Point bandwidth, and Parameter Server throughput that models the distributed TensorFlow communication pattern
  - Supports gRPC workload generation that captures the TensorFlow deep learning workload characteristics
- More bottlenecks in DLoBD stacks and lack of benchmarking tools
- Future Work
  - Designing more generic and highly optimized DL Stacks and Benchmark Suites

Q & A
Thank You!