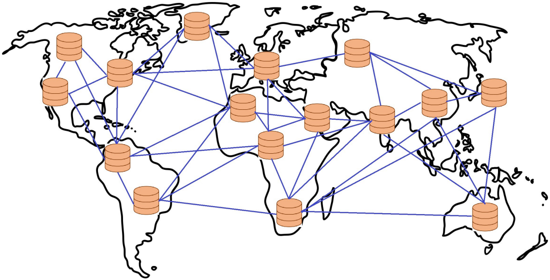
WeightGrad: Geo-Distributed Data Analysis Using Quantization for Faster Convergence and Better Accuracy

Syeda Nahida Akter

Muhammad Abdullah Adnan

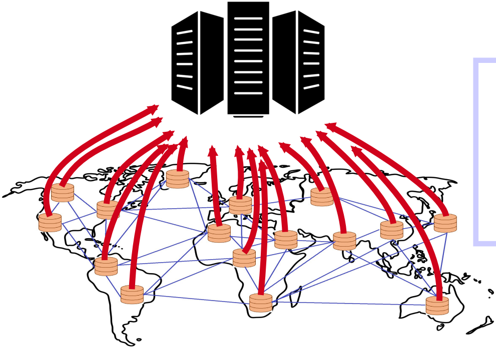
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)
Dhaka, Bangladesh

Problem Overview



- Large scale cloud organizations are establishing data centers and "edge" clusters globally to provide their users low latency access to their services.

Problem Overview

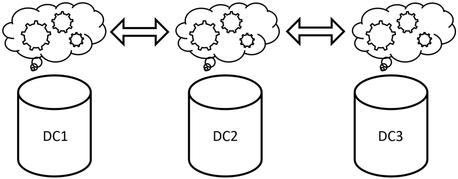


Problem

- Powerful machines.
- Huge memory.
- Large amount of time.

Problem Overview

Clearly, **centralization** is not a feasible solution which motivates the need to **distribute the DNN** system across multiple data centers.



Problem Definition

For distributed setup, we define the problem as

Problem Definition

For distributed setup, we define the problem as

- How to efficiently utilize limited WAN b/w

Problem Definition

For distributed setup, we define the problem as

- How to efficiently utilize limited WAN b/w
- How to ensure faster convergence without loss of accuracy

Methodology

We propose **WeightGrad** that

Methodology

We propose **WeightGrad** that

- adapts both **weight and gradient quantization** to provide best speedup possible on WAN

Methodology

We propose **WeightGrad** that

- adapts both **weight and gradient quantization** to provide best speedup possible on WAN
- proposes a **synchronous structure** to prevent the loss in accuracy due to quantization

WeightGrad System

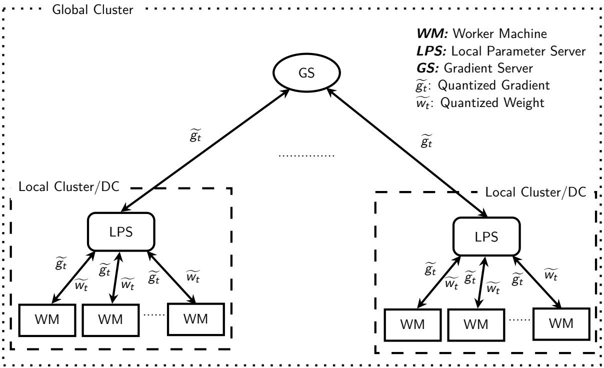


Figure: WeightGrad Tree Structure

Two Level Structure

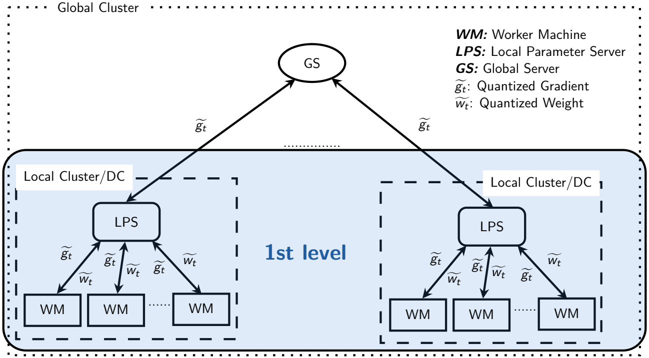


Figure: WeightGrad Tree Structure

Two Level Structure

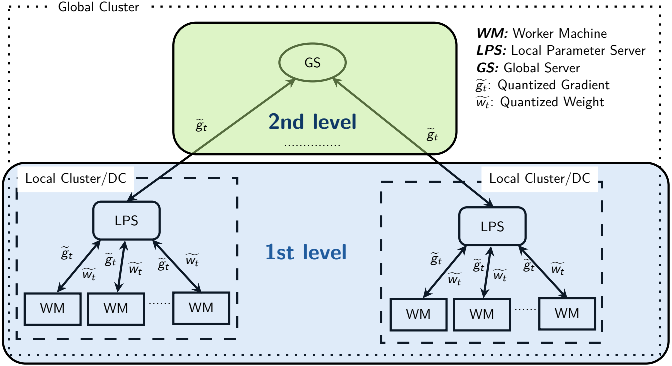


Figure: WeightGrad Tree Structure

Local Cluster

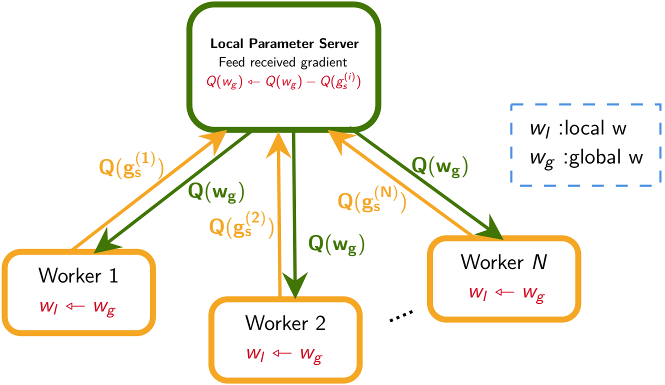


Figure: WeightGrad: Local Cluster

Local Cluster

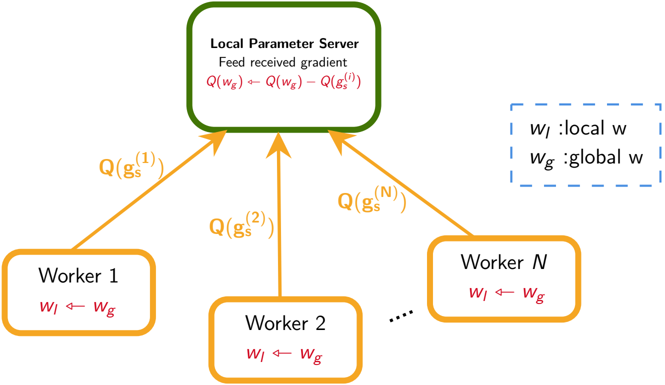


Figure: WeightGrad: Local Cluster

Local Cluster

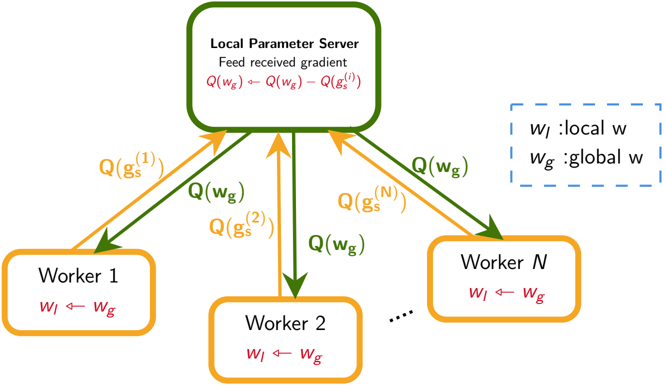


Figure: WeightGrad: Local Cluster

LPS with Gradient Synchronizer

LPS with Gradient Synchronizer

Dynamic Threshold

- Gradient Synchronizer maintains a dynamic threshold for distributing the gradient updates.

LPS with Gradient Synchronizer

Dynamic Threshold

- Gradient Synchronizer maintains a dynamic threshold for distributing the gradient updates.

Fixed Interval

- To synchronize the communication process, Gradient Synchronizer maintains a fixed interval T, within which it receives aggregated gradient values from the GS.
- If an LPS does not get update from the GS within T, it stops sending updates to the WMs until gradient updates are received from the GS.

LPS with Gradient Synchronizer

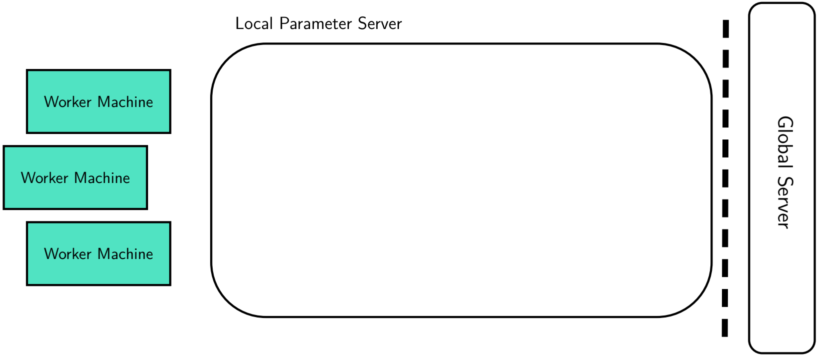


Figure: LPS Structure

LPS with Gradient Synchronizer

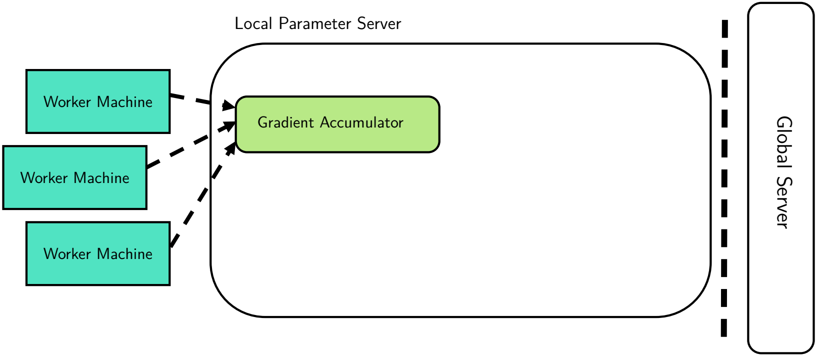


Figure: LPS Structure

LPS with Gradient Synchronizer

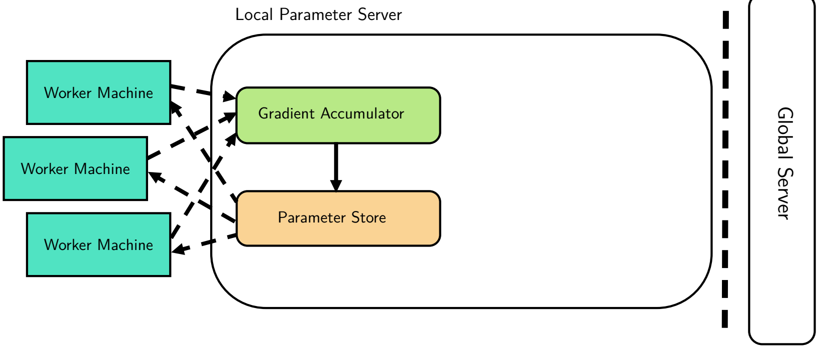


Figure: LPS Structure

LPS with Gradient Synchronizer

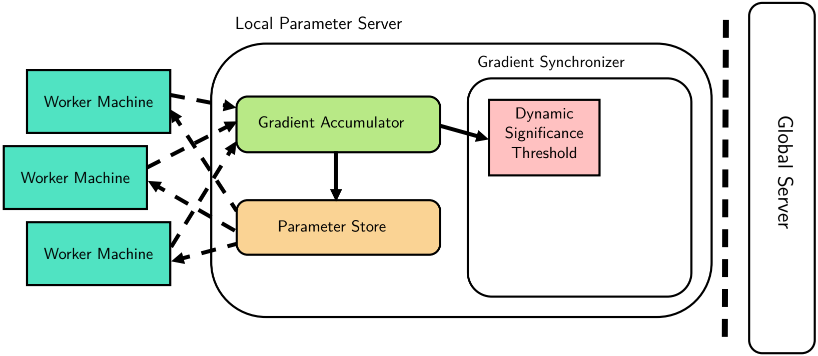


Figure: LPS Structure

LPS with Gradient Synchronizer

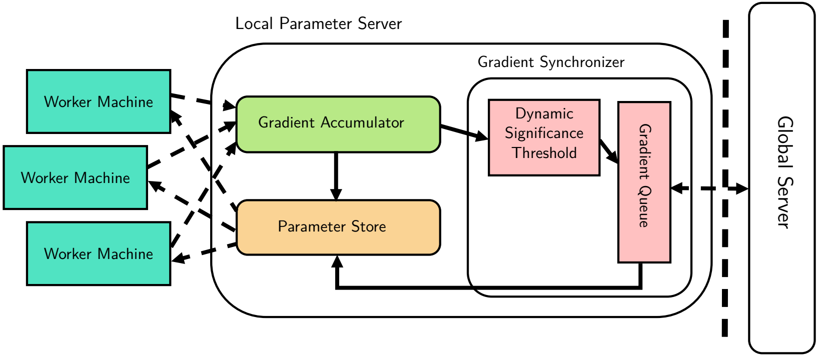


Figure: LPS Structure

LPS with Gradient Synchronizer

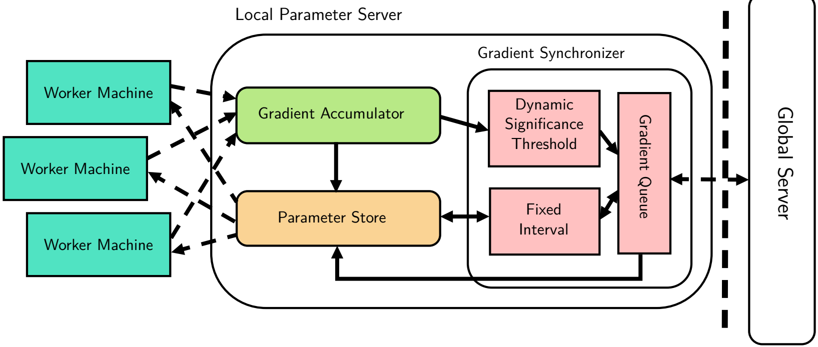
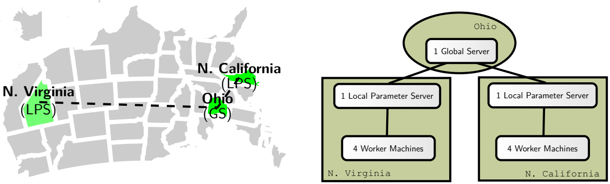


Figure: LPS Structure

Amazon EC-2

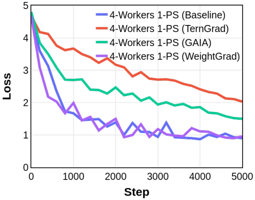


(a) (b)

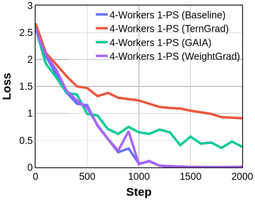
Figure: (a)Deployment Regions in AWS(b)Instance Hierarchy

Instances	Instance Type	RAM	vCPU	GPU	B/W
11	g3s.xlarge, 64-bit Ubuntu Server 16.04 LTS	30.5 GiB	4	NVIDIA Tesla M60 GPU	10 Gbps

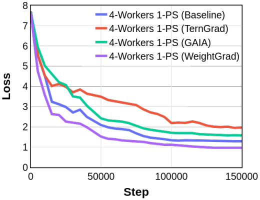
Training Loss Analysis



(a) Training Loss for CifarNet



(b) Training Loss for VGGNet



(c) Training Loss for ImageNet

Figure: (a) Training loss for CifarNet model on CIFAR-10 dataset, (b) Training loss for VGGNet model on CIFAR-10 dataset, (c) Training loss for AlexNet on ImageNet dataset

SpeedUp Analysis

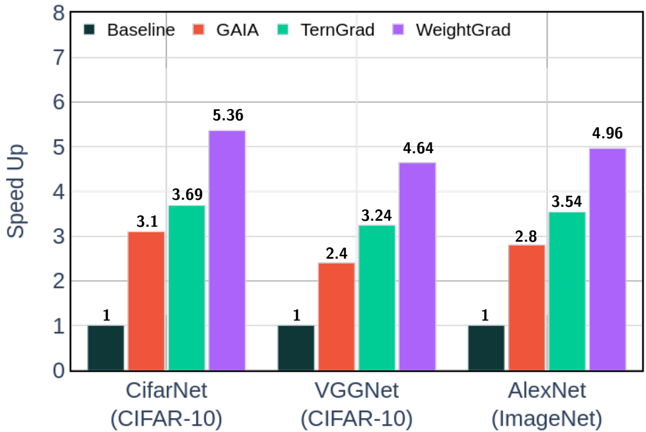


Figure: Training Speed Comparison

SpeedUp Analysis

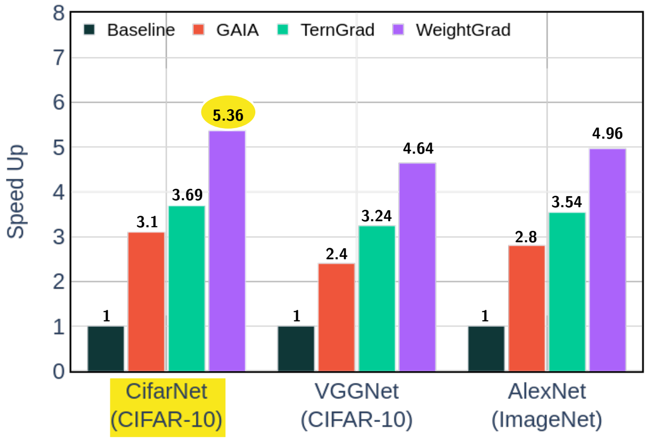


Figure: Training Speed Comparison

SpeedUp Analysis

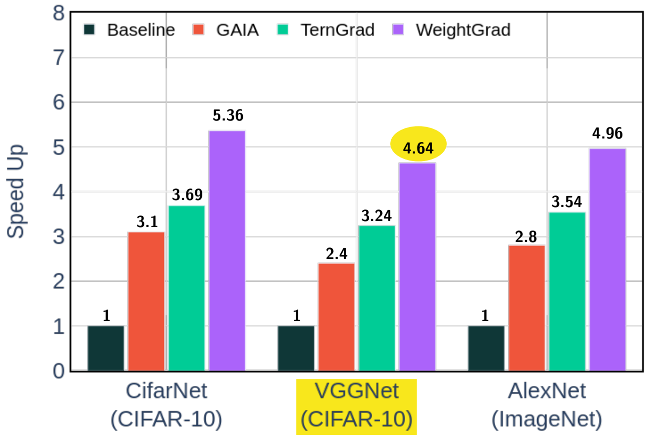


Figure: Training Speed Comparison

SpeedUp Analysis

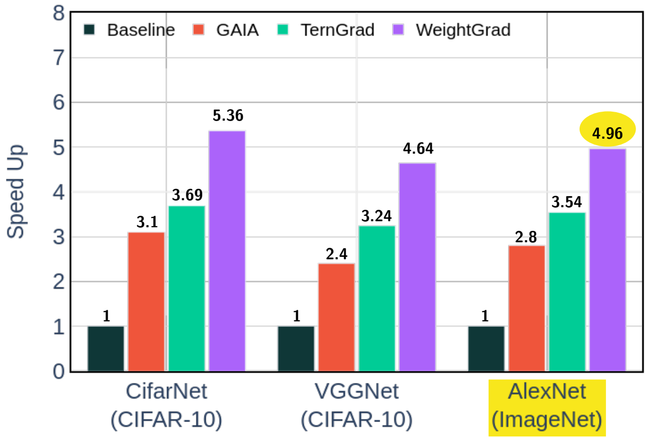


Figure: Training Speed Comparison

Accuracy Comparison: CIFAR-10

Model	SGD	Base LR	Total mini-batch size	Steps	Gradients	Workers	Accuracy
CifarNet	GD	0.1	128	50k	Baseline	4	84.56%
					Gaia	4	83.48%(-1.08%)
					TernGrad	4	82.41%(-2.15%)
					WeightGrad	**4**	**84.56%(-0.00%)**
	GD	0.1	512	50k	Baseline	8	83.19%
					Gaia	8	83.04%(-0.13%)
					TernGrad	8	81.40%(-1.79%)
					WeightGrad	**8**	**83.21%(+0.03%)**
VGG-Net	GD	0.1	512	50k	Baseline	8	88.14%
					Gaia	8	87.19%(-0.95%)
					TernGrad	8	86.3%(-1.84%)
					WeightGrad	**8**	**88.13%(-0.01%)**

Table: Result of WeightGrad on CIFAR-10 dataset

Accuracy Comparison: ImageNet

Model	Steps	Training Method	Top-1 Accuracy	Top-5 Accuracy
AlexNet	185k	Baseline	58.17%	80.19%
		Gaia	58.02%(-0.15%)	80.20%(+0.01%)
		TernGrad	57.32%(-0.85%)	80.18%(-0.01%)
		Deep Gradient Compression	58.20%(+0.03%)	80.20%(+0.01%)
		WeightGrad	**59.28%(+1.06%)**	**80.25%(+0.06)**

Table: Comparison of training methods on ImageNet data

Conclusions

Conclusions

- **Problem**: How to perform DNN on geo-distributed data?

Conclusions

- **Problem**: How to perform DNN on geo-distributed data?
  - Limited WAN bandwidth and slower convergence rate
  - DNN models contain billions of parameters

Conclusions

- **Problem**: How to perform DNN on geo-distributed data?
  - Limited WAN bandwidth and slower convergence rate
  - DNN models contain billions of parameters
- We introduce **WeightGrad**,
  - **A loss-aware weight-quantized deep learning system with quantized gradient**

Conclusions

- **Problem**: How to perform DNN on geo-distributed data?
  - Limited WAN bandwidth and slower convergence rate
  - DNN models contain billions of parameters
- We introduce **WeightGrad**,
  - **A loss-aware weight-quantized deep learning system with quantized gradient**
  - Maintains a **tight synchronization over LAN** to simulate each data center as a centralized system and a **loose synchronization over WAN** to reduce communication cost.

Conclusions

- **Problem**: How to perform DNN on geo-distributed data?
  - Limited WAN bandwidth and slower convergence rate
  - DNN models contain billions of parameters
- We introduce **WeightGrad**,
  - **A loss-aware weight-quantized deep learning system with quantized gradient**
  - Maintains a **tight synchronization over LAN** to simulate each data center as a centralized system and a **loose synchronization over WAN** to reduce communication cost.
- **Results:** WeightGrad achieves,
  - **2.4-5.36$\times$ speedup** over state-of-the-art distributed systems

Conclusions

- **Problem**: How to perform DNN on geo-distributed data?
  - Limited WAN bandwidth and slower convergence rate
  - DNN models contain billions of parameters
- We introduce **WeightGrad**,
  - **A loss-aware weight-quantized deep learning system with quantized gradient**
  - Maintains a **tight synchronization over LAN** to simulate each data center as a centralized system and a **loose synchronization over WAN** to reduce communication cost.
- **Results:** WeightGrad achieves,
  - **2.4-5.36× speedup** over state-of-the-art distributed systems
  - **0.03-1.06% accuracy gain** over Baseline

THANK YOU!

Syeda Nahida Akter and Muhammad Abdullah Adnan, "WeightGrad: Geo-Distributed Data Analysis Using Quantization for Faster Convergence and Better Accuracy," Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD 2020), San Diego, CA, USA, August 23-27, 2020.