Communication Efficient Distributed Training

Ji Liu, Ph.D.

[https://www.eyhost.biz/big_data.html]

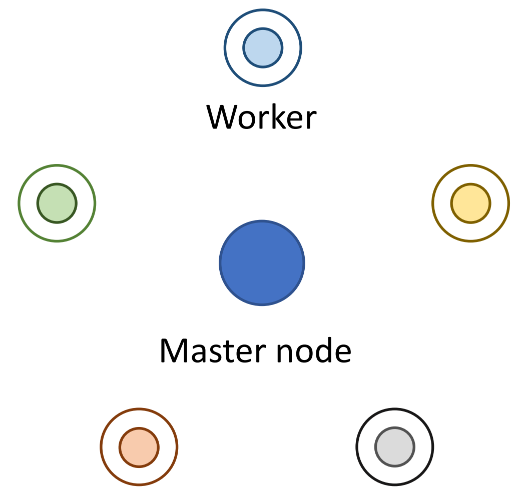
Objective

model

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{a}^{(i)} \sim \mathcal{D}_i} F(\boldsymbol{x}; \boldsymbol{a}^{(i)})$$

# of workers

samples on worker i

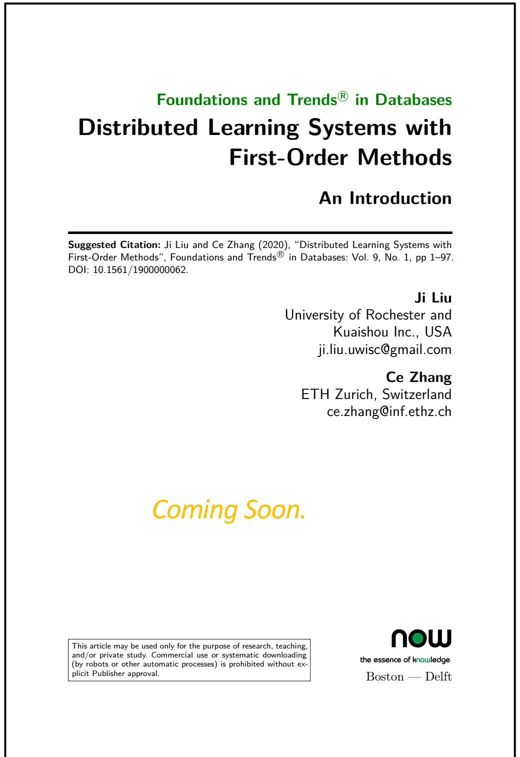


Centralized distributed learning

All functions are assumed to be L-Lipschitzian

How to reduce communication cost?

Summary



Foundations and Trends® in Databases

Distributed Learning Systems with First-Order Methods

An Introduction

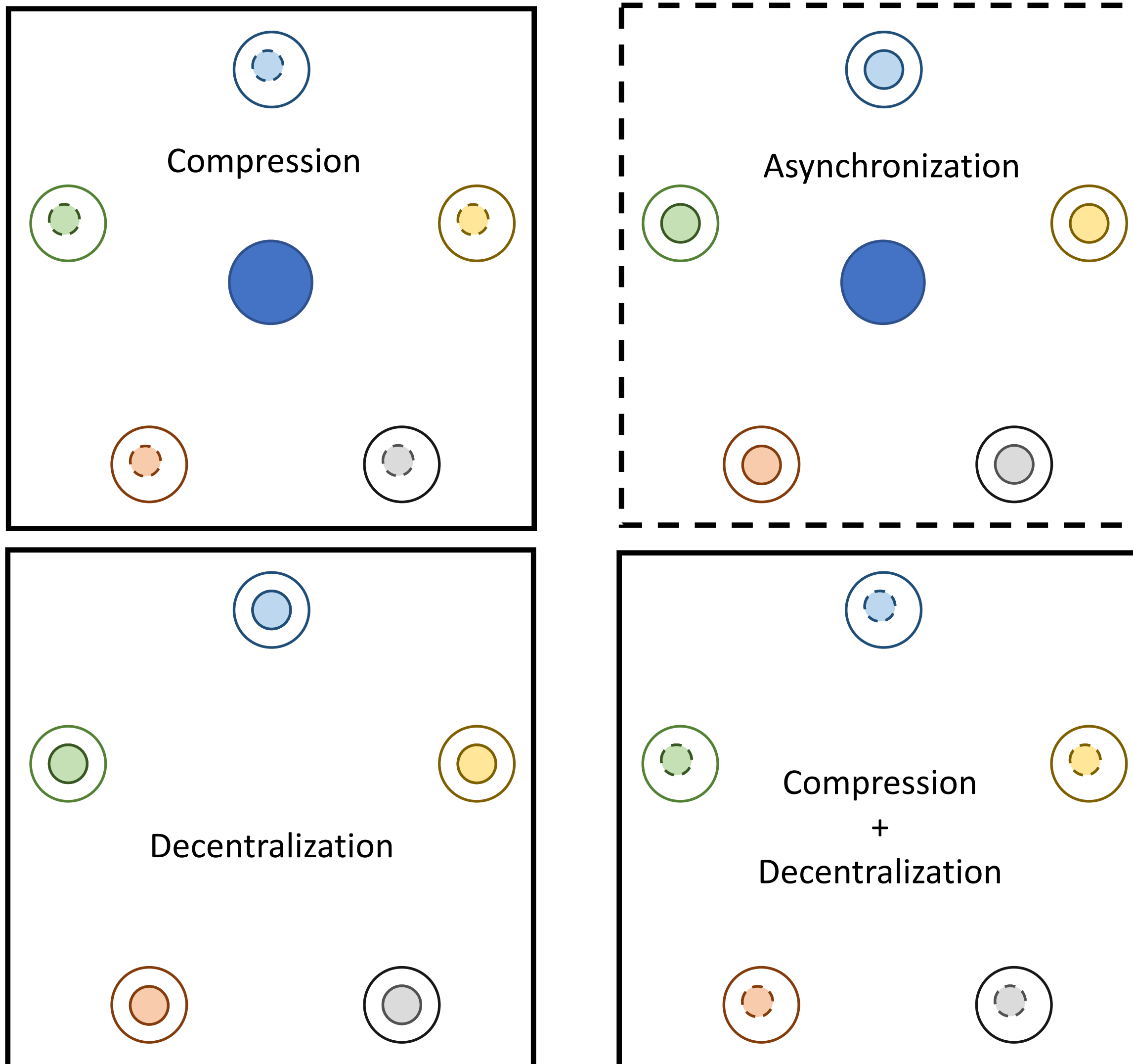
Suggested Citation: Ji Liu and Ce Zhang (2020), "Distributed Learning Systems with First-Order Methods", Foundations and Trends® in Databases: Vol. 9, No. 1, pp 1–97. DOI: 10.1561/1900000062.

Ji Liu
University of Rochester and
Kuaishou Inc., USA
ji.liu.uwisc@gmail.com

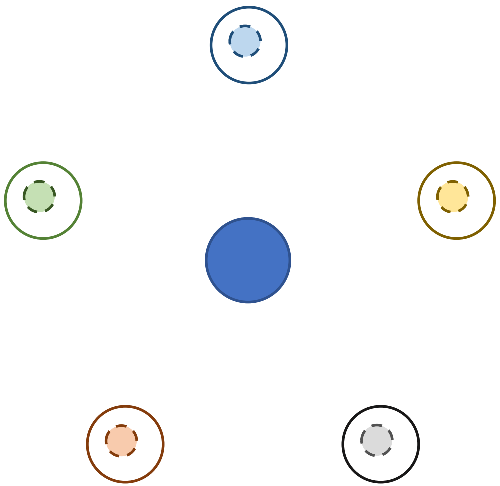
Ce Zhang
ETH Zurich, Switzerland
ce.zhang@inf.ethz.ch

Coming Soon.

now
the essence of knowledge
Boston — Delft



Compression

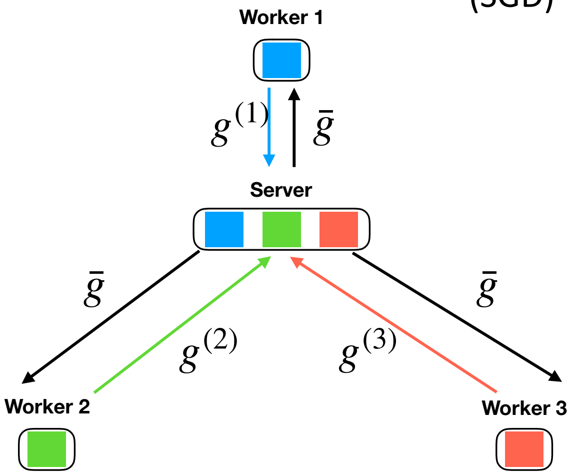


Algorithm

$\mathbf{C}(\cdot)$ Compression operator (maybe randomized)

$$\boldsymbol{g}^{(i)} := \nabla F(\boldsymbol{x}; \boldsymbol{a}^{(i)})$$

(SGD) $\mathbf{x} \leftarrow \mathbf{x} - \gamma \bar{\mathbf{g}}$



$$\bar{\mathbf{g}} = \frac{1}{3}(\mathbf{g}^{(1)} + \mathbf{g}^{(2)} + \mathbf{g}^{(3)})$$ (Standard)

Exchange 2N full vectors

$$\bar{\mathbf{g}} = \frac{1}{3}(\mathbf{C}(\mathbf{g}^{(1)}) + \mathbf{C}(\mathbf{g}^{(2)}) + \mathbf{C}(\mathbf{g}^{(3)}))$$ (Single compression)

Exchange N(1+c) full vectors

$$\bar{\mathbf{g}} = \mathbf{C}\left(\frac{1}{3}(\mathbf{C}(\mathbf{g}^{(1)}) + \mathbf{C}(\mathbf{g}^{(2)}) + \mathbf{C}(\mathbf{g}^{(3)}))\right)$$ (Double compression)

Exchange 2cN full vectors

Unfortunately

To ensure convergence, it should satisfy $\mathbb{E}(\mathbf{C}(\mathbf{x})) = \mathbf{x}$

Early methods only work for $\mathbf{C}(\cdot)$ compression operator

- *Randomized quantization (unbiased)*
- ~~*Randomized quantization (biased)*~~
- ~~*1bit quantization*~~
- ~~*Clipping*~~
- ~~*Top-k sparsification*~~

Can we relax it to allow more aggressive or even arbitrary compression?

Double Squeeze: Error Compensated SGD

Worker i

$$\boldsymbol{g}^{(i)} \leftarrow \nabla F(\boldsymbol{x}; \boldsymbol{a}^{(i)})$$

Local gradient

$$\boldsymbol{v}^{(i)} \leftarrow \mathcal{C}\left(\boldsymbol{g}^{(i)} + \boldsymbol{\delta}^{(i)}\right)$$

Error compensation

$$\boldsymbol{\delta}^{(i)} \leftarrow \left(\boldsymbol{g}^{(i)} + \boldsymbol{\delta}^{(i)}\right) - \boldsymbol{v}^{(i)}$$

Compression error

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \gamma \overline{\boldsymbol{v}}$$

Pull $\overline{\mathbf{V}}$ to update model

Master s

$$\overline{\boldsymbol{g}} \leftarrow \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}^{(i)}$$

Aggregate compressed gradient

$$\overline{\boldsymbol{v}} \leftarrow \mathcal{C}\left(\overline{\boldsymbol{g}} + \overline{\boldsymbol{\delta}}\right)$$

Compress the error compensated g

$$\overline{\boldsymbol{\delta}} \leftarrow \left(\overline{\boldsymbol{g}} + \overline{\boldsymbol{\delta}}\right) - \overline{\boldsymbol{v}}$$

Compression error

Intuition

Essential updating rule of DoubleSqueeze (SGD alike)

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma \overline{\boldsymbol{g}}_t + \gamma(\hat{\boldsymbol{\delta}}_t - \hat{\boldsymbol{\delta}}_{t-1})$$

$$\overline{\boldsymbol{g}}_t = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_t^i$$

$$\hat{\boldsymbol{\delta}}_t = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\delta}_t^{(i)} + \overline{\boldsymbol{\delta}}_t$$

C-SGD (Uncompressed)

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_0 - \gamma \sum_{s=0}^{t} \overline{\boldsymbol{g}}_s$$

Naive Compressed C-SGD

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_0 - \gamma \sum_{s=0}^{t} \overline{\boldsymbol{g}}_s + \gamma \sum_{s=0}^{t} \hat{\boldsymbol{\delta}}_s$$

DoubleSqueeze

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_0 - \gamma \sum_{s=0}^{t} \overline{\boldsymbol{g}}_s + \gamma \hat{\boldsymbol{\delta}}_t$$

Much smaller

Convergence

Assumption
$\mathbb{E}[\|\mathbf{C}(\mathbf{x}) - \mathbf{x}\|^2] \leq \sigma'^2$

Convergence rates

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left(\|\nabla f(\overline{\boldsymbol{x}}_t)\|^2\right) \lesssim \frac{1}{T} + \frac{\sigma}{\sqrt{nT}}$$

SGD

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left(\|\nabla f(\overline{\boldsymbol{x}}_t)\|^2\right) \lesssim \frac{1}{T} + \frac{\sigma}{\sqrt{nT}} + \frac{\sigma'}{\sqrt{T}}$$

C-SGD (C(.) needs to be unbiased)

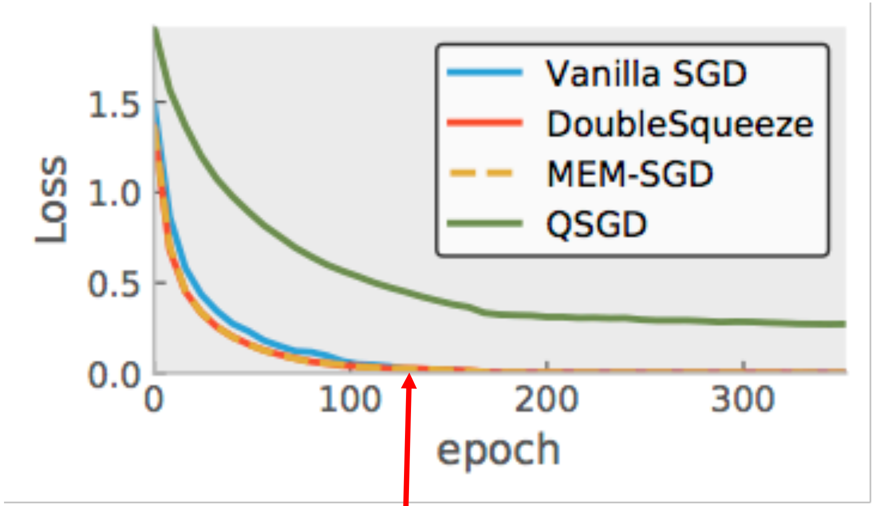
$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left(\|\nabla f(\overline{\boldsymbol{x}}_t)\|^2\right) \lesssim \frac{1}{T} + \frac{\sigma}{\sqrt{nT}} + \left(\frac{\sigma'}{T}\right)^{\frac{2}{3}}$$

Double squeeze
EC-SGD

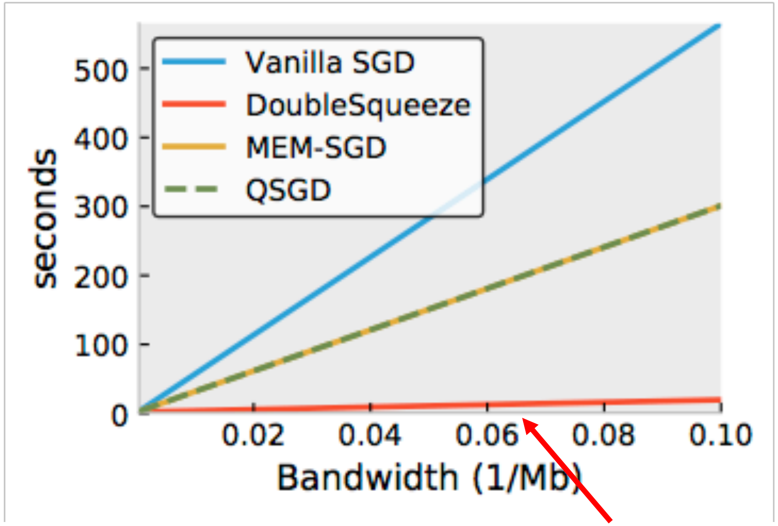
better

Experiments

ResNet-18. CIFAR-10. 8 workers

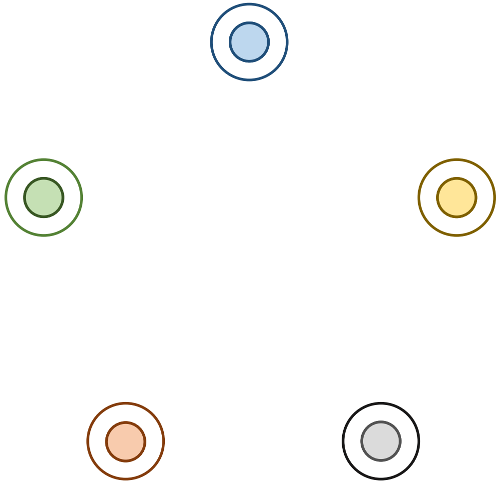


Iteration (epoch) is consistent with SGD

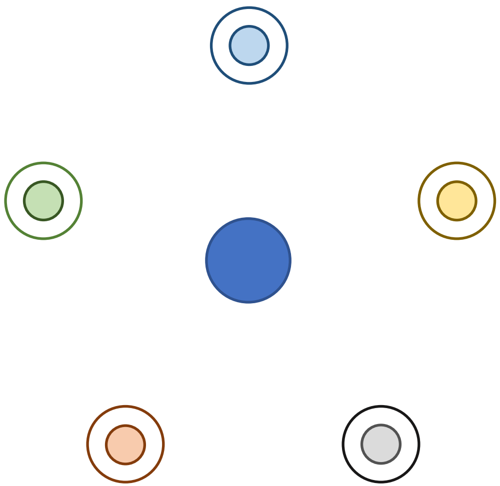


Running time in each iteration is faster

Decentralization



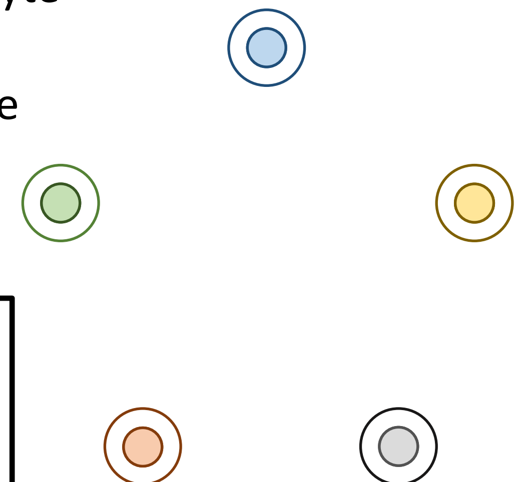
- alpha: latency per message
- beta: transfer time per byte
- N: # workers
- B: # bytes of the message



Centralized communication
(fully exchanged)

O(N * alpha + NB * beta)

How does the decentralized approach compare to the centralized approach?

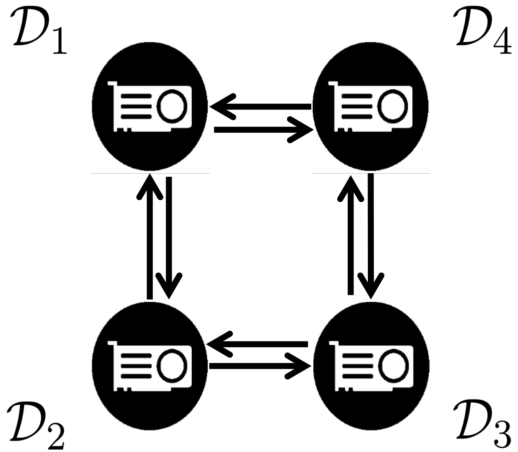


Decentralized communication
(partially exchanged)

O(alpha + B * beta)

Objective

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) := \frac{1}{n} \sum_{i=1}^{n} \underbrace{\mathbb{E}_{\boldsymbol{a}^{(i)} \sim \mathcal{D}_i} F(\boldsymbol{x}; \boldsymbol{a}^{(i)})}_{=: f_i(\boldsymbol{x})}$$



Centralized-SGD:

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \gamma \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{g}(\boldsymbol{x}; \boldsymbol{a}^{(i)})$$

shared model

local sample

Centralized-SGD:

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \gamma \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{g}(\boldsymbol{x}; \boldsymbol{a}^{(i)})$$

Decentralized-SGD:

$$\begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \cdots \\ \boldsymbol{x}^{(n)} \end{bmatrix} \leftarrow W \left(\begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \cdots \\ \boldsymbol{x}^{(n)} \end{bmatrix} - \gamma \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}^{(1)}; \boldsymbol{a}^{(1)}) \\ \boldsymbol{g}(\boldsymbol{x}^{(2)}; \boldsymbol{a}^{(2)}) \\ \cdots \\ \boldsymbol{g}(\boldsymbol{x}^{(n)}; \boldsymbol{a}^{(n)}) \end{bmatrix} \right)$$

Local model

Local sample

n individual models

Average the local model with neighbor's, e.g.,

$$\mathbf{x}^{(2)} \leftarrow \frac{1}{3} \sum_{i=1,2,3} \left(\mathbf{x}^{(i)} - \gamma \mathbf{g}^{(i)} \right)$$

Decentralized SGD

$$\begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \cdots \\ \boldsymbol{x}^{(n)} \end{bmatrix} \leftarrow \boxed{W} \left(\begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \\ \cdots \\ \boldsymbol{x}^{(n)} \end{bmatrix} - \gamma \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}^{(1)}; \boldsymbol{a}^{(1)}) \\ \boldsymbol{g}(\boldsymbol{x}^{(2)}; \boldsymbol{a}^{(2)}) \\ \cdots \\ \boldsymbol{g}(\boldsymbol{x}^{(n)}; \boldsymbol{a}^{(n)}) \end{bmatrix} \right)$$

weight matrix: symmetric, doubly stochastic

(W1 = 1, W^T1=1, nonnegative, W=W^T)

ring network

$$W = \begin{pmatrix} 1/3 & 1/3 & & & & 1/3 \\ 1/3 & 1/3 & 1/3 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1/3 & 1/3 & 1/3 \\ 1/3 & & & & 1/3 & 1/3 \end{pmatrix}$$

Assumptions

- **Lipschitzian** All $f_i(\cdot)$ are with L-Lipschitzian gradient
- **Bounded variance**

$$\mathbb{E}_{\boldsymbol{a}\sim\mathcal{D}_i}\|\nabla F(\boldsymbol{x};\boldsymbol{a})-\nabla f_i(\boldsymbol{x})\|^2 \le \sigma^2,\ \forall i,\ \forall \boldsymbol{x}$$

$$\|\nabla f_i(\boldsymbol{x})-\nabla f(\boldsymbol{x})\|^2 \le \zeta^2,\ \forall i,\ \forall \boldsymbol{x}$$

data variance **within** each worker (σ^2)

data variance **among** workers (ζ^2)

Assumptions

- **Lipschitzian** All $f_i(\cdot)$ are with L-Lipschitzian gradient
- **Bounded variance**

$$\mathbb{E}_{\boldsymbol{a}\sim\mathcal{D}_i}\|\nabla F(\boldsymbol{x};\boldsymbol{a}) - \nabla f_i(\boldsymbol{x})\|^2 \leq \sigma^2, \forall i, \forall \boldsymbol{x}$$

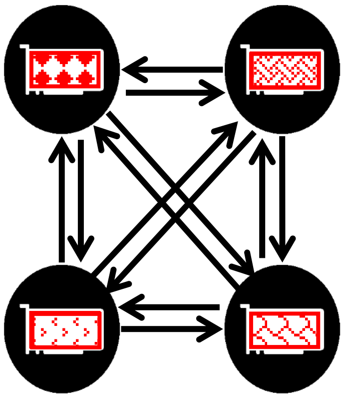
$$\|\nabla f_i(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2 \leq \zeta^2, \forall i, \forall \boldsymbol{x}$$

- **Spectral gap**

$$\rho := \max_{j\geq 2} |\lambda_j(W)|$$

Measure how fast the information can spread across the network

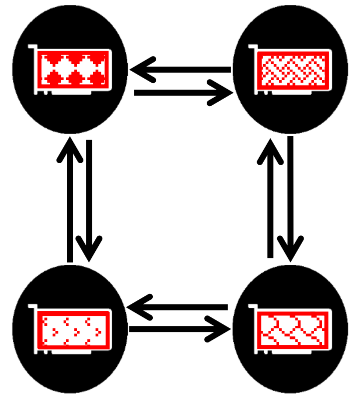
Fully connected network



$$W = \frac{\mathbf{1}\mathbf{1}^\top}{N}$$

$$\rho = 0$$

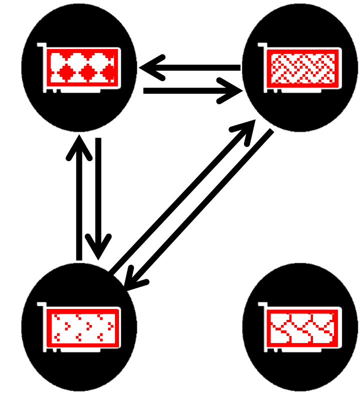
Ring network



$$W = \begin{pmatrix} 1/3 & 1/3 & & & & 1/3 \\ 1/3 & 1/3 & 1/3 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1/3 & 1/3 & 1/3 \\ 1/3 & & & & 1/3 & 1/3 \end{pmatrix}$$

$$\rho \approx \left(1 - \frac{16\pi^2}{3N^2}\right)$$

Disconnected network



$$W = \begin{pmatrix} D & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix}$$

$$\rho = 1$$

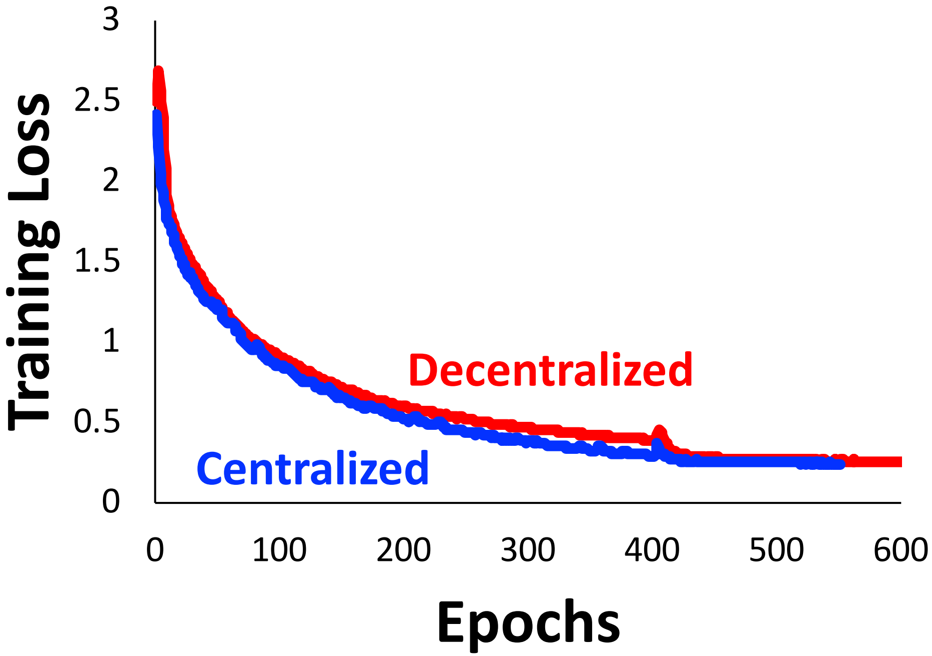
Theorem [DSGD] Choose the learning rate approximately. When T is sufficiently large, we have

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left(\|\nabla f(\overline{\boldsymbol{x}}_t)\|^2\right) \lesssim \frac{1}{T} + \frac{\sigma}{\sqrt{nT}} + \left(\frac{\zeta\rho}{T(1-\rho)}\right)^{\frac{2}{3}}$$

Average of local models

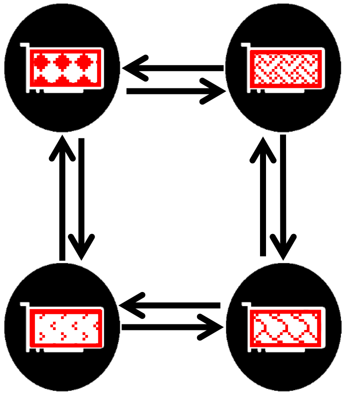
Convergence rate of CSGD

Cost of using decentralized communication (minor)



DECENTRALIZED METHOD

Ring Topology

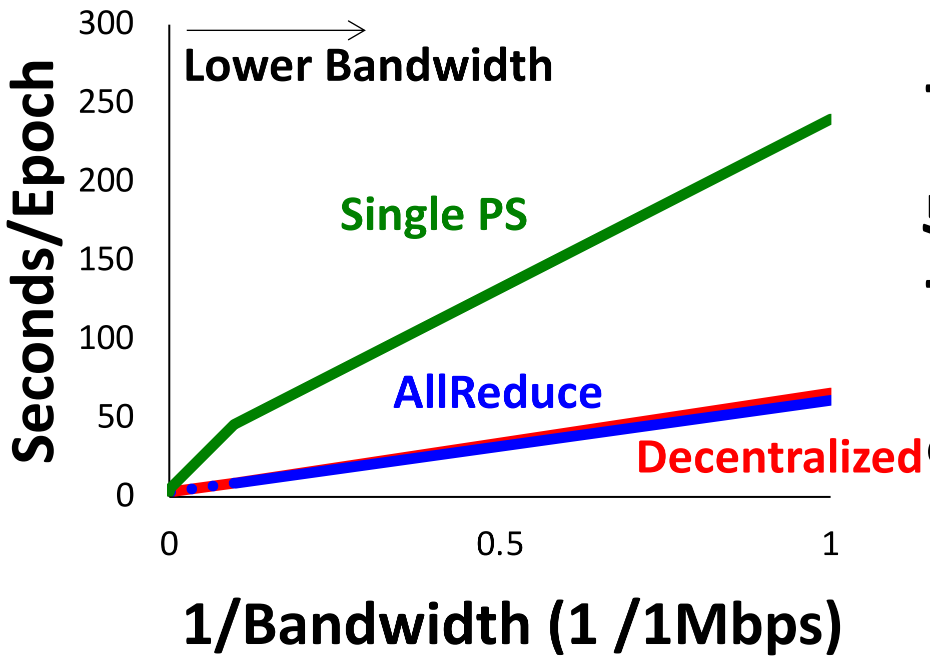


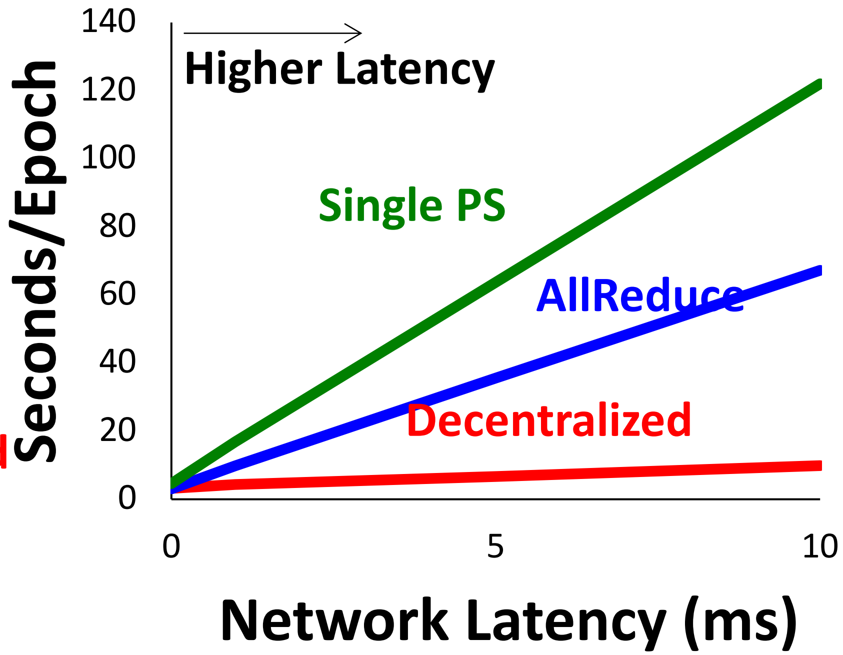
100 GPUs

ResNet

CIFAR10

Centralized includes PS and AllReduce!





Decentralized algorithms **outperform** centralized algorithms for networks with low bandwidth and high latency

Take Away Message

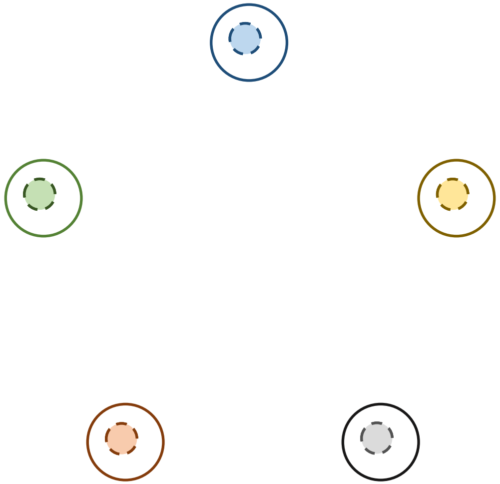
Theoretical view

Decentralized-SGD achieves the same convergence rate as Centralized-PSGD

Practical view

When the network is with high latency, decentralized communication can outperform its centralized counterpart.

Compression + Decentralization

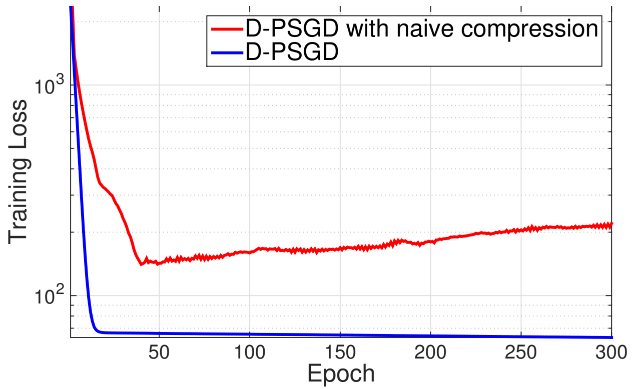


Naïve compression does not work

Can we further reduce the communication cost?

Naïve compression for D-SGD

$$\boldsymbol{x}_{t+1}^{(i)} = \sum_j W_{ij} \mathcal{C}\left(\boldsymbol{x}_t^{(j)}\right) - \gamma \nabla F(\boldsymbol{x}_t^{(i)}; \boldsymbol{a}^{(i)})$$



DCD-SGD

Store a copy of its neighbors' models

$$\hat{\boldsymbol{x}}_{t+1}^{(i)} = \sum_j W_{ij} \boldsymbol{x}_t^{(j)} - \gamma \nabla F(\boldsymbol{x}_t^{(i)}; \boldsymbol{a}^{(i)})$$

$$\boldsymbol{x}_{t+1}^{(i)} = \hat{\boldsymbol{x}}_t^{(i)} + \mathcal{C}\left(\hat{\boldsymbol{x}}_t^{(i)} - \boldsymbol{x}_t^{i}\right)$$

Compress the difference and send to its neighbors

$$\sup_{\boldsymbol{x}} \frac{\mathbb{E}\|\mathcal{C}(\boldsymbol{x}) - \boldsymbol{x}\|^2}{\|\boldsymbol{x}\|^2} \leq \alpha^2$$

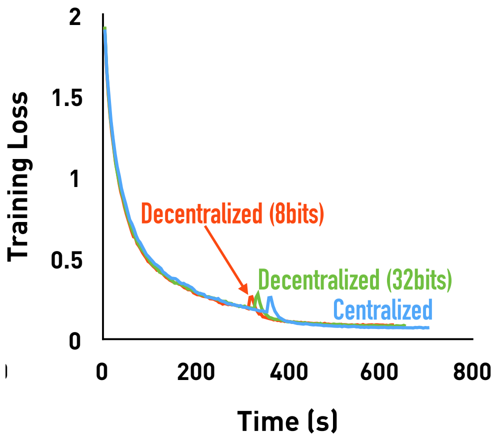
$$\mathbb{E}(\mathcal{C}(\mathbf{x})) = \mathbf{x}$$

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left(\|\nabla f(\overline{\boldsymbol{x}}_t)\|^2\right) \lesssim \frac{1}{T} + \frac{\sigma(1+\alpha^2)}{\sqrt{nT}} + \frac{\zeta^{\frac{2}{3}}(1+\alpha^2)}{T^{\frac{2}{3}}}$$

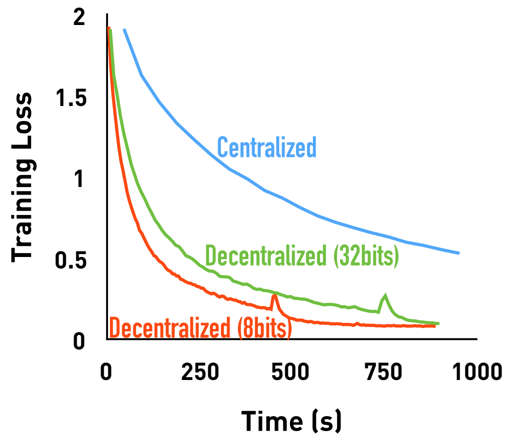
Consistent with D-SGD

[NIPS 2018]

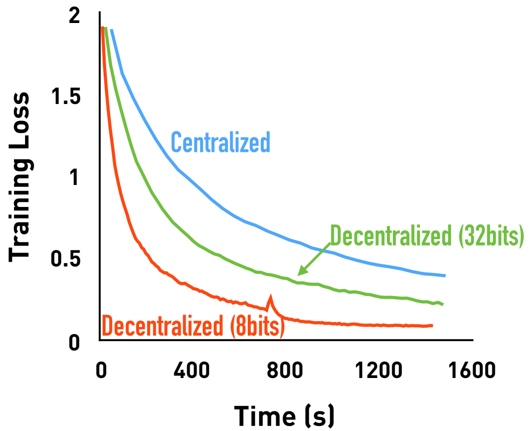
Experiments



(b) Time vs Training Loss Bandwidth = 1.4Gbps, Latency = 0.13ms



(c) Time vs Training Loss Bandwidth = 1.4Gbps, Latency = 20ms



(d) Time vs Training Loss Bandwidth = 5Mbps, Latency = 20ms

Limitation of DCD-SGD

Two Issues of DCD-SGD:

Require $\mathbb{E}(\mathcal{C}(\mathbf{x})) = \mathbf{x}$

Diverges when using 4-bit compression in most cases

Can we fix it by using error compression strategy?

How About

$$X_{t+1} = (X_t - \gamma G_t)W$$
$$= X_t - \gamma G_t + (X_t - \gamma G_t)(W - I)$$

Share this with Error Compensation

One More Thing: DeepSqueeze

DCD-SGD + Error Compensation

Local:

Error Compensation

$$\boldsymbol{v}_{t+1}^{(i)} = \mathcal{C}\left(\boldsymbol{x}_t^{(i)} - \gamma \boldsymbol{g}_t^{(i)} + \boldsymbol{\delta}_t^{(i)}\right)$$

$$\boldsymbol{\delta}_{t+1}^{(i)} = \boldsymbol{v}_{t+1}^{(i)} - \left(\boldsymbol{x}_t^{(i)} - \gamma \boldsymbol{g}_t^{(i)} + \boldsymbol{\delta}_t^{(i)}\right)$$

Communicate:

$$\boldsymbol{x}_{t+1}^{(i)} = \boldsymbol{x}_t^{(i)} - \gamma \boldsymbol{g}_t^{(i)} + \eta \sum_{j \in \mathcal{N}_i} (W_{ij} - I_{ij}) \boldsymbol{v}_{t+1}^{(j)}$$

Control the compression error explicitly

DeepSqueeze V.S. DCD-PSGD

DCD-SGD

$$\sup_{\boldsymbol{x}} \frac{\mathbb{E}\|\mathcal{C}(\boldsymbol{x}) - \boldsymbol{x}\|^2}{\|\boldsymbol{x}\|^2} \leq \alpha^2$$

$$\mathbb{E}(\mathcal{C}(\mathbf{x})) = \mathbf{x}$$

Fails for 4-bit compression

$$\mathcal{O}\left(\frac{1}{T} + \frac{\sigma(1+\alpha^2)}{\sqrt{nT}} + \frac{\zeta^{\frac{2}{3}}(1+\alpha^2)}{T^{\frac{2}{3}}}\right)$$

DeepSqueeze

$$\sup_{\boldsymbol{x}} \frac{\mathbb{E}\|\mathcal{C}(\boldsymbol{x}) - \boldsymbol{x}\|^2}{\|\boldsymbol{x}\|^2} \leq \alpha^2$$

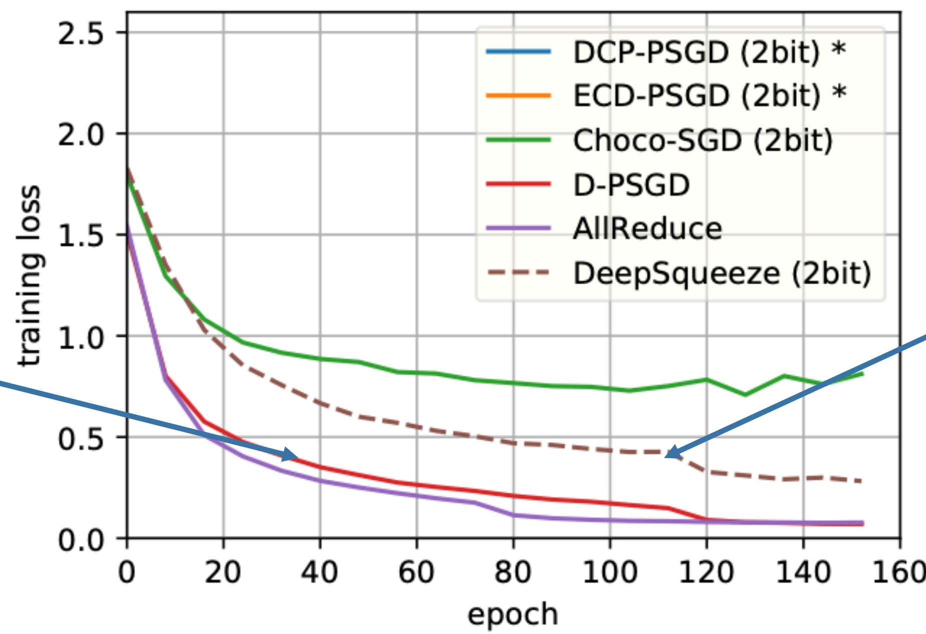
Compression can be biased

Robust to 2-bit compression

$$\mathcal{O}\left(\frac{1}{T} + \frac{\sigma\left(1 + \frac{\alpha^2\sqrt{n}}{\sqrt{T}}\right)}{\sqrt{nT}} + \frac{\zeta^{\frac{2}{3}}(1+\alpha^2)}{T^{\frac{2}{3}}}\right)$$

Experiments

DSGD



DeepSqueeze (save 16x communication cost)

Summary

Foundations and Trends® in Databases

Distributed Learning Systems with First-Order Methods

An Introduction

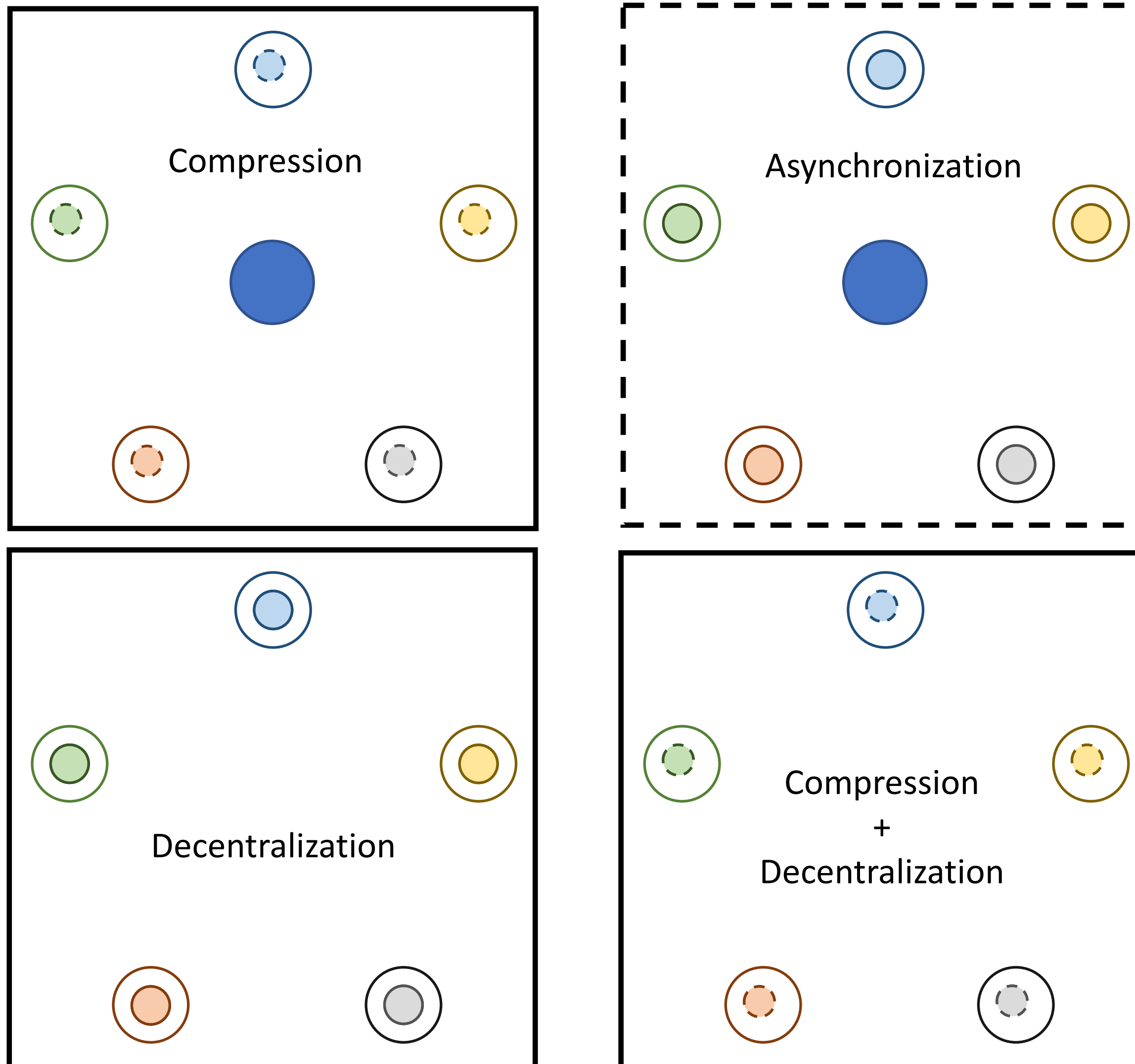
Suggested Citation: Ji Liu and Ce Zhang (2020), "Distributed Learning Systems with First-Order Methods", Foundations and Trends® in Databases: Vol. 9, No. 1, pp 1–97. DOI: 10.1561/1900000062.

Ji Liu
University of Rochester and
Kuaishou Inc., USA
ji.liu.uwisc@gmail.com

Ce Zhang
ETH Zurich, Switzerland
ce.zhang@inf.ethz.ch

Coming Soon.

now
the essence of knowledge
Boston — Delft



Q&A