Moshpit SGD: Communication-Efficient Decentralized Training on Heterogeneous Unreliable Devices

Max Ryabinin* — HSE, Yandex

Eduard Gorbunov* — MIPT, HSE, Yandex

Vsevolod Plokhotnyuk — HSE, Yandex

Gennady Pekhimenko — UToronto, Vector

MLO Seminar, EPFL

NATIONAL RESEARCH UNIVERSITY

December 20, 2021

Moshpit SGD: Communication-Efficient Decentralized Training on Heterogeneous Unreliable Devices

Max Ryabinin*
Yandex, Russia
HSE University, Russia

Eduard Gorbunov*
MIPT, Russia
HSE University, Russia
Yandex, Russia

Vsevolod Plokhotnyuk
Yandex, Russia
HSE University, Russia

Gennady Pekhimenko
University of Toronto, Canada
Vector Institute, Canada

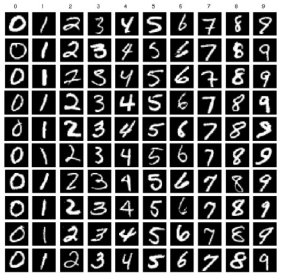
The talk is based on our NeurIPS 2021 paper

Outline

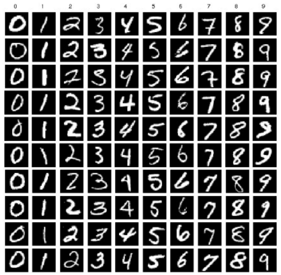
1. Motivation
2. Moshpit All-Reduce
3. Moshpit SGD

1. Motivation

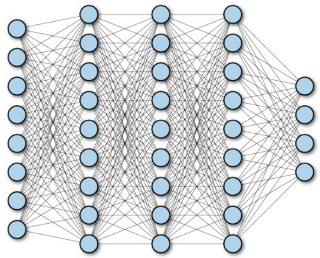
MNIST



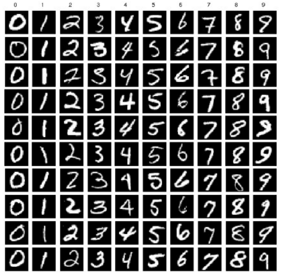
MNIST



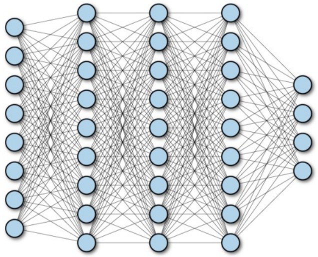
Fully connected NN with 3 hidden layers



MNIST

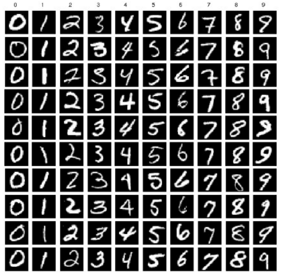


Fully connected NN
with 3 hidden layers

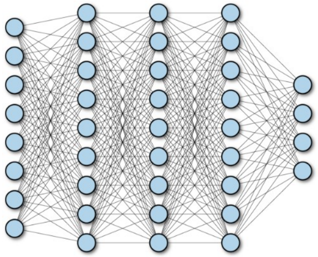


Requires several minutes to train if executed on a good enough laptop

MNIST



Fully connected NN
with 3 hidden layers



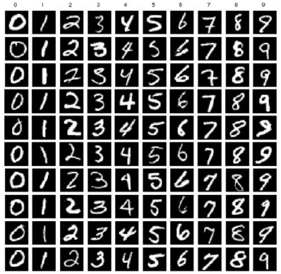
Requires several minutes to train if executed on a good enough laptop

Common Crawl

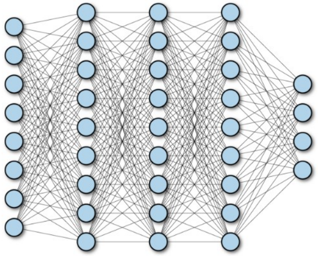
Books Corpus

Wikipedia

MNIST



Fully connected NN
with 3 hidden layers



Requires several minutes to train if executed on a good enough laptop

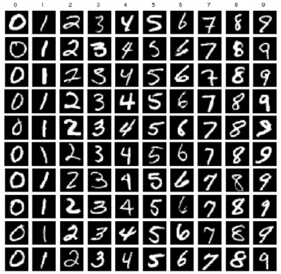
GPT-3

Common Crawl

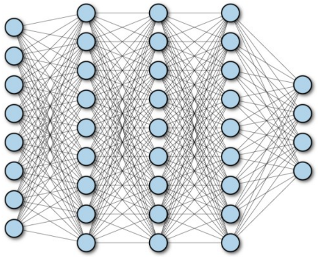
Books Corpus

Wikipedia

MNIST



Fully connected NN
with 3 hidden layers



Requires several minutes to train if executed on a good enough laptop

GPT-3

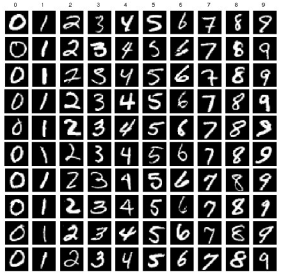
Common Crawl

Books Corpus

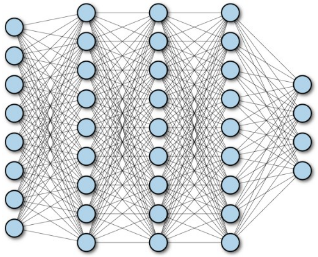
Wikipedia

Requires several **years** to train if executed on top-of-the-line GPU server

MNIST



Fully connected NN with 3 hidden layers



Requires several minutes to train if executed on a good enough laptop

GPT-3

Common Crawl

Books Corpus

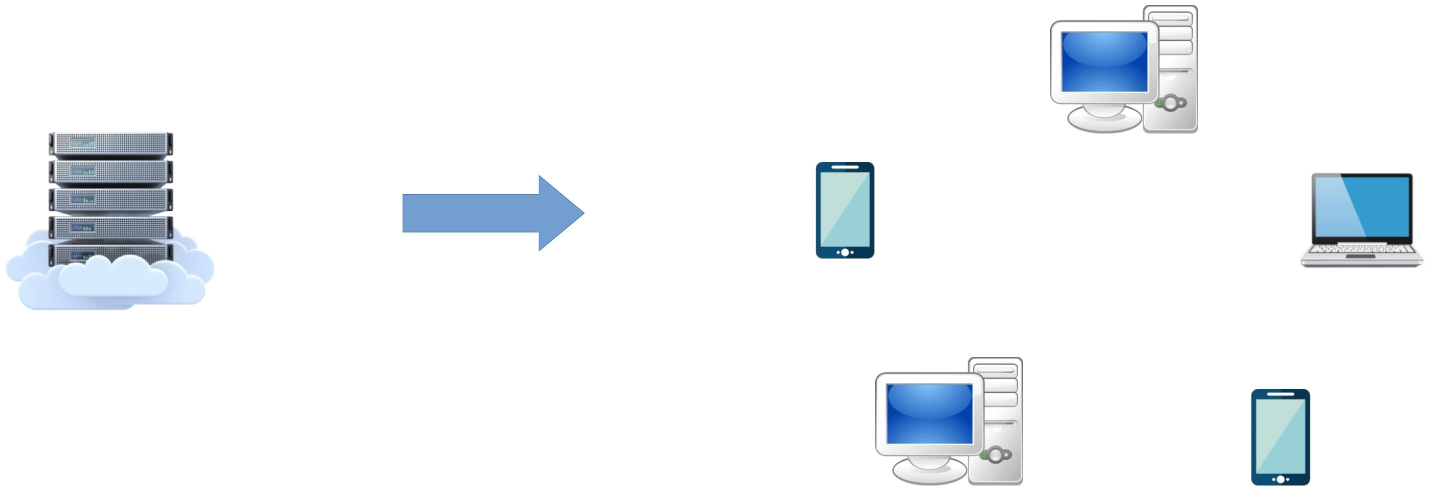
Wikipedia

Requires several **years** to train if executed on top-of-the-line GPU server

It is mandatory to have efficient distributed algorithms

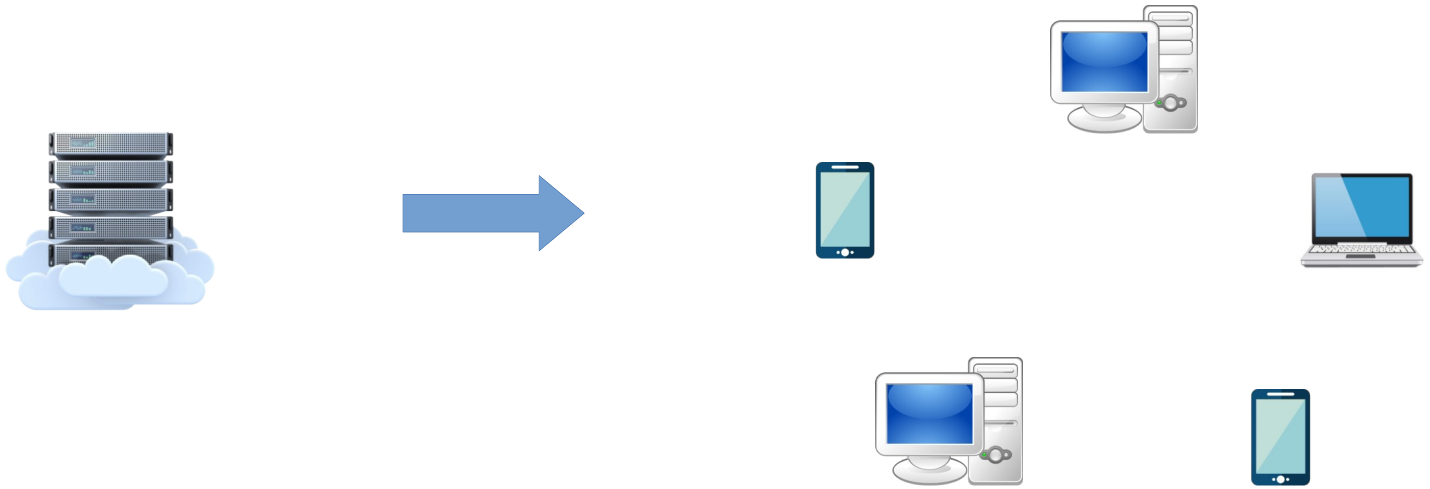
Faulty Workers

Faulty Workers



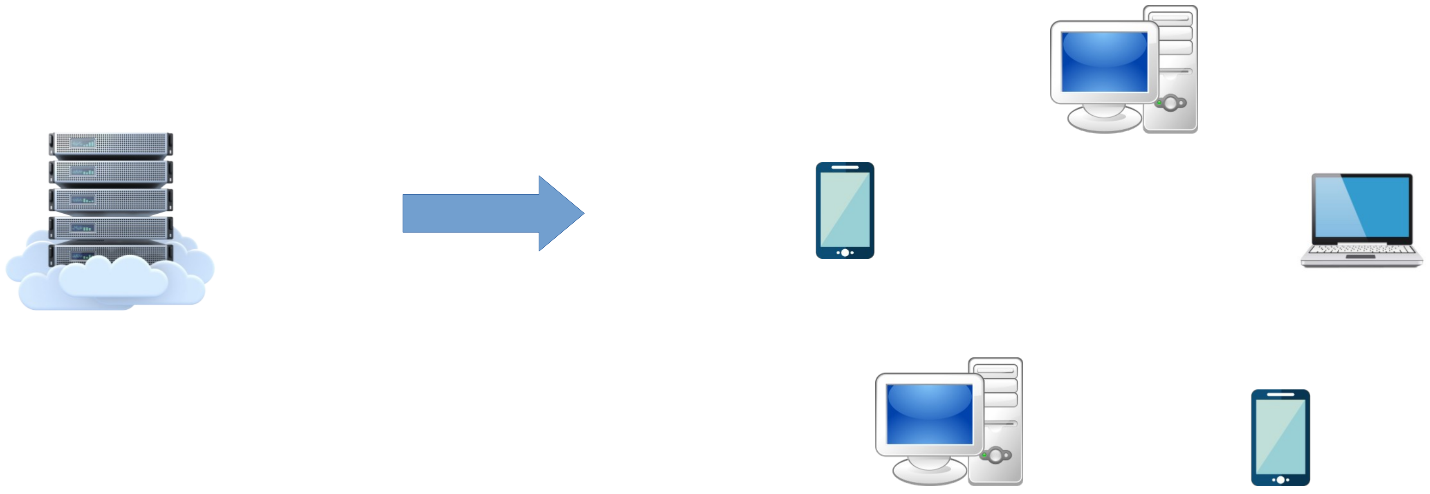
- Instead of one „powerful“ machine, multiple machines are used

Faulty Workers



- Instead of one „powerful“ machine, multiple machines are used
- Some machines can fault to execute the communication protocol at arbitrary stages of the work

Faulty Workers

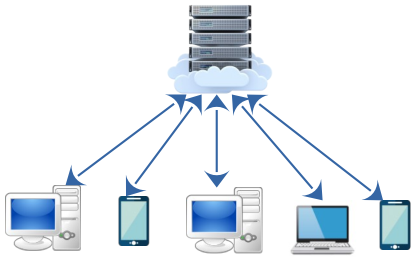


- Instead of one „powerful“ machine, multiple machines are used
- Some machines can fault to execute the communication protocol at arbitrary stages of the work

It is important to have fault-tolerant distributed methods

Communication

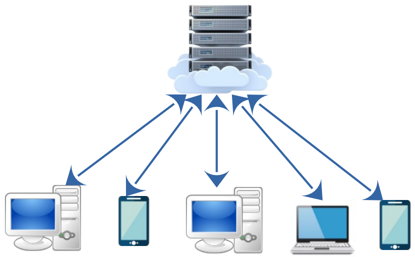
- With Parameter-Server (PS):
  - ✔ Simple and widely applicable approach
  - ✘ Not scalable: for large number of participants the communication is a bottleneck



Devices send and receive full vectors

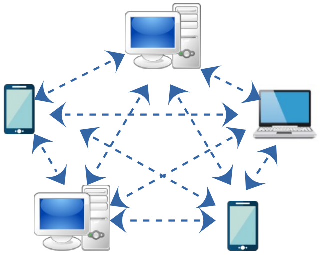
Communication

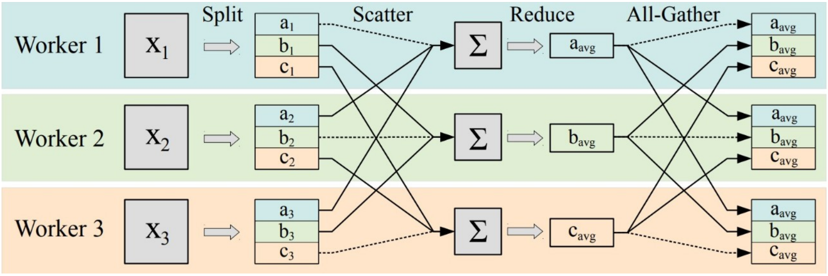
- With Parameter-Server (PS):
  - ✔ Simple and widely applicable approach
  - ✘ Not scalable: for large number of participants the communication is a bottleneck



Devices send and receive full vectors

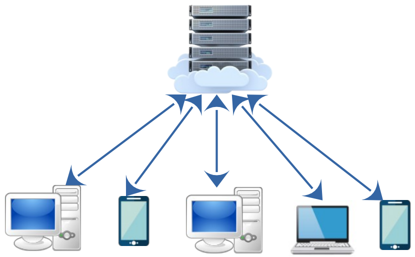
- Without PS via All-Reduce:





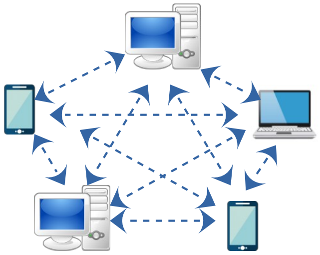
Communication

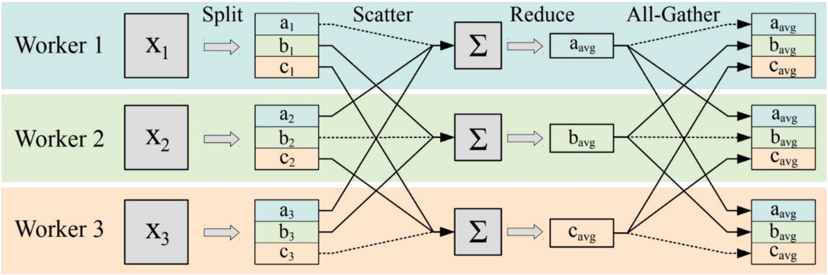
- With Parameter-Server (PS):
  - ✅ Simple and widely applicable approach
  - ❌ Not scalable: for large number of participants the communication is a bottleneck



Devices send and receive full vectors

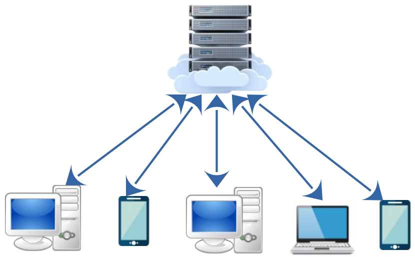
- Without PS via All-Reduce:
  - ✅ Scalable approach
  - ❌ Not robust to faults





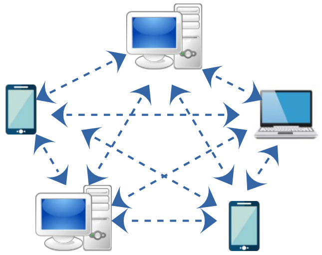
Communication

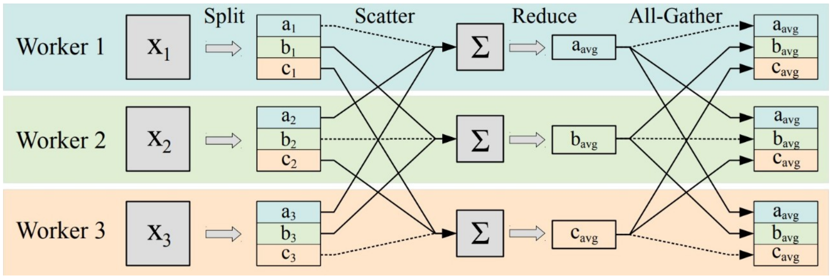
- With Parameter-Server (PS):
  - ✅ Simple and widely applicable approach
  - ❌ Not scalable: for large number of participants the communication is a bottleneck



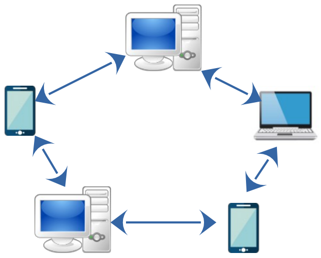
Devices send and receive full vectors

- Without PS via All-Reduce:
  - ✅ Scalable approach
  - ❌ Not robust to faults



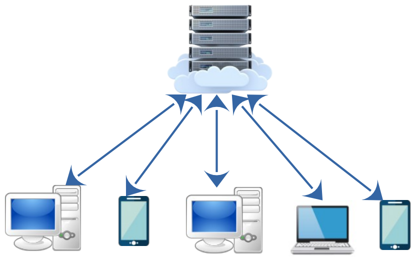


- Without PS via gossip:



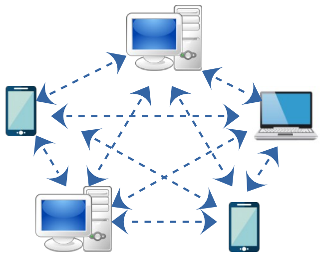
Communication

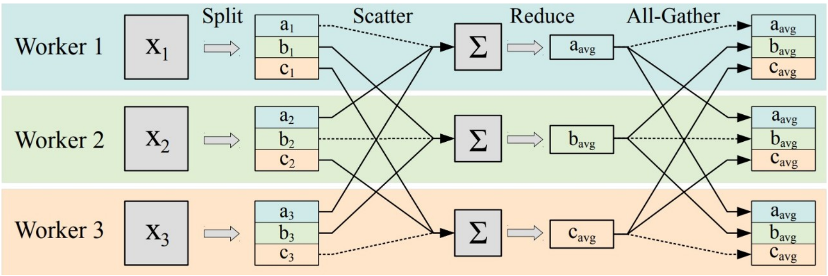
- With Parameter-Server (PS):
  - ✅ Simple and widely applicable approach
  - ❌ Not scalable: for large number of participants the communication is a bottleneck



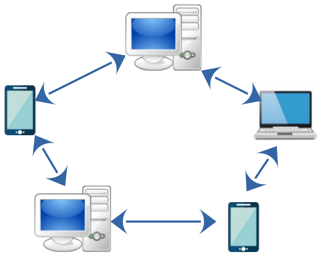
Devices send and receive full vectors

- Without PS via All-Reduce:
  - ✅ Scalable approach
  - ❌ Not robust to faults





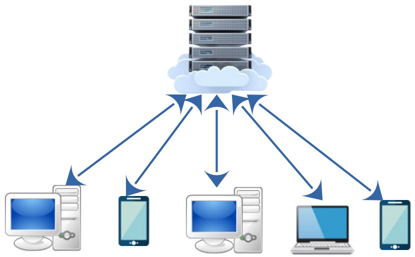
- Without PS via gossip:



$$g_i^{k+1} = \sum_{j=1}^{n} \mathbf{M}_{ij} g_j^k$$

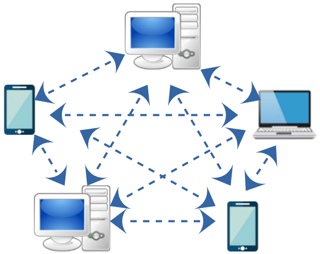
Communication

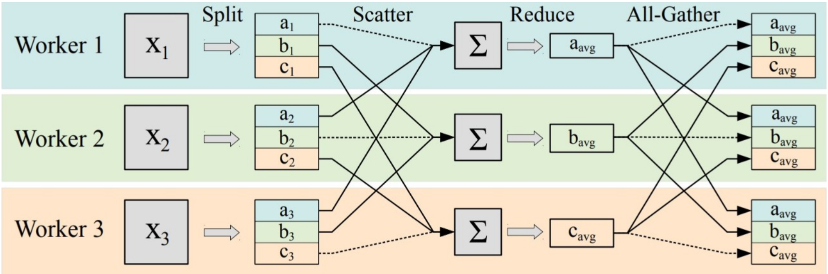
- With Parameter-Server (PS):
  - ✅ Simple and widely applicable approach
  - ❌ Not scalable: for large number of participants the communication is a bottleneck



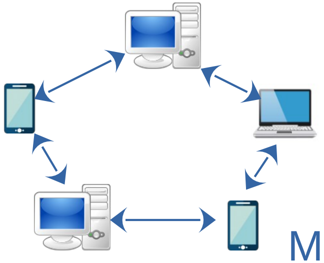
Devices send and receive full vectors

- Without PS via All-Reduce:
  - ✅ Scalable approach
  - ❌ Not robust to faults





- Without PS via gossip:

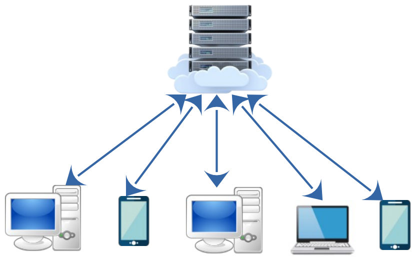


$$g_i^{k+1} = \sum_{j=1}^{n} \mathbf{M}_{ij} g_j^k$$

Mixing matrix defines the communication pattern

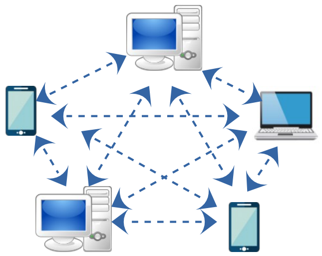
Communication

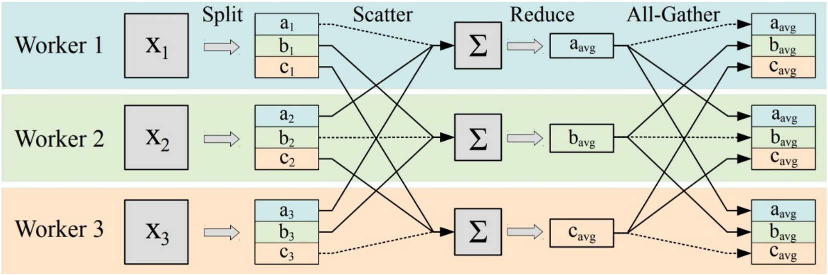
- With Parameter-Server (PS):
  - ✅ Simple and widely applicable approach
  - ❌ Not scalable: for large number of participants the communication is a bottleneck



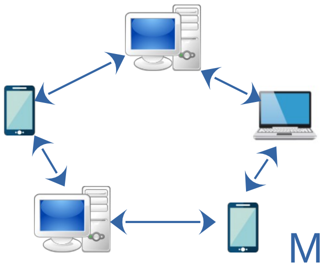
Devices send and receive full vectors

- Without PS via All-Reduce:
  - ✅ Scalable approach
  - ❌ Not robust to faults





- Without PS via gossip:



$$g_i^{k+1} = \sum_{j=1}^{n} \mathbf{M}_{ij} g_j^k$$

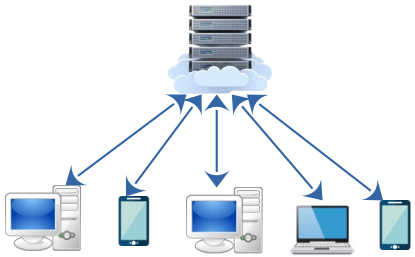
Devices send and receive full vectors

Mixing matrix defines the communication pattern

Communication

With Parameter-Server (PS):

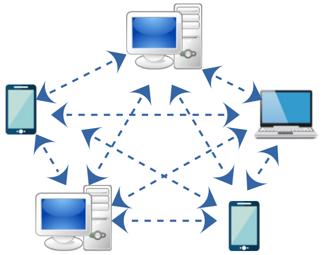
- ✔ Simple and widely applicable approach
- ✘ Not scalable: for large number of participants the communication is a bottleneck

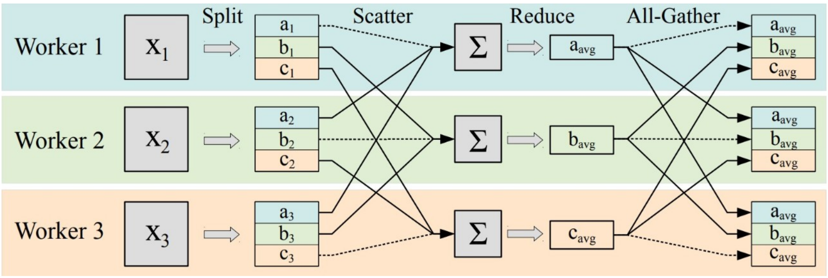


Devices send and receive full vectors

Without PS via All-Reduce:

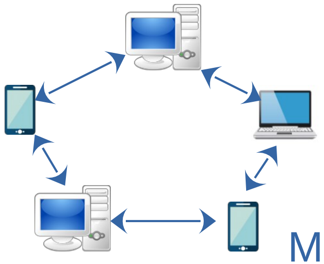
- ✔ Scalable approach
- ✘ Not robust to faults





Without PS via gossip:

- ✔ Scalable approach
- ✘ Inevitable dependence on mixing matrix and graph structure



$$g_i^{k+1} = \sum_{j=1}^{n} \mathbf{M}_{ij} g_j^k$$

Devices send and receive full vectors

Mixing matrix defines the communication pattern

2. Moshpit All-Reduce

Moshpit All-Reduce: Main Idea

- All-Reduce protocols are fragile: the fault of 1 worker affects all other workers

Moshpit All-Reduce: Main Idea

- All-Reduce protocols are fragile: the fault of 1 worker affects all other workers
- The idea: execute All-Reduce in small groups

Moshpit All-Reduce: Main Idea

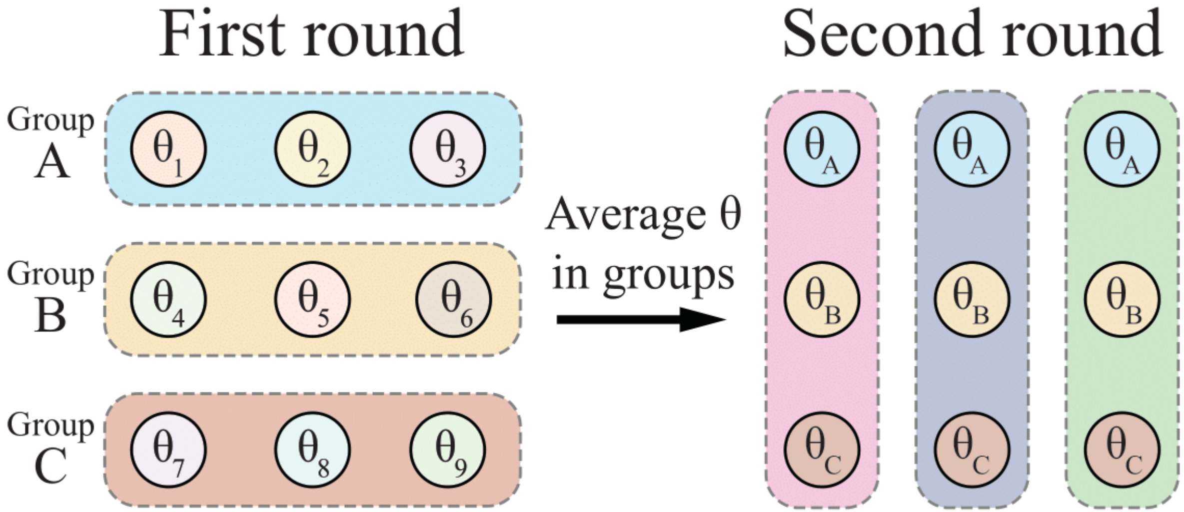
- All-Reduce protocols are fragile: the fault of 1 worker affects all other workers
- The idea: execute All-Reduce in small groups

↓

The fault of one peer affects only its group

Moshpit All-Reduce: Ideal Case

Workers form *d* dimensional hypercube with *M* workers along each axis



$d = 2$, $M = 3$

Moshpit All-Reduce: General Case

Algorithm 1 Moshpit All-Reduce (for i-th peer)

Input: parameters $\{\theta_j\}_{j=1}^{N}$, number of peers N, d, M, number of iterations T, peer index i
$\theta_i^0 := \theta_i$
$C_i^0 :=$ `get_initial_index(i)`
for $t \in 1 \dots T$ **do**
  `DHT`$[C_i^{t-1}, t]$`.add`$($`address`$_i)$
  `Matchmaking()` // wait for peers to assemble
  `peers`$_t :=$ `DHT.get`$([C_i^{t-1}, t])$
  $\theta_i^t, c_i^t :=$ `AllReduce`$(\theta_i^{t-1},$ `peers`$_t)$
  $C_i^t := (C_i^{t-1}$`[1:]`$, c_i^t)$ // same as eq. (1)
end for
Return θ_i^T

$$\text{get\_initial\_index}\ (i) = \left(\left\lfloor i/M^{d-1} \right\rfloor \bmod M\right)_{j\in\{1,\ldots,d\}}$$

$$C_i^t := \left(c_i^{t-d+1}, c_i^{t-d+2}, \ldots, c_i^t\right)$$

Distributed Hash Table — an efficient decentralized data structure

Moshpit All-Reduce: Theoretical Properties

- If $N = M^d$ and there are no faults, then Moshpit All-Reduce finds an exact average after d steps

Moshpit All-Reduce: Theoretical Properties

- If $N = M^d$ and there are no faults, then Moshpit All-Reduce finds an exact average after d steps
- **Correctness:** if all workers have a non-zero probability of successfully running a communication round and the order of peers_t is random, then all local vectors converge to the global average with probability 1:

Moshpit All-Reduce: Theoretical Properties

- If $N = M^d$ and there are no faults, then Moshpit All-Reduce finds an exact average after d steps

- **Correctness:** if all workers have a non-zero probability of successfully running a communication round and the order of peers_t is random, then all local vectors converge to the global average with probability 1:

$$\forall i \quad \left\| \theta_i^t - \frac{1}{N} \sum_i \theta_i^0 \right\|_2^2 \xrightarrow[t\to\infty]{} 0$$

Moshpit All-Reduce: Theoretical Properties

- If $N = M^d$ and there are no faults, then Moshpit All-Reduce finds an exact average after d steps

- **Correctness:** if all workers have a non-zero probability of successfully running a communication round and the order of peers_t is random, then all local vectors converge to the global average with probability 1:

$$\forall i \quad \left\| \theta_i^t - \frac{1}{N}\sum_i \theta_i^0 \right\|_2^2 \xrightarrow[t\to\infty]{} 0$$

- **Exponential convergence to the average:** for a version of Moshpit All-Reduce with random splitting into r groups at each step, we have

Moshpit All-Reduce: Theoretical Properties

- If $N = M^d$ and there are no faults, then Moshpit All-Reduce finds an exact average after d steps
- **Correctness:** if all workers have a non-zero probability of successfully running a communication round and the order of peers_t is random, then all local vectors converge to the global average with probability 1:

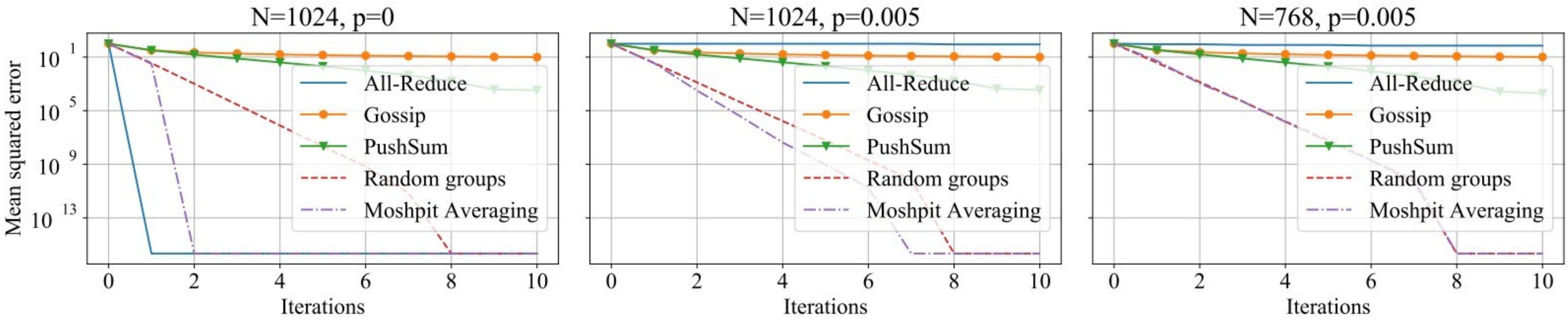
$$\forall i \quad \left\| \theta_i^t - \boxed{\frac{1}{N}\sum_i \theta_i^0} \right\|_2^2 \xrightarrow[t\to\infty]{} 0$$

- **Exponential convergence to the average:** for a version of Moshpit All-Reduce with random splitting into r groups at each step, we have

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\left\|\theta_i^T - \boxed{\bar{\theta}}\right\|^2\right] = \left(\frac{r-1}{N} + \frac{r}{N^2}\right)^T \frac{1}{N}\sum_{i=1}^{N}\left\|\theta_i - \boxed{\bar{\theta}}\right\|^2$$

Moshpit All-Reduce: Experiments

- We verify the performance gains in a controlled setting
- With non-zero failure probability, All-Reduce takes too many retries!
- On the other hand, Gossip-based methods converge very slowly
- Moshpit All-Reduce outperforms baselines with $p > 0$ and gets the average in two rounds with $p = 0$



3. Moshpit SGD

The Problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

- Function *f(x)* is available through stochastic gradients only
- Each worker has an access to the stochastic gradients of *f(x)*

Moshpit SGD

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{if } k+1 \bmod \tau \neq 0 \\ \texttt{Moshpit All-Reduce}_{j \in P_{k+1}}(x_j - \gamma g_j^k), & \text{if } k+1 \bmod \tau = 0 \end{cases}$$

Number of active workers
at iteration *k+1*

Moshpit SGD

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{if } k+1 \bmod \tau \neq 0 \\ \texttt{Moshpit All-Reduce}_{j \in P_{k+1}}(x_j - \gamma g_j^k), & \text{if } k+1 \bmod \tau = 0 \end{cases}$$

Number of active workers at iteration *k+1*

Local-SGD with Moshpit All-Reduce instead of averaging

Assumptions

- Homogeneity: $f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$

Assumptions

- Homogeneity: $$f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$$
- Bounded variance: $$\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x_i^k\right)\right\|^2 \mid x_i^k\right] \leq \sigma^2$$

Assumptions

- Homogeneity:

$$f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$$

- Bounded variance:

$$\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x_i^k\right)\right\|^2 \mid x_i^k\right] \leq \sigma^2$$

- Effect of peers' vanishing is bounded:

$$\mathbb{E}\left[\left\langle x^{k+1} - \widehat{x}^{k+1}, x^{k+1} + \widehat{x}^{k+1} - 2x^*\right\rangle\right] \leq \Delta_{pv}^k$$

Assumptions

- Homogeneity: $$f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$$
- Bounded variance: $$\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x_i^k\right)\right\|^2 \mid x_i^k\right] \leq \sigma^2$$
- Effect of peers' vanishing is bounded: $$\mathbb{E}\left[\left\langle \boxed{x^{k+1}} - \widehat{x}^{k+1}, \boxed{x^{k+1}} + \widehat{x}^{k+1} - 2x^* \right\rangle\right] \leq \Delta_{pv}^k$$

$$\boxed{x^{k+1} = \frac{1}{N_{k+1}} \sum_{i \in P_{k+1}} x_i^{k+1}}$$

Assumptions

- Homogeneity: $$f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$$
- Bounded variance: $$\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x_i^k\right)\right\|^2 \mid x_i^k\right] \le \sigma^2$$
- Effect of peers' vanishing is bounded: $$\mathbb{E}\left[\left\langle \boxed{x^{k+1}} - \widehat{x}^{k+1}, \boxed{x^{k+1}} + \widehat{x}^{k+1} - 2x^*\right\rangle\right] \le \Delta_{pv}^k$$

$$\boxed{N_k = |P_k|}$$

$$\boxed{x^{k+1} = \frac{1}{\boxed{N_{k+1}}} \sum_{i \in P_{k+1}} x_i^{k+1}}$$

Assumptions

- Homogeneity: $$f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$$
- Bounded variance: $$\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x_i^k\right)\right\|^2 \mid x_i^k\right] \leq \sigma^2$$
- Effect of peers' vanishing is bounded: $$\mathbb{E}\left[\left\langle \boxed{x^{k+1}} - \boxed{\widehat{x}^{k+1}}, \boxed{x^{k+1}} + \boxed{\widehat{x}^{k+1}} - 2x^*\right\rangle\right] \leq \Delta_{pv}^k$$

$$\boxed{N_k = |P_k|}$$

$$\boxed{x^{k+1} = \frac{1}{\boxed{N_{k+1}}} \sum_{i \in P_{k+1}} x_i^{k+1}}$$

$$\boxed{\widehat{x}^{k+1} = \frac{1}{N_k} \sum_{i \in P_k} \left(x_i^k - \gamma g_i^k\right)}$$

Assumptions

- Function f is μ-strongly convex

Assumptions

- Function f is μ-strongly convex
- Averaging quality:

$$\mathbb{E}\left[\frac{1}{n_{a\tau}}\sum_{i\in P_{a\tau}}\left\|x_i^{a\tau}-x^{a\tau}\right\|^2\right]\leq\gamma^2\delta_{aq}^2$$

Moshpit SGD: Complexity

Moshpit SGD finds $\hat{x}$ such that $\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \varepsilon$ after

Moshpit SGD: Complexity

Moshpit SGD finds $\hat{x}$ such that $\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \varepsilon$ after

$$\widetilde{\mathcal{O}}\left(\frac{L}{(1-\delta_{pv,1})\,\mu} + \frac{\delta_{pv,2}^2 + \sigma^2/n_{\min}}{(1-\delta_{pv,1})\,\mu\varepsilon} + \sqrt{\frac{L\left((\tau-1)\sigma^2 + \delta_{aq}^2\right)}{(1-\delta_{pv,1})^2\,\mu^2\varepsilon}}\right)$$

iterations
when $\mu > 0$

Moshpit SGD: Complexity

Moshpit SGD finds $\hat{x}$ such that $\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \varepsilon$ after

$$\widetilde{\mathcal{O}}\left(\frac{L}{(1-\delta_{pv,1})\,\mu} + \frac{\delta_{pv,2}^2 + \sigma^2/n_{\min}}{(1-\delta_{pv,1})\,\mu\varepsilon} + \sqrt{\frac{L\left((\tau-1)\sigma^2 + \delta_{aq}^2\right)}{(1-\delta_{pv,1})^2\,\mu^2\varepsilon}}\right)$$

iterations when $\mu > 0$

$$\mathcal{O}\left(\frac{LR_0^2}{\varepsilon} + \frac{R_0^2\left(\delta_{pv,2}^2 + \sigma^2/n_{\min}\right)}{\varepsilon^2} + \frac{R_0^2\sqrt{L\left((\tau-1)\sigma^2 + \delta_{aq}^2\right)}}{\varepsilon^{3/2}}\right)$$

iterations when $\mu = 0$

Moshpit SGD: Complexity

Moshpit SGD finds $\hat{x}$ such that $\mathbb{E}\left[f(\hat{x}) - f(x^*)\right] \leq \varepsilon$ after

$$\widetilde{\mathcal{O}}\left(\frac{L}{(1-\delta_{pv,1})\,\mu} + \frac{\delta_{pv,2}^2 + \sigma^2/n_{\min}}{(1-\delta_{pv,1})\,\mu\varepsilon} + \sqrt{\frac{L\left((\tau-1)\sigma^2 + \delta_{aq}^2\right)}{(1-\delta_{pv,1})^2\,\mu^2\varepsilon}}\right)$$

iterations when *μ* > *0*

$$\mathcal{O}\left(\frac{LR_0^2}{\varepsilon} + \frac{R_0^2\left(\delta_{pv,2}^2 + \sigma^2/n_{\min}\right)}{\varepsilon^2} + \frac{R_0^2\sqrt{L\left((\tau-1)\sigma^2 + \delta_{aq}^2\right)}}{\varepsilon^{3/2}}\right)$$

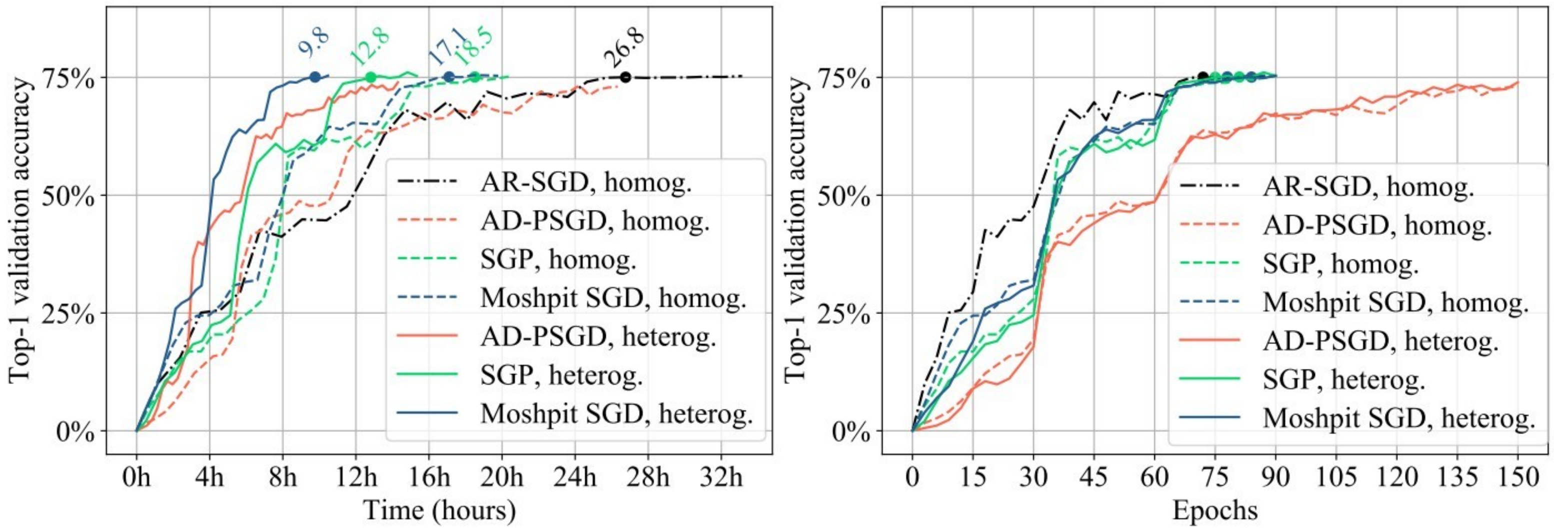
iterations when *μ* = *0*

If $\delta_{pv,1} \leq 1/2,\, N_{\min} = \Omega(N),\, \delta_{pv,2}^2 = \mathcal{O}\left(\sigma^2/N_{\min}\right),\, \delta_{aq}^2 = \mathcal{O}((\tau-1)\sigma^2)$, then

the complexity of Moshpit SGD matches the complexity of <u>centralized</u> Local-SGD

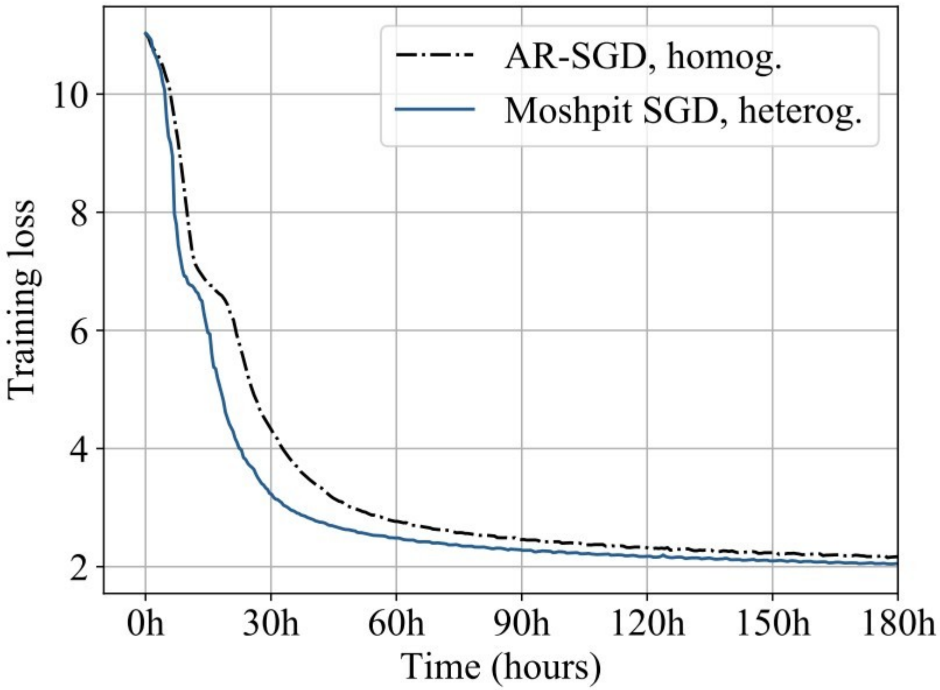
Moshpit SGD: ResNet-50 on Imagenet

- We evaluate and several baselines in two environments
- (16 nodes with 1xV100 and 64 workers with 81 different GPUs)
- Comparable to All-Reduce in terms of iterations, faster in terms of time
- Decentralized methods run faster, but achieve worse results



Moshpit SGD: ALBERT on BookCorpus

- Baseline: All-Reduce on 8 V100
- Moshpit SGD: 66 preemptible GPUs
- Cost of spot instances are much smaller, yet we converge 1.5x faster



4. Conclusion

Summary

- We propose a simple method for communication-efficient distributed training
- Built-in fault tolerance, convergence similar to standard methods

Some of My Recent Works

EG, Marina Danilova, Innokentiy Shibaev, Pavel Dvurechensky, Alexander Gasnikov
Near-Optimal High Probability Complexity Bounds for Non-Smooth Stochastic Optimization with Heavy-Tailed Noise
arXiv:2106.05958

EG*, Alexander Borzunov*, Michael Diskin, Max Ryabinin
Secure Distributed Training at Scale
arXiv:2106.11257

Ilyas Fatkhullin, Igor Sokolov, EG, Zhize Li, Peter Richtárik
EF21 with Bells & Whistles: Practical Algorithmic Extensions of Modern Error Feedback
arXiv:2110.03294

EG, Nicolas Loizou, Gauthier Gidel
Extragradient Method: O(1/K) Last-Iterate Convergence for Monotone Variational Inequalities and Connections With Cocoercivity
arXiv:2110.04261

EG, Hugo Berard, Gauthier Gidel, Nicolas Loizou
Stochastic Extragradient: General Analysis and Improved Rates
arXiv:2111.08611