Parameter Hub

A Rack-Scale Parameter Server for Efficient Cloud-based Distributed Deep Neural Network Training

Liang Luo, Jacob Nelson, Luis Ceze, Amar Phanishayee and Arvind Krishnamurthy
- DNN training is computationally expensive
- Needs to train it in distributed fashion
- People use cloud for DDNN training

Major cloud providers all have an ecosystem for cloud-based DDNN training.
Distributed Training

INDEPENDENT FORWARD/BACKWARD PASSES + COORDINATED PARAMETER EXCHANGE
Distributed Training

INDEPENDENT FORWARD/BACKWARD PASSES + COORDINATED PARAMETER EXCHANGE

Time

Parameter Server

Worker 1

F1 → B1 → F2 → B2 → F3 → B2

Worker 2

F1 → B1 → F2 → B2 → F3 → B2

Worker

Parameter Server

(F)orward Pass (B)ackward Pass (A)ggregation (O)ptimization
Cloud-based Distributed Training Today
IN THE CONTEXT OF THE CLOUD
Cloud-based Distributed Training Today
FORWARD AND BACKWARD PASSES IN WORKER
Cloud-based Distributed Training Today

AGGREGATION AND OPTIMIZATION IN PS

ToR
Worker 1
PS 1

Network Core

ToR
PS 2
Worker 2
DDNN training is communication bound

- Problem gets worse over time: shifting bottleneck.
- With modern GPUs most of the time is spent on communication.
- Making GPUs faster will do little to increase throughput.
- Wasting compute resources.
DDNN training is communication bound

- AlexNet
- Inception V3
- ResNet 269
- GoogleNet
Bottlenecks in Cloud-based DDNN training

MAPPING OF TRAINING WORKLOAD TO THE CLOUD IS INEFFICIENT.
Bottlenecks in Cloud-based DDNN training

FRAMEWORK BOTTLENECKS

Worker 1

Training Framework

GPU

Network
Bottlenecks in Cloud-based DDNN training

FRAMEWORK BOTTLENECKS

- ResNet 269
- Inception
- GoogleNet
- AlexNet

- Compute
- Data Copy and Communication
- Aggregator
- Optimizer
- Synchronization and other Overheads

Seconds
0
2
4
6
8
10
12
14
16
Bottlenecks in Cloud-based DDNN training

MAPPING OF TRAINING WORKLOAD TO THE CLOUD IS INEFFICIENT.
Bottlenecks in Cloud-based DDNN training
BANDWIDTH BOTTLENECK
Bottlenecks in Cloud-based DDNN training

INSUFFICIENT BANDWIDTH

Minimum bandwidth required for each of the popular NNs for communication to not bottleneck computation?

8 workers, GTX 1080 Ti, central parameter servers. MxNet

- GoogleNet / Inception: 40 Gbps
- ResNet: 100 Gbps
- AlexNet: 1200 Gbps
- Cloud Bandwidth

10 Gbps
25 Gbps
1000 Gbps
1300 Gbps
Bottlenecks in Cloud-based DDNN training
MAPPING OF TRAINING WORKLOAD TO THE CLOUD IS INEFFICIENT.
Bottlenecks in Cloud-based DDNN training

DEPLOYMENT-RELATED OVERHEAD
Bottlenecks in Cloud-based DDNN training

DEPLOYMENT-RELATED OVERHEAD

- Transient congestion, or oversubscription by design
- Cross-rack communication cost is higher than Intra-rack communication.
- Comm. bottlenecked by slowest link.
Parameter Hub Optimizations
CODESIGNING SOFTWARE, HARDWARE AND CLUSTER CONFIGURATION FOR EFFICIENT CLOUD-BASED DDNN TRAINING
Eliminating framework bottlenecks:

PHub Optimizations: streamlining DDNN training pipeline
Eliminating framework bottlenecks:

PHub Optimizations: streamlining DDNN training pipeline
Software Optimizations
Software Optimizations

GRADIENT AGGREGATION AND OPTIMIZATION

Each core reads the input Q from different workers and writes to different locations to the output queue.

For each input Q, launch a series of threads for aggregation. This is used in MxNet. (Wide Aggregation)

Sequentially aggregates the same portion of gradients within each queue. (Tall Aggregation)

Organize processors into hierarchy. Perform NUMA aware tree reduction.

Requires synchronization.

Great locality. No synchronization.

Great locality. No synchronization.

Too much coherence and synchronization.
Software Optimizations
TALL AGGREGATION AND OPTIMIZATION

- Chunk a gradient into a series of virtual gradients deterministically.
- A virtual gradient is mapped to a particular core on the server.
- Virtual gradients are transferred independently.
- A chunk is only processed by a single core: maintaining maximum locality.

Gradient Array for Key 0 from 8 workers

Core Mappings

Aggregated
Software Optimizations
TALL AGGREGATION AND OPTIMIZATION

When Aggregation is done, PHub:
- PHub optimizes a chunk with the same core that aggregates that chunk.

Key 0 from 8 workers
When Aggregation is done, PHub:
- PHub optimizes a chunk with the same core that aggregates that chunk.
- Allows overlapping of aggregation, optimization and gradient transmission.
Software Optimizations

NOT ENOUGH ON THEIR OWN!

Typical server configuration is unbalanced
Eliminating bandwidth bottlenecks:
PBox hardware: balanced computation and communication resources.
Eliminating bandwidth bottlenecks:
PBox hardware: balanced computation and communication resources.
Hardware Optimization

THE PBOX

• Balanced computation and communication
• Extends the balance and locality notion across NUMA domains and NICs.
Hardware Optimization

THE PBOX

- Balanced computation and communication
- Extends the balance and locality notion across NUMA domains and NICs.
Hardware Optimization

THE PBOX

- Balanced computation and communication
- Extends the balance and locality notion across NUMA domains and NICs.
Eliminating deployment bottlenecks:

PHub hierarchical reduction: reducing cross rack traffic
Eliminating deployment bottlenecks:

PHub hierarchical reduction: reducing cross rack traffic
PBox Deployment

RACK SCALE PARAMETER SERVICE

Cluster
Network

ToR

Worker/PS 1

Worker/PS N

CM

ToR

Worker/PS 1

Worker/PS 2
PBox Deployment

RACK SCALE PARAMETER SERVICE
Two-Phase Hierarchical Aggregation

ADAPTING TO THE DATACENTER NETWORK TOPOLOGY

Cluster Network

- N times traffic reduction!

1. Intra-Rack central aggregation

2. Inter-Rack aggregation
Up to 2.7x performance in 10Gbps cloud-like environment

8 Workers. GTX 1080 Ti. MxNet: InfiniBand-enhanced baseline. PBox. Batch Size 64 for ResNext, 128 for ResNet 269, 256 for all others.
Framework Bottlenecks

- Data Copy
- Aggregation and Optimization
- Synchronization

![Graph showing performance of different models](image-url)
Scalability
LINEAR SCALING IN COMM. ONLY BENCHMARK

Memory Bandwidth (MB/s)

Number of active workers

PHub training
Scalability

PCI-E TO MEMORY SUBSYSTEM BRIDGE

120 Machines training ResNet 50

Microbenchmark limit

PHub training

Memory Bandwidth (MB/s)

Number of active workers
Scalability Beyond a Single Rack

EMULATING HIERARCHICAL AGGREGATION

Overhead of Phub cross-rack synchronization

<table>
<thead>
<tr>
<th>Number of racks</th>
<th>Overhead (%)</th>
<th>Overhead (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>14.3%</td>
<td>0.4%</td>
</tr>
<tr>
<td>3</td>
<td>14.7%</td>
<td>1.6%</td>
</tr>
<tr>
<td>4</td>
<td>15.1%</td>
<td>1.8%</td>
</tr>
<tr>
<td>5</td>
<td>15.5%</td>
<td>1.8%</td>
</tr>
<tr>
<td>6</td>
<td>15.9%</td>
<td>1.4%</td>
</tr>
<tr>
<td>7</td>
<td>16.2%</td>
<td>1.5%</td>
</tr>
<tr>
<td>8</td>
<td>16.5%</td>
<td>1.9%</td>
</tr>
</tbody>
</table>

AlexNet

ResNet 50
Cost Analysis – for infrastructure builders

25% BETTER THROUGHPUT/$

Accounting for network devices (switch ports, network adapters, network cables), GPU costs, and PBox’s entire machine cost.

Core oversubscription 2:1
Parameter Hub

A software, hardware and cluster configuration codesign that target three major bottlenecks in the cloud for more efficient DDNN training