Scalable Distributed Training with Parameter Hub: a whirlwind tour
TVM Stack

High-Level Differentiable IR

Tensor Expression IR

LLVM, CUDA, Metal  VTA

Optimization

AutoTVM  AutoVTA

Hardware Fleet

Edge FPGA  Cloud FPGA  ASIC
Groundwork for bringing TVM to the distributed world for training and inference, on commercial cloud, or in your own cluster.
Parameter Hub

Optimized, topology-aware and dynamic mechanism for inter-machine communication
Parameter Hub

Optimized, topology-aware and dynamic mechanism for inter-machine communication

* In the cloud-based training context
Deep Learning constitutes an important workload in cloud today.

Major cloud providers all have an ecosystem for cloud learning.
Deep Learning constitutes an important workload in cloud today.

Major cloud providers all have an ecosystem for cloud learning.
Server demand for DL inference across data centers nearly quadrupled in less than 2 years. Source: Facebook
Server demand for DL inference across data centers nearly quadrupled in less than 2 years. Source: Facebook
EC2 reclaims your GPU instances as they run out of capacity.
<table>
<thead>
<tr>
<th>Instance ID</th>
<th>Instance Type</th>
<th>State</th>
<th>ID</th>
<th>Reason</th>
<th>Duration</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-0a9e784f</td>
<td>g3.xlarge</td>
<td>closed</td>
<td>i-09545e7b74364847</td>
<td>instance-terminated-no-capacity</td>
<td>one-time</td>
<td>$2.28</td>
</tr>
<tr>
<td>i-0a9e784f</td>
<td>g3.xlarge</td>
<td>closed</td>
<td>i-09545e7b74364847</td>
<td>instance-terminated-no-capacity</td>
<td>one-time</td>
<td>$2.28</td>
</tr>
<tr>
<td>i-0a9e784f</td>
<td>g3.xlarge</td>
<td>closed</td>
<td>i-09545e7b74364847</td>
<td>instance-terminated-no-capacity</td>
<td>one-time</td>
<td>$2.28</td>
</tr>
</tbody>
</table>

EC2 reclaims your GPU instances as they run out of capacity.
Distributed Training

INDEPENDENT FORWARD/BACKWARD PASSES + COORDINATED PARAMETER EXCHANGE

Time
Parameter Server

Worker 1
Worker 2

F1 → B1
F1 → B1

F1 → B1
F1 → B1

F2 → B2
F2 → B2

F2 → B2
F2 → B2

A1 → O1
A2 → O2

A1 → O1
A2 → O2

(F) orward Pass (B) ackward Pass (A) ggregation (O) ptimization

Worker Parameter Server
Distributed Training
INDEPENDENT FORWARD/BACKWARD PASSES +
COORDINATED PARAMETER EXCHANGE
Distributed Training Today
IN THE CONTEXT OF THE CLOUD

Network Core

ToR
Machine with GPUs
Machine

ToR
Machine with GPUs
Machine
Distributed Training Today
FORWARD AND BACKWARD PASSES IN WORKER

Network Core

ToR
Worker 1
PS 1

ToR
PS 2
Worker 2
Distributed Training Today
AGGREGATION AND OPTIMIZATION IN PS

Network Core

ToR
Worker 1
PS 1

ToR
PS 2
Worker 2
Distributed 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.
Distributed training is communication bound

AlexNet
Inception V3
ResNet 269
GoogleNet
Bottlenecks in DDNN training

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

FRAMEWORK BOTTLENECKS
Bottlenecks in DDNN training

FRAMEWORK BOTTLENECKS

Training Framework

GPU

Network

Worker 1
Bottlenecks in DDNN training

FRAMEWORK BOTTLENECKS
Bottlenecks in DDNN training

FRAMEWORK BOTTLENECKS

<table>
<thead>
<tr>
<th>Framework</th>
<th>Compute</th>
<th>Data Copy and Communication</th>
<th>Aggregator</th>
<th>Optimizer</th>
<th>Synchronization and other Overheads</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet 269</td>
<td>0.4</td>
<td>0.8</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
</tr>
<tr>
<td>Inception</td>
<td>0.4</td>
<td>0.8</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
</tr>
<tr>
<td>GoogleNet</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
</tr>
<tr>
<td>AlexNet</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
</tr>
</tbody>
</table>
Bottlenecks in DDNN training

MAPPING OF TRAINING WORKLOAD TO THE CLOUD IS INEFFICIENT.
Bottlenecks in 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
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

Cloud Bandwidth

- 1300 Gbps
- 1000 Gbps
- 25 Gbps
- 10 Gbps
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

Cloud Bandwidth

1300 Gbps

1000 Gbps

25 Gbps

10 Gbps
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
- Cloud Bandwidth:
  - 1300 Gbps
  - 1000 Gbps
  - 25 Gbps
  - 10 Gbps

Cloud Bandwidth
Bottlenecks in Cloud-based DDNN training

INSUFFICIENT BANDWIDTH

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

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

8 workers, GTX 1080 Ti, central parameter servers. MxNet
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.
Parameter Hub Optimizations
CODESIGNING SOFTWARE, HARDWARE WITH 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

Network Core

ToR Worker 1

PS 1

ToR PS 2

Worker 2

GRADIENTS

MEMORY

CPU

Software Optimizations
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

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.
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.
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.

Gradient Array for Key 0 from 8 workers
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.
When Aggregation is done, PHub:
- PHub optimizes a chunk with the same core that aggregates that chunk.
Software Optimizations
TALL AGGREGATION AND OPTIMIZATION

When Aggregation is done, PHub:
- PHub optimizes a chunk with the same core that aggregates that chunk.
- FP32-level streaming aggregation and optimization to hide communication latency.
Software Optimizations
TALL AGGREGATION AND OPTIMIZATION

When Aggregation is done, PHub:
- PHub optimizes a chunk with the same core that aggregates that chunk.
- FP32-level streaming aggregation and optimization to hide communication latency.
Eliminating deployment bottlenecks:

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

PHub hierarchical reduction: reducing cross rack traffic
Two-Phase Hierarchical Aggregation

RACK SCALE PARAMETER SERVICE
Two-Phase Hierarchical Aggregation

RACK SCALE PARAMETER SERVICE

Cluster Network

CM

ToR

Worker/PS 1

Worker/PS N

Rack

ToR

PBox

Worker/PS 1

Worker/PS 2
Two-Phase Hierarchical Aggregation

ADAPTING TO THE DATACENTER NETWORK TOPOLOGY
Two-Phase Hierarchical Aggregation
ADAPTING TO THE DATACENTER NETWORK TOPOLOGY
Two-Phase Hierarchical Aggregation
ADAPTING TO THE DATACENTER NETWORK TOPOLOGY

1. Intra-Rack central aggregation
2. Inter-Rack aggregation

N times traffic reduction!
Efficient DDNN Training in Commercial Cloud

ACTIVE TOPOLOGY PROBING

VMs
Azure/EC2
Efficient DDNN Training in Commercial Cloud

ACTIVE TOPOLOGY PROBING

DPDK-based latency Probe

VMs
Azure/EC2
Efficient DDNN Training in Commercial Cloud
ACTIVE TOPOLOGY PROBING

DPDK-based latency Probe → Distance Matrix

VMs
Azure/EC2
Efficient DDNN Training in Commercial Cloud

ACTIVE TOPOLOGY PROBING

- DPDK-based latency Probe
- Distance Matrix
- Clustering Algorithms

VMs
Azure/EC2
Efficient DDNN Training in Commercial Cloud

ACTIVE TOPOLOGY PROBING

VMs
Azure/EC2

DPDK-based latency Probe

\{\ldots\}\n
Distance Matrix

Clustering Algorithms

Inferred Network Topology*

* Inferred Network Topology
Efficient DDNN Training in Commercial Cloud

ACTIVE TOPOLOGY PROBING

VMs
Azure/EC2

DPDK-based latency Probe

{ Distance Matrix }

Clustering Algorithms

Inferred Network Topology*

Automagic Schedule Generation
Efficient DDNN Training in Commercial Cloud
ACTIVE TOPOLOGY PROBING

VMs Azure/EC2

DPDK-based latency Probe

Distance Matrix

Clustering Algorithms

Inferred Network Topology*

Automagic Schedule Generation

Hierarchical Reduction Plan
Efficient DDNN Training in Commercial Cloud

ACTIVE TOPOLOGY PROBING
Performance in commercial cloud with PHub

Framework Integration

Support for Mxnet/Pytorch/Caffe2.

```cpp
var pHub = std::make_shared<PHub>(cfg.redisIp, nMap, keySize, appAddrs, cntr,
sizeof(float), cfg.rank, plp);
pHub->ToggleUseSchedule(pSchedule);
pHub->Reduce();
```
Groundwork for bringing TVM to the distributed world for training and inference, on commercial cloud, or in your own cluster.
Hardware Parameter Hub
Hardware Parameter Hub

Balanced computation and communication resource.

- 10 ConnectX-3 Card
- 560+Gbps Network BW
- 800Gbps PCIe
- Fully supported by Software Parameter Hub
Hardware Parameter Hub

35GB/s aggregation throughput. Supports 100+ ResNet-50 training nodes with a single machine.

<table>
<thead>
<tr>
<th>Gloo HD</th>
<th>Gloo Ring</th>
<th>PS-Lite</th>
<th>PHub SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>5</td>
<td>4.5</td>
<td>2</td>
</tr>
</tbody>
</table>
Hardware Parameter Hub
ResNet-50.
See paper for detailed estimates.

Better training throughput/\$. 
Hardware Parameter Hub

ResNet-50.
See paper for detailed estimates.

25% Better training throughput/$. 