# AggregaThor: Byzantine Machine Learning via Robust Gradient Aggregation

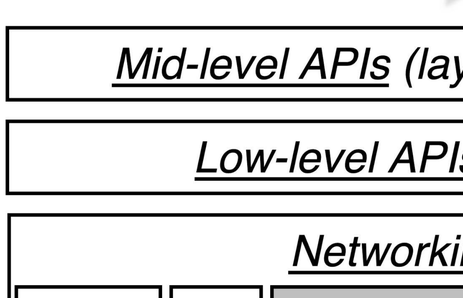
Georgios Damaskinos, El Mahdi El Mhamdi, Rachid Guerraoui,
Arsany Guirguis, Sébastien Rouault
firstname.lastname@epfl.ch
SysML'19, Stanford, CA, USA

EPFL

SYSML

## Problem - Setting

### Distributed ML

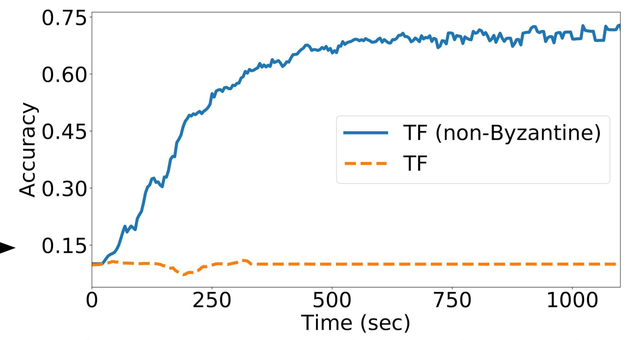



### Threat Model

- Reliable server
- Omniscient workers
- Not omnipotent workers
  - Cannot change other responses
  - Only affect via their response

### Byzantine Worker

Examples:
- unreliable comm.
- software bug
- *corrupted data*
- security flaw




1 Byzantine out of 19 workers can poison learning!

## AggregaThor




### Aggregation layer

- Suitable for *weak* and *strong* Byzantine resilience
- Multi-Krum
  - New scalable *weak* Byzantine-resilient scheme
  - N ≥ 2f + 3 (N: workers, f: bound for Byzantine)
  - Selects $1 \leq m \leq N-f-2$ grads with smallest sum of *scores*:

$$\text{score}_i = \sum_{j \in \{m \text{ gradients with smallest L2 distance from } G_i\}} \|G_i - G_j\|^2$$

- Bulyan [El Mhamdi et al., 2018]
  - Strong Byzantine-resilient scheme
  - N ≥ 4f + 3
  - We prove compliance with multi-Krum for $1 \leq m \leq N-2f-2$

### TensorFlow runtime

- Patch to prevent workers from modifying:
  - TF graph
  - TF shared variables

### LossyMPI

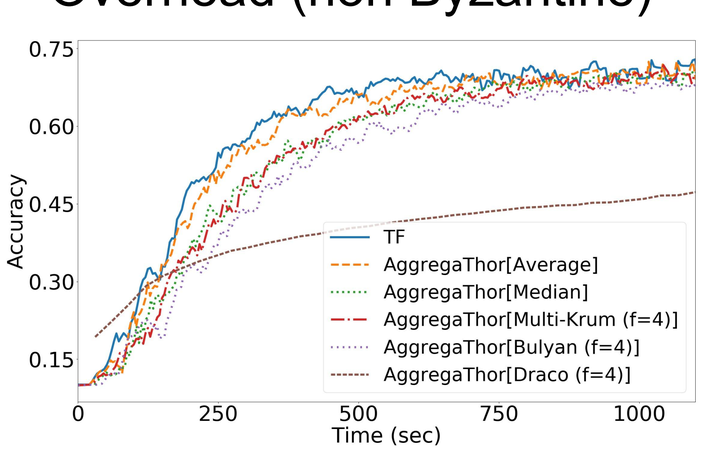
- New UDP-based (unreliable) communication protocol
- Timeout → replace missing values with random ones

### Contributions:

1. Scalable Byzantine Learning
2. Compatible with any TF application using synchronous SGD
3. Performance boost from unreliable communication

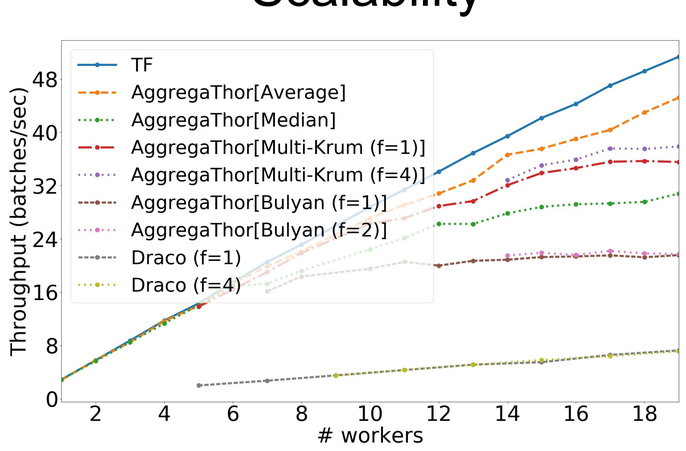
## Evaluation

### Overhead (non Byzantine)



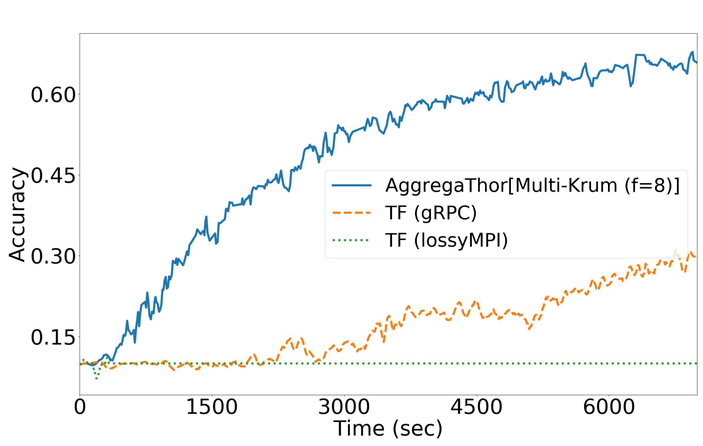

Overhead: 19% to 43%

### Scalability




Better throughput for larger f

### UDP Speedup




LossyMPI is 6x faster than gRPC

## Take away

1. AggregaThor enables scalable Byzantine-resilient learning on top of TensorFlow
2. Unreliable communication can be viewed as a Byzantine fault

Artifacts Available — acm

Artifacts Evaluated — acm — Reusable

https://github.com/LPD-EPFL/AggregaThor