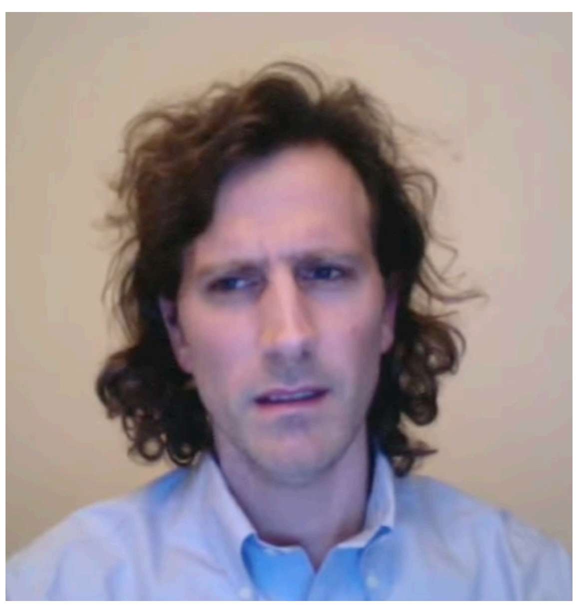
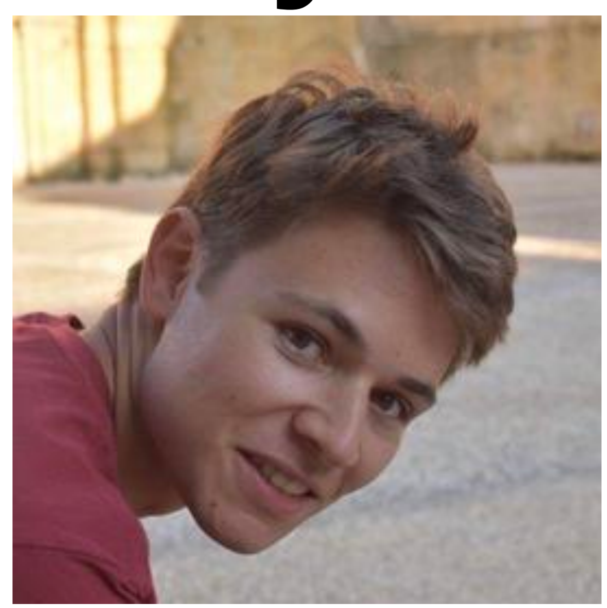
Asynchronous SGD Beats Minibatch SGD Under Arbitrary Delays


Konstantin Mishchenko, Francis Bach, Mathieu Even, Blake Woodworth

Talk plan

1. Why Asynchronous SGD?
2. Overview of known results
3. Motivation and intuition
4. New results
5. Limitations

Problem

$$\min_{x \in \mathbb{R}^d} f(x) = \mathbb{E}[f(x;\xi)]$$

Problem

$$\min_{x \in \mathbb{R}^d} f(x) = \mathbb{E}[f(x;\xi)]$$

Smoothness: $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$

Problem

$$\min_{x \in \mathbb{R}^d} f(x) = \mathbb{E}[f(x;\xi)]$$

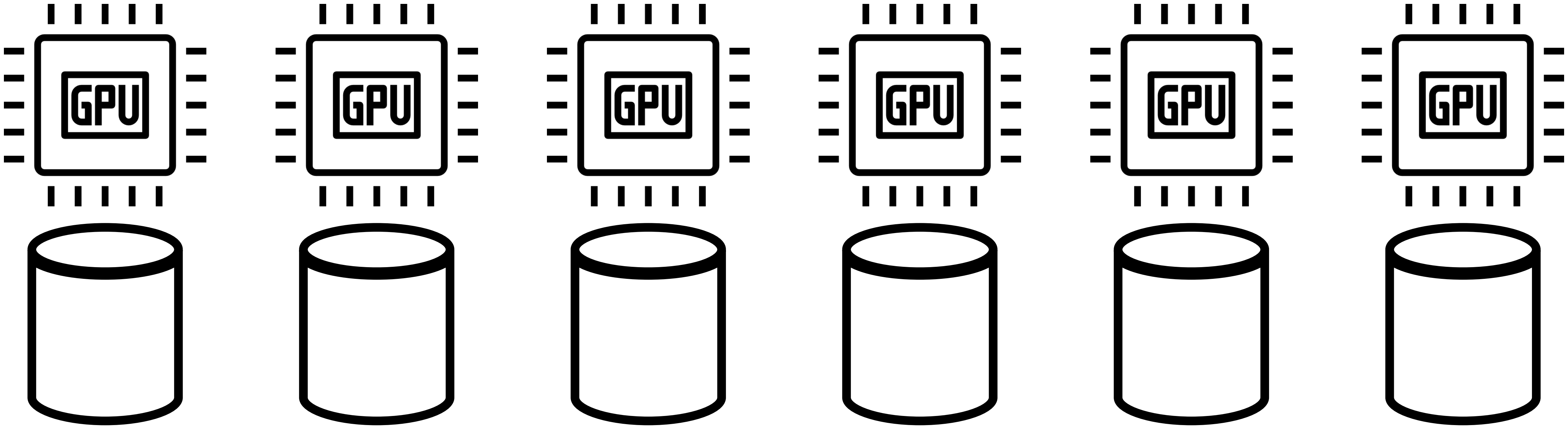
Smoothness: $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$

Variance: $\mathbb{E}[\|\nabla f(x;\xi) - \nabla f(x)\|^2] \leq \sigma^2$

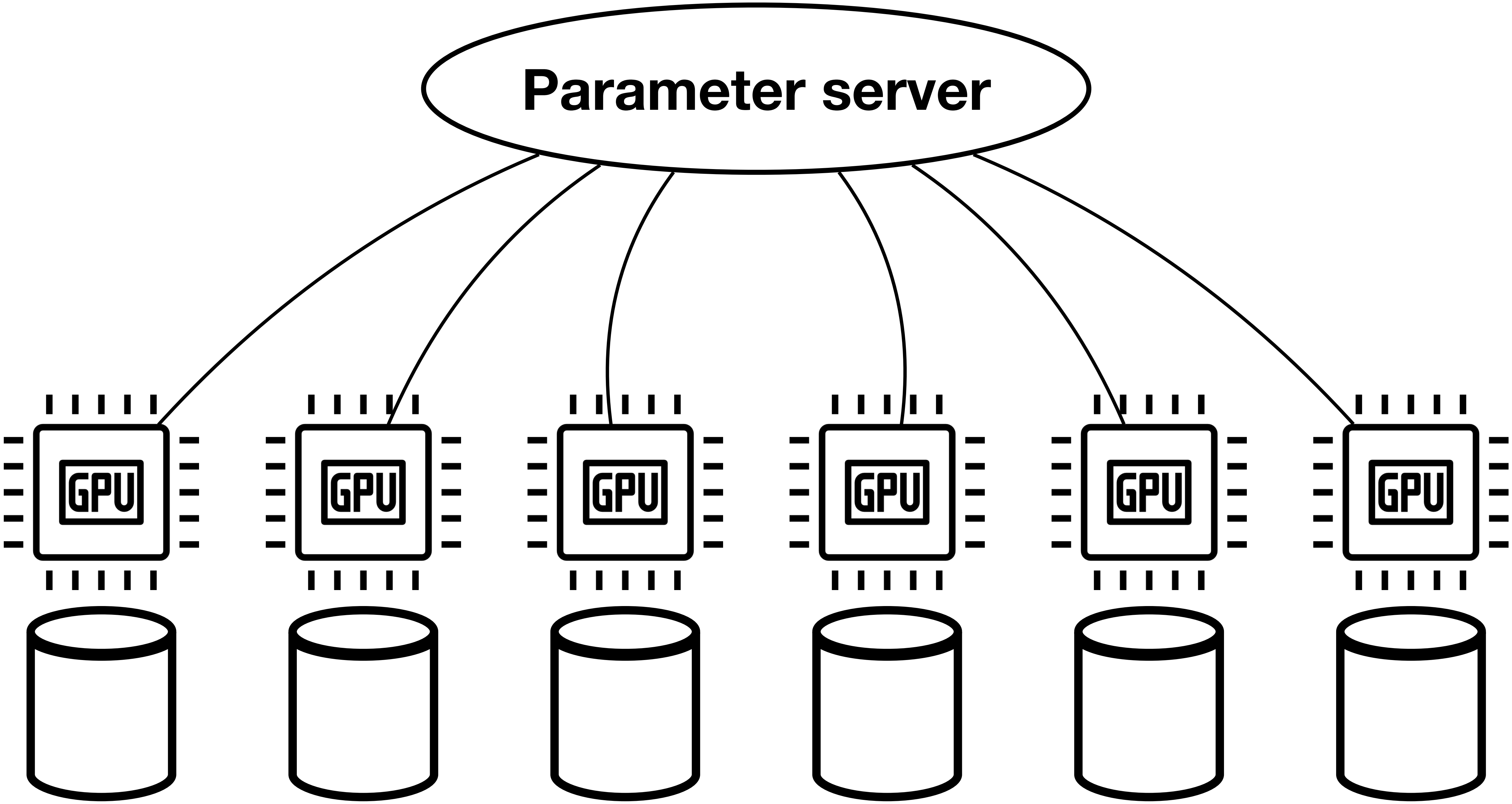
Problem

$$\min_{x \in \mathbb{R}^d} f(x) = \mathbb{E}[f(x; \xi)]$$

Goal: Parallelize on M devices (same data!)

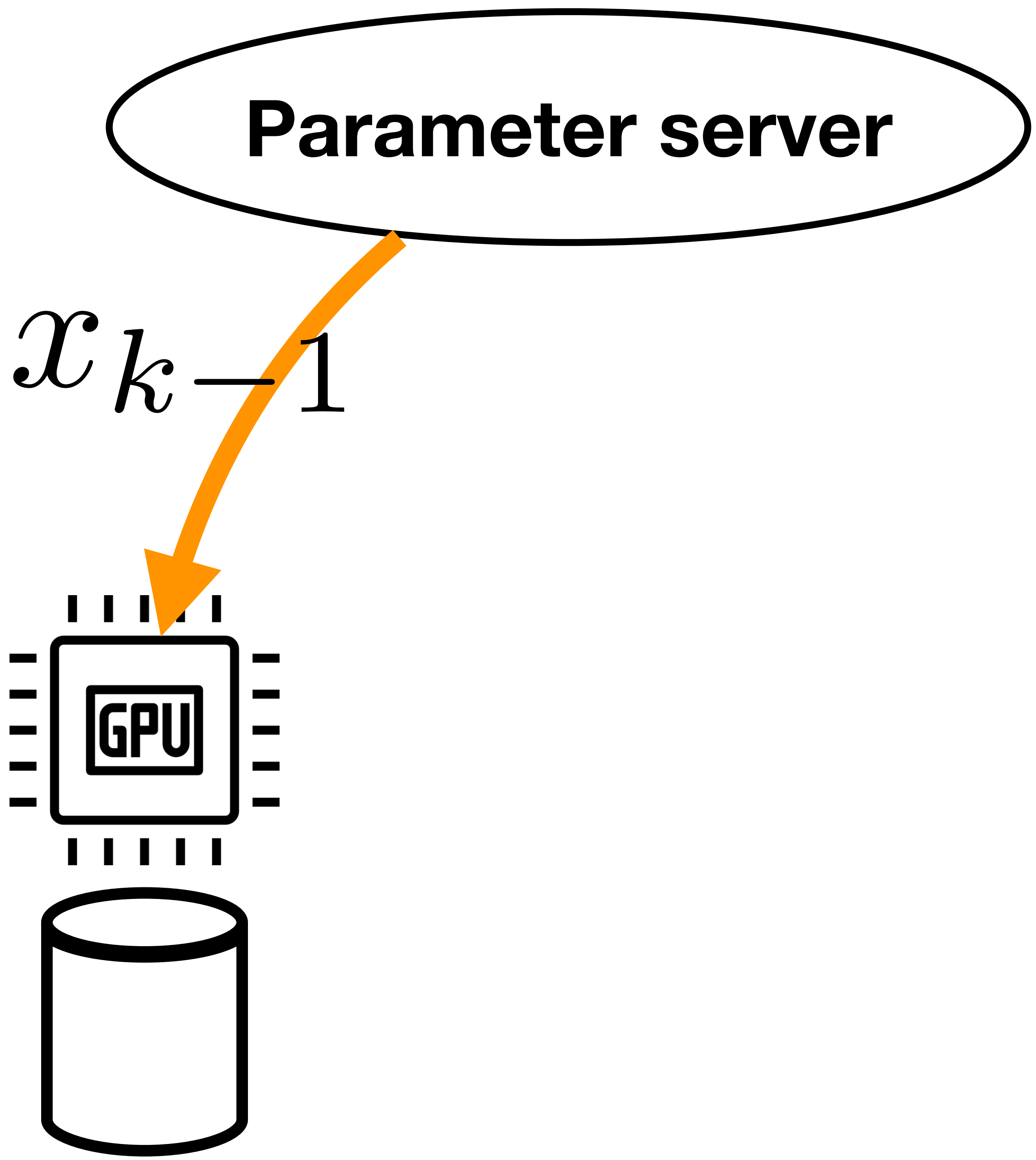


Minibatch SGD

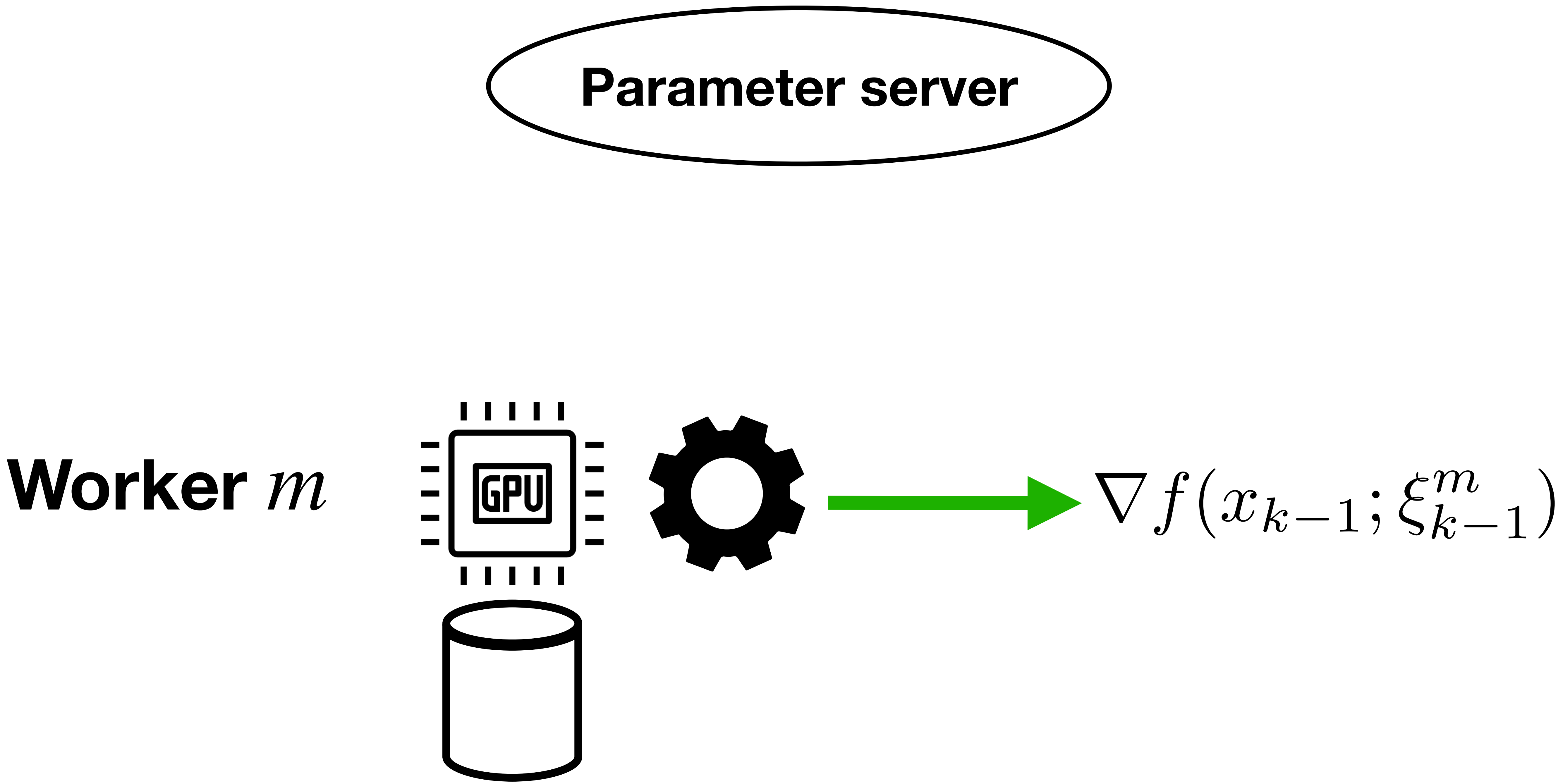


Minibatch SGD

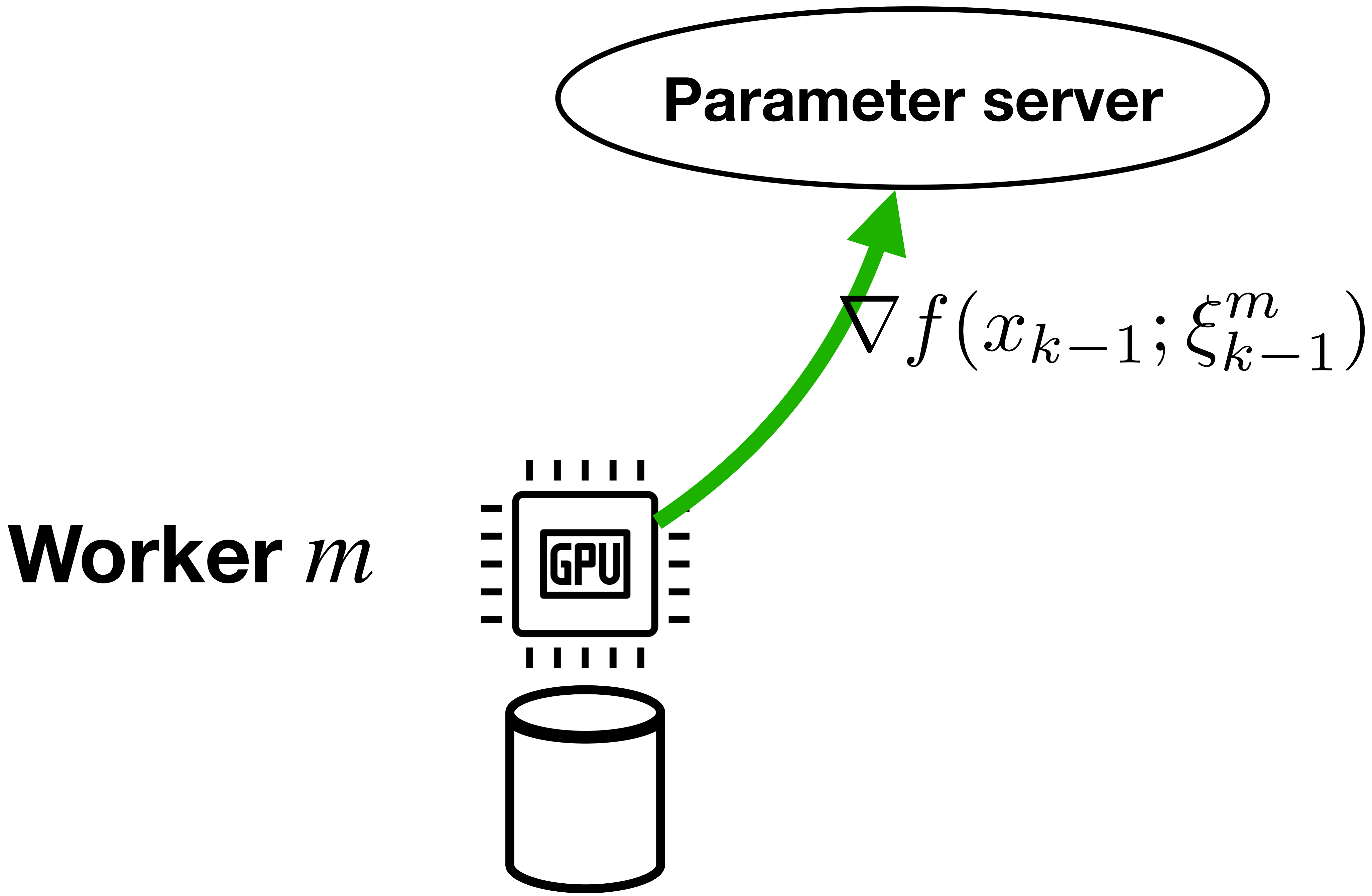
Worker m



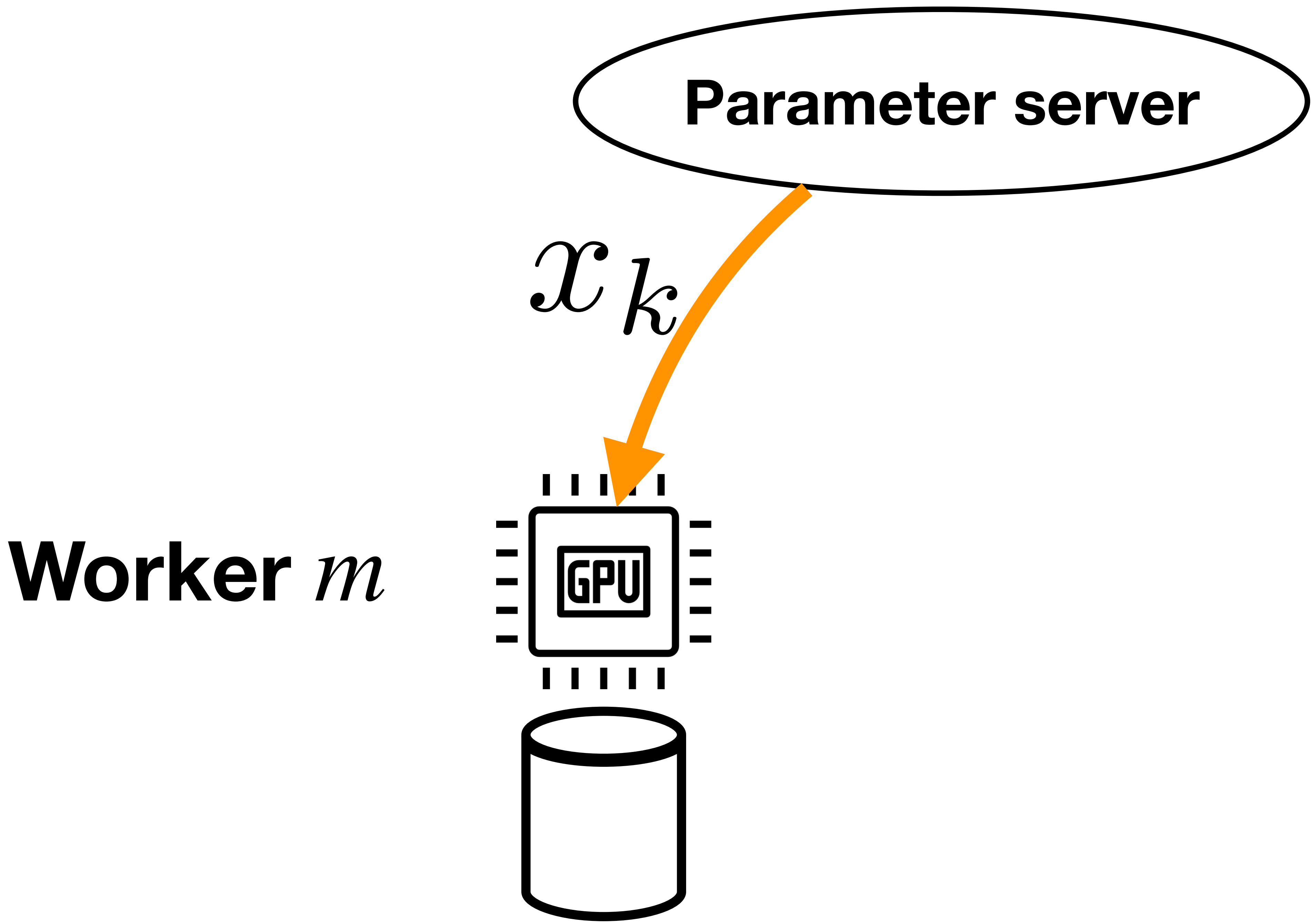
Minibatch SGD



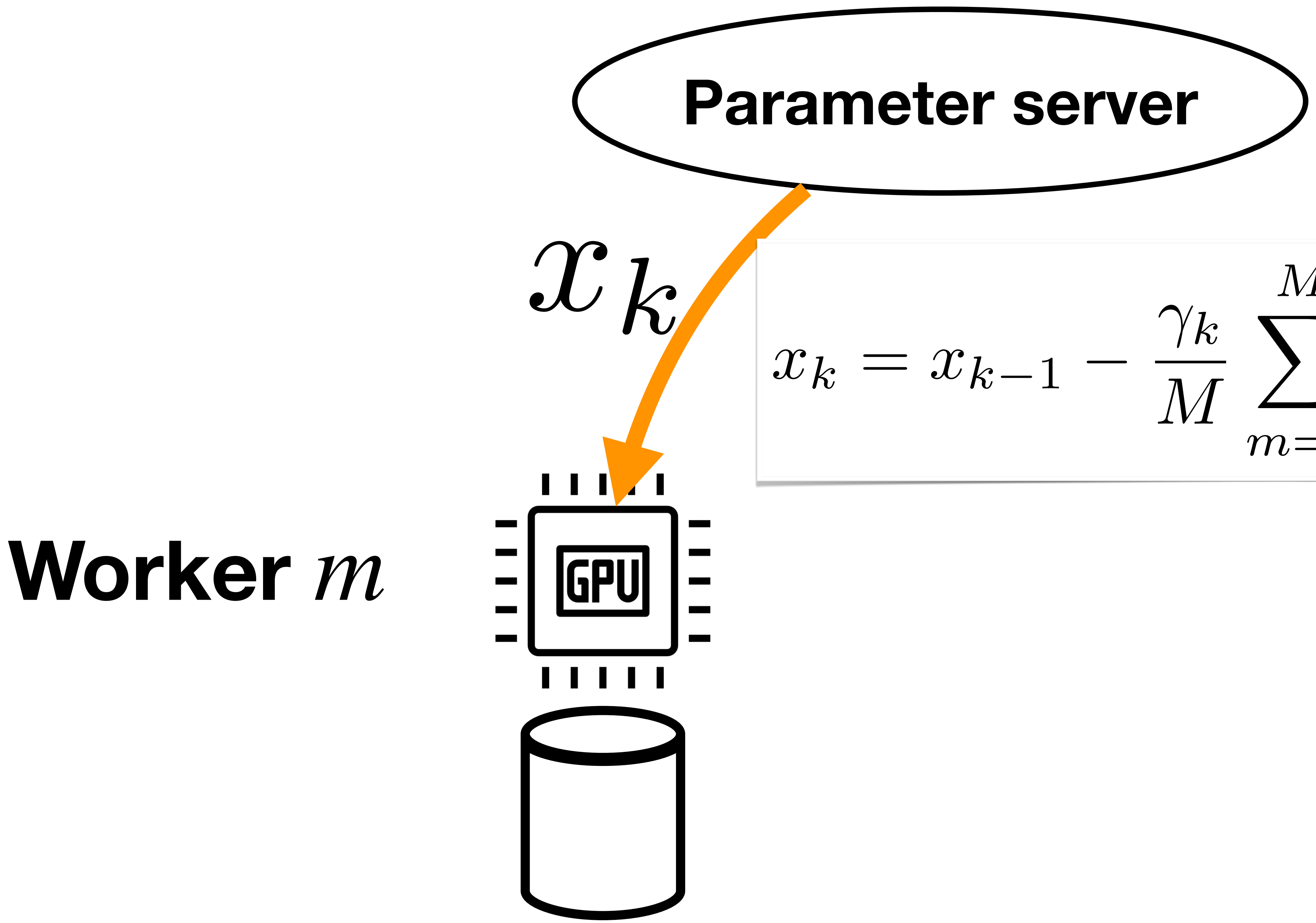
Minibatch SGD



Minibatch SGD

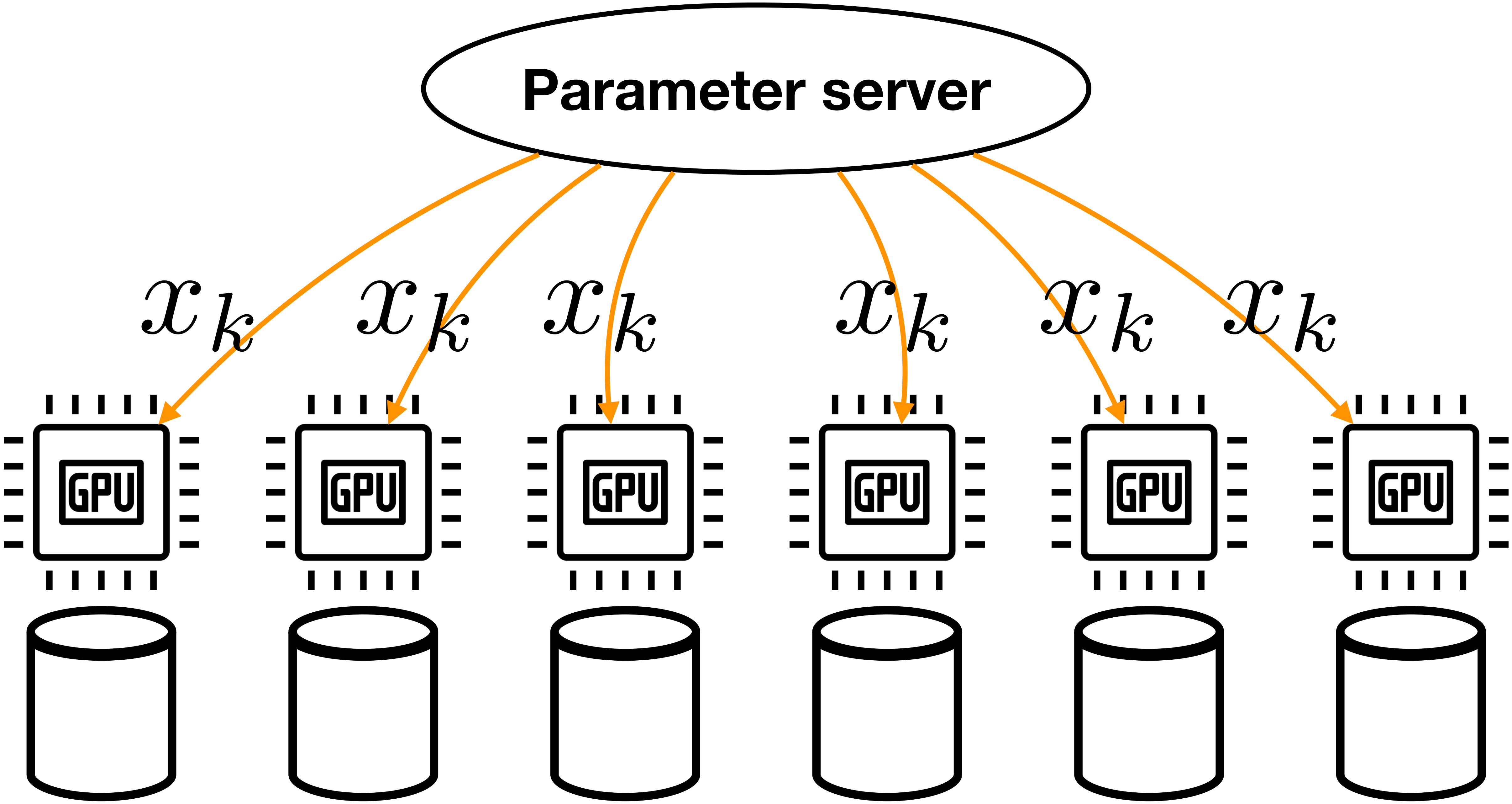


Minibatch SGD

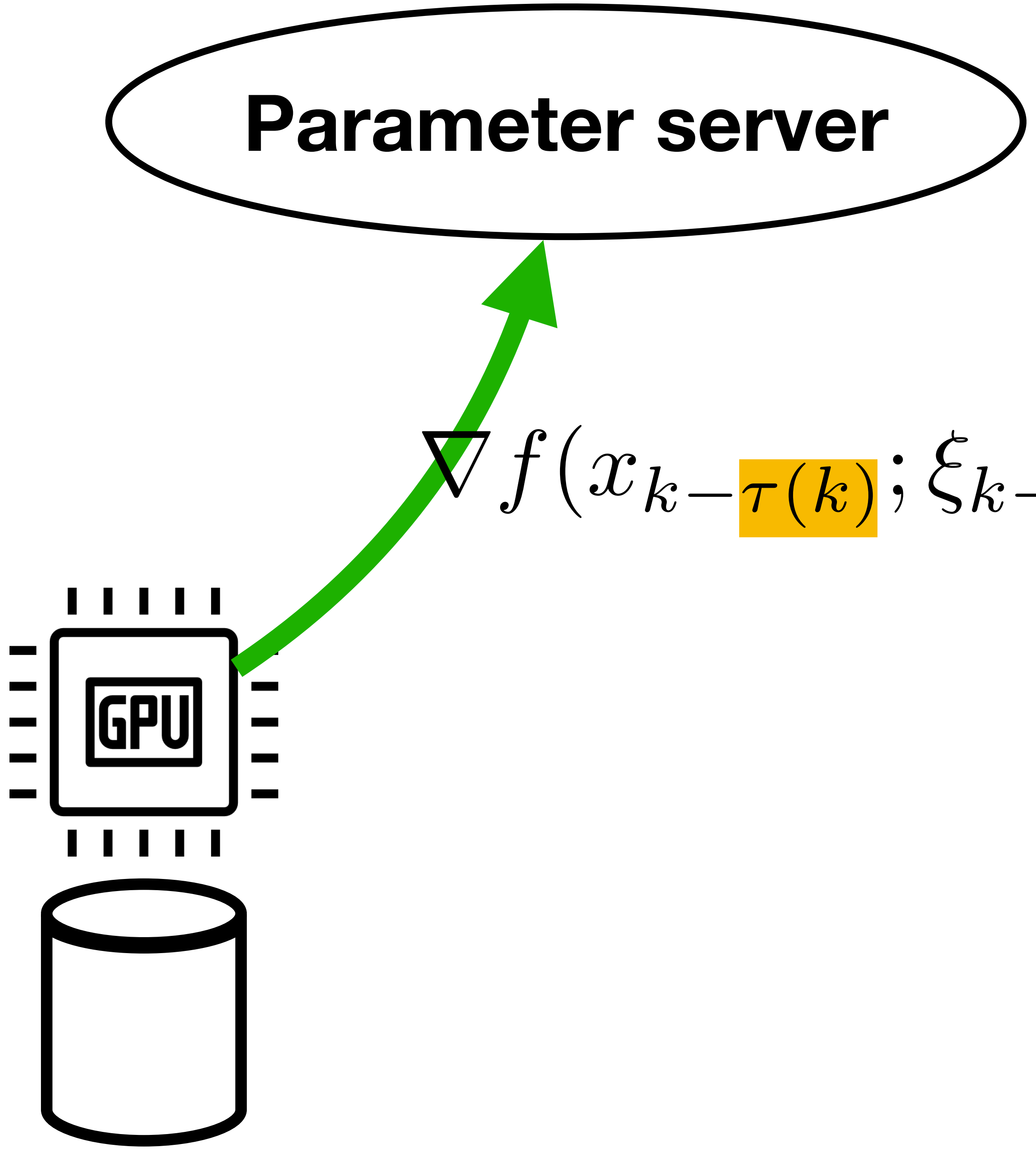


$$x_k = x_{k-1} - \frac{\gamma_k}{M} \sum_{m=1}^{M} \nabla f(x_{k-1}; \xi_{k-1}^m)$$

Minibatch SGD



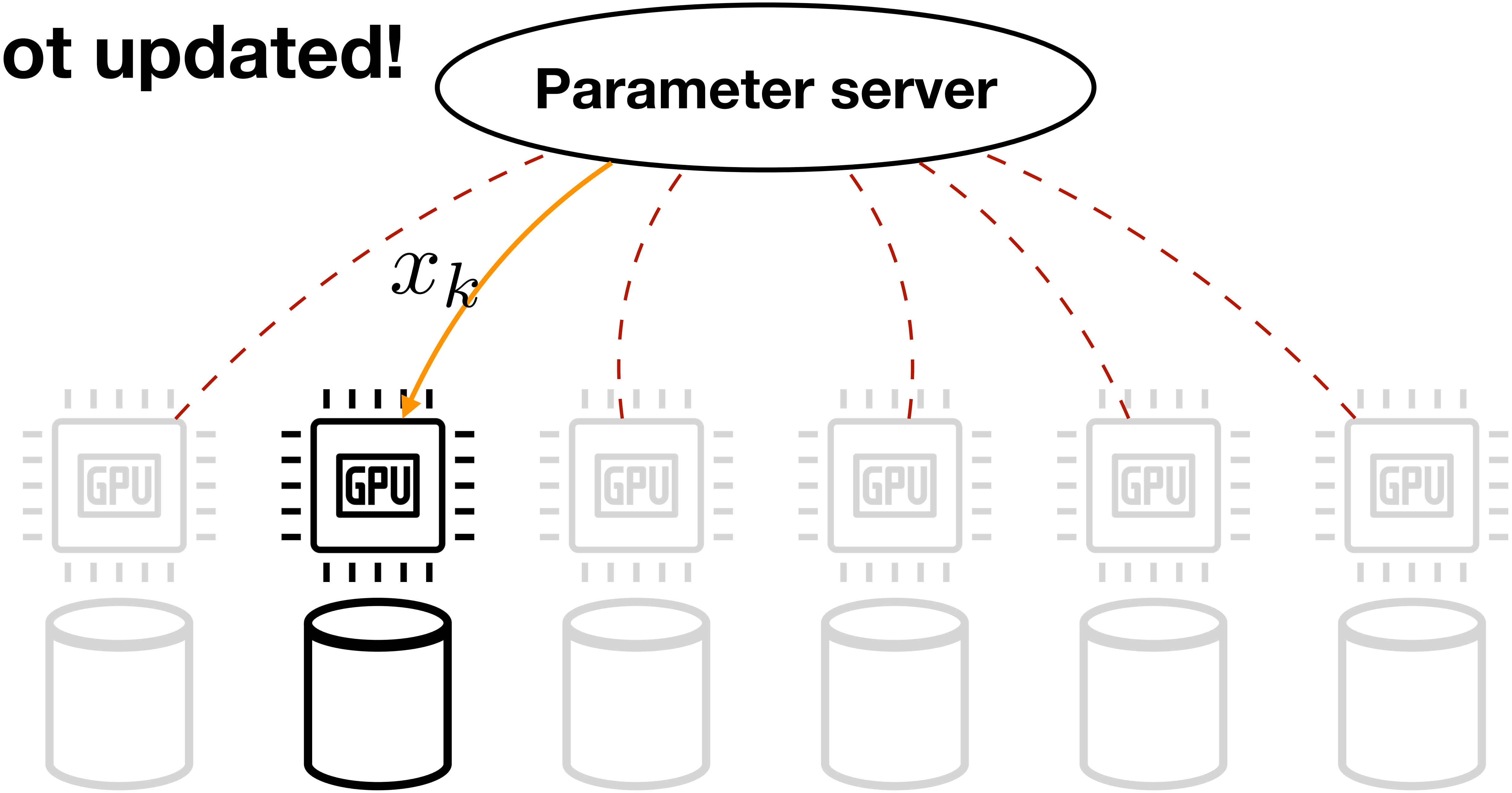
Asynchronous SGD



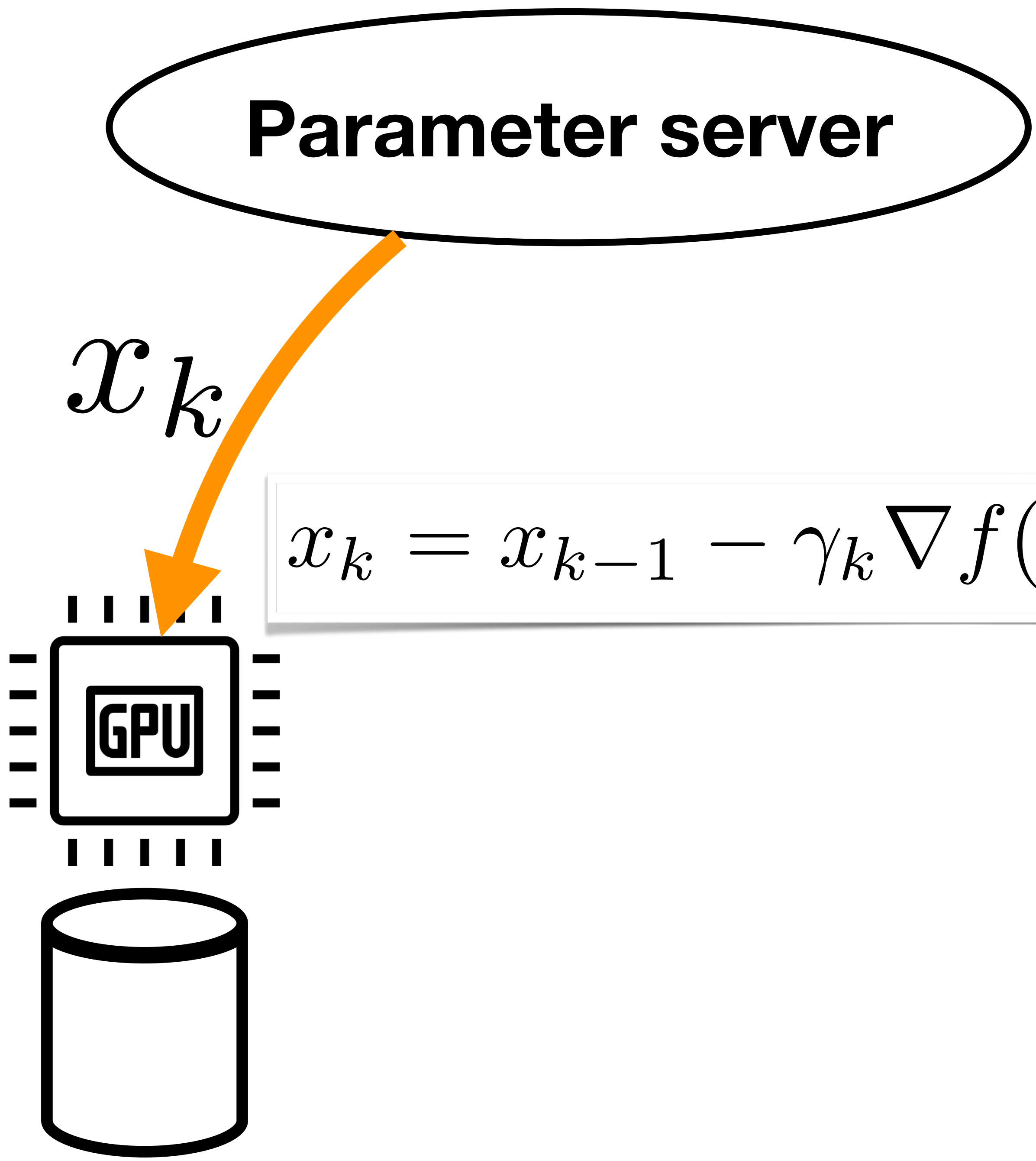
$$\nabla f(x_{k-\tau(k)}; \xi_{k-\tau(k)})$$

Asynchronous SGD

Other workers not updated!



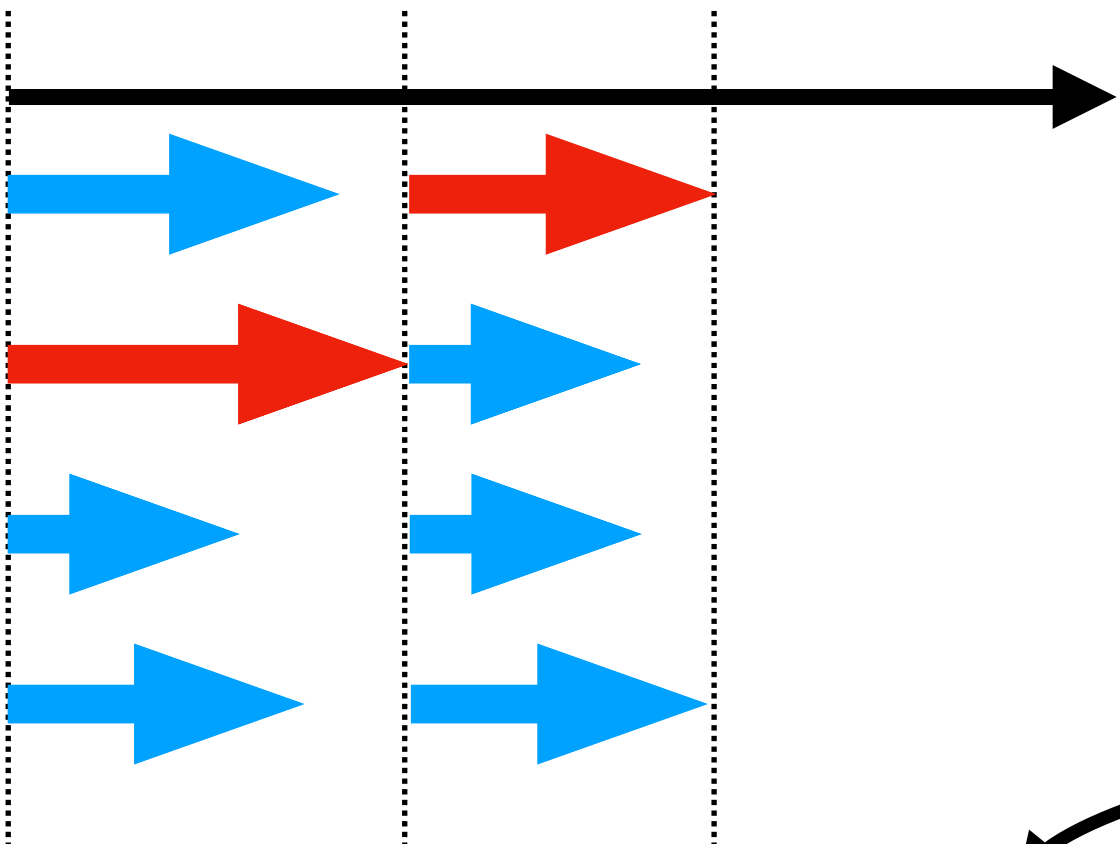
Asynchronous SGD



$$x_k = x_{k-1} - \gamma_k \nabla f(x_{k-\tau(k)}; \xi_{k-\tau(k)})$$

Asynchronous SGD

Minibatch SGD



$$x_k = x_{k-1} - \frac{\gamma_k}{M} \sum_{m=1}^{M} \nabla f(x_{k-1}; \xi_{k-1}^{m})$$

s_m: Seconds per gradient by worker m

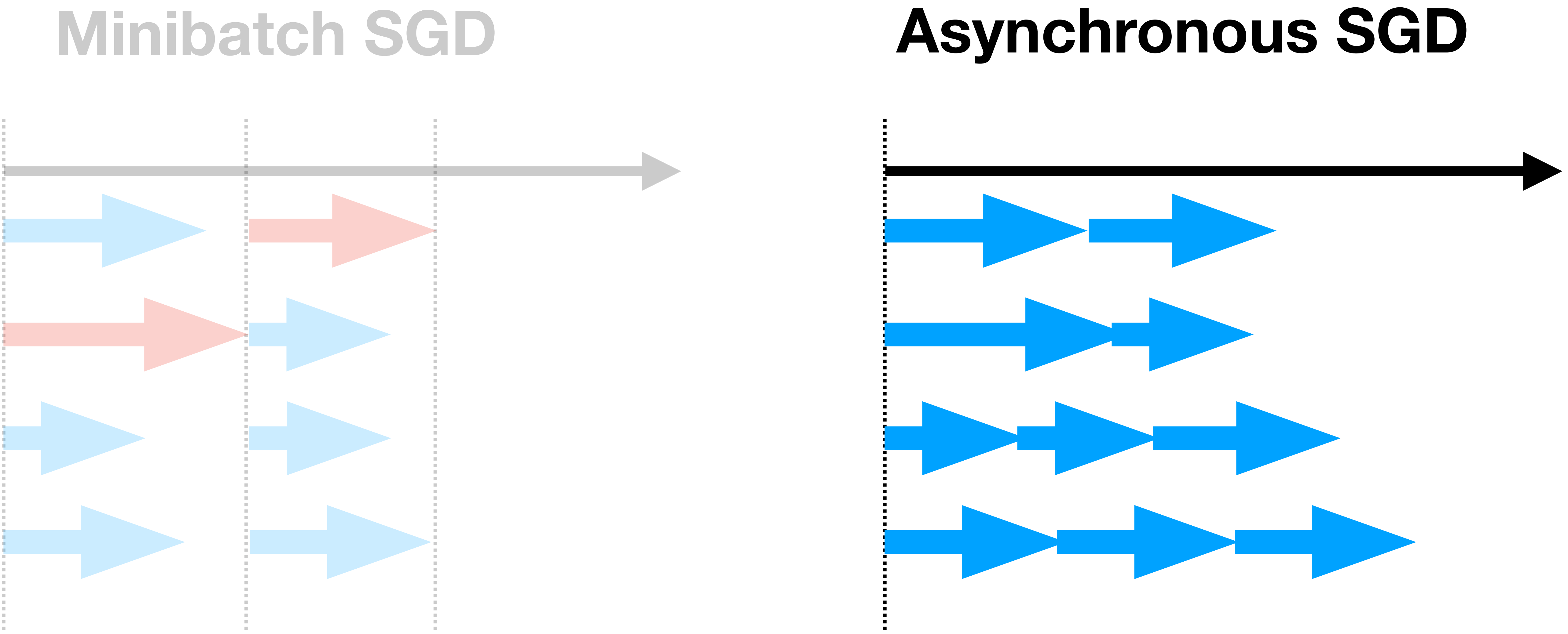
$$s_{\max} = \max_{1 \le m \le M} s_m$$

Time budget

Number of workers

$$\frac{S}{s_{\max}} M \text{ gradients}$$

Asynchronous SGD

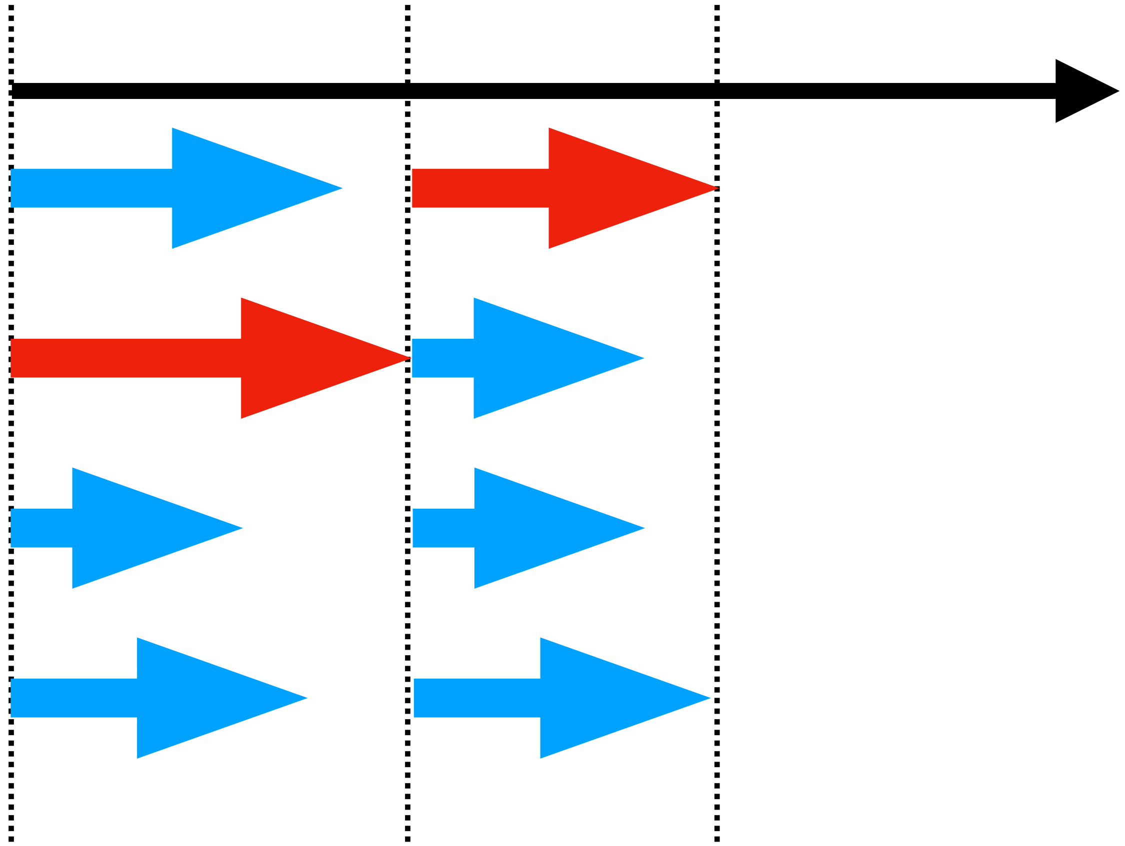


$$x_k = x_{k-1} - \gamma_k \nabla f(x_{k-\tau(k)}; \xi_{k-\tau(k)})$$

$$\sum_{m=1}^{M} \frac{S}{s_m} \text{ gradients}$$

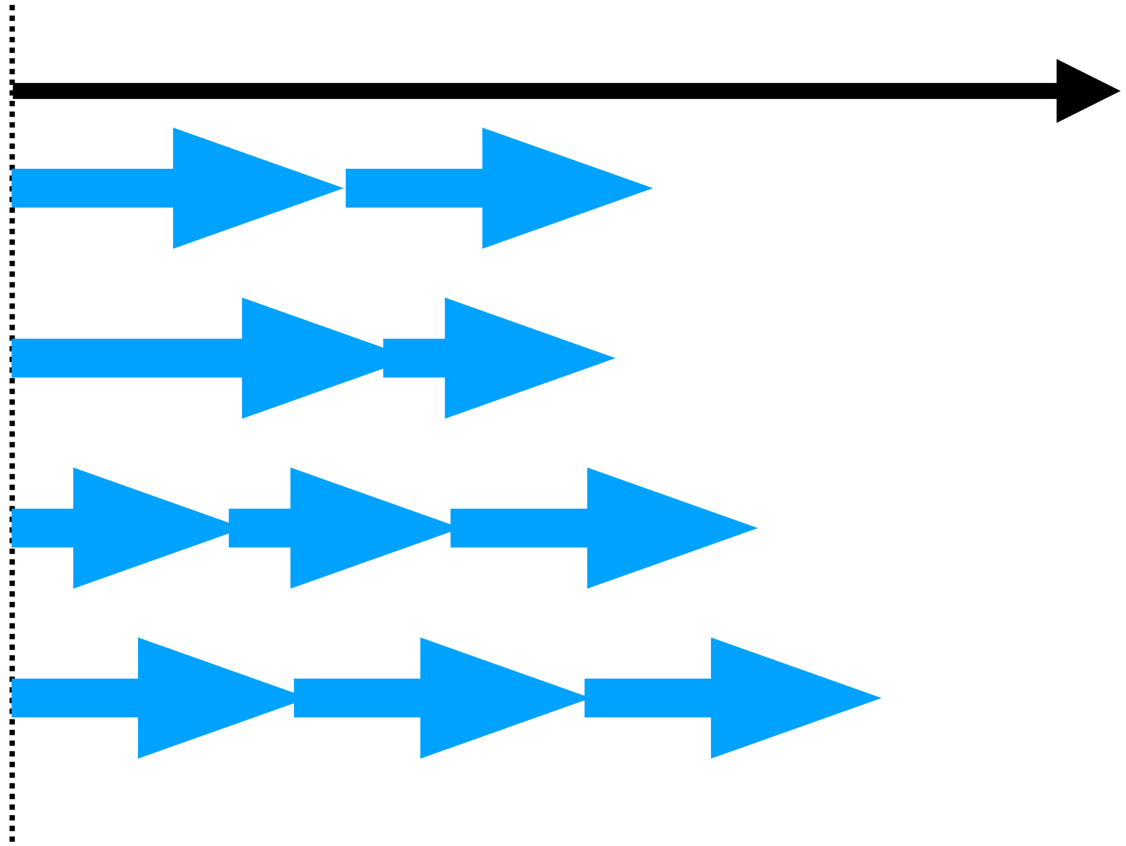
Asynchronous SGD

Minibatch SGD



$$\frac{SM}{s_{\max}} \text{ gradients}$$

Asynchronous SGD



$$\sum_{m=1}^{M} \frac{S}{s_m} \text{ gradients}$$

Asynchronous SGD

Minibatch SGD

$$\frac{SM}{s_{\max}} \text{ gradients}$$

Asynchronous SGD

$$\sum_{m=1}^{M} \frac{S}{s_m} \text{ gradients}$$

Improvement: $\frac{1}{M} \sum_{m=1}^{M} \frac{s_{\max}}{s_m}$

Asynchronous SGD

Motivation:

1. **Heterogeneous cluster**

Asynchronous SGD

Motivation:

1. **Heterogeneous cluster**
2. **Unstable network/devices**

Asynchronous SGD

Motivation:

1. **Heterogeneous cluster**
2. **Unstable network/devices**
3. **Random computation time**

Asynchronous SGD

Motivation:

1. **Heterogeneous cluster**
2. **Unstable network/devices**
3. **Random computation time**
4. **Real decentralization**

Asynchronous SGD

Motivation:

1. **Heterogeneous cluster**
2. **Unstable network/devices**
3. **Random computation time**
4. **Real decentralization**

The drawback: delays

Is it not solved?

State of the literature:

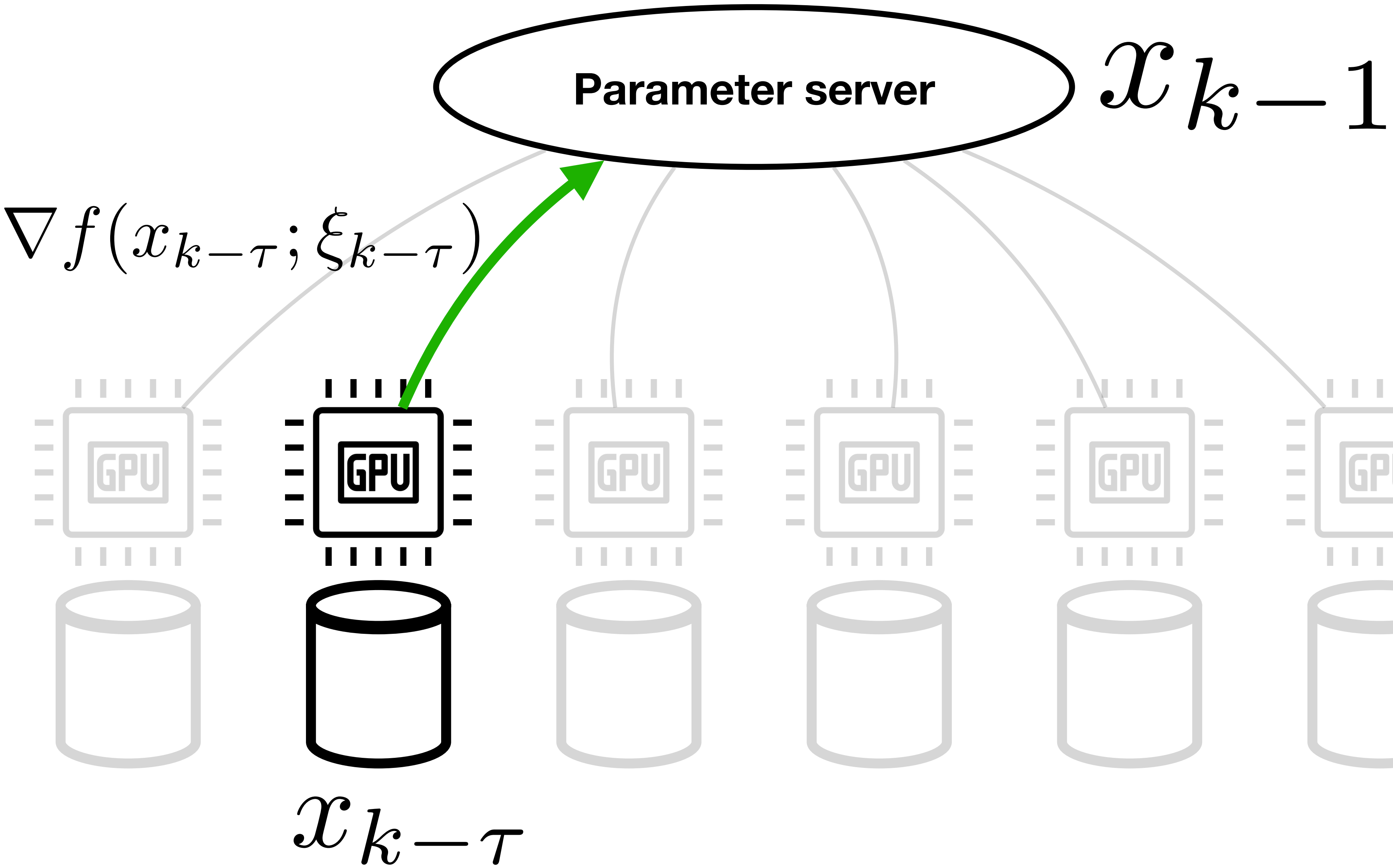
1. **Tight bounds for constant delays**

Tight, but in a toy setting

Is it not solved?

State of the literature:

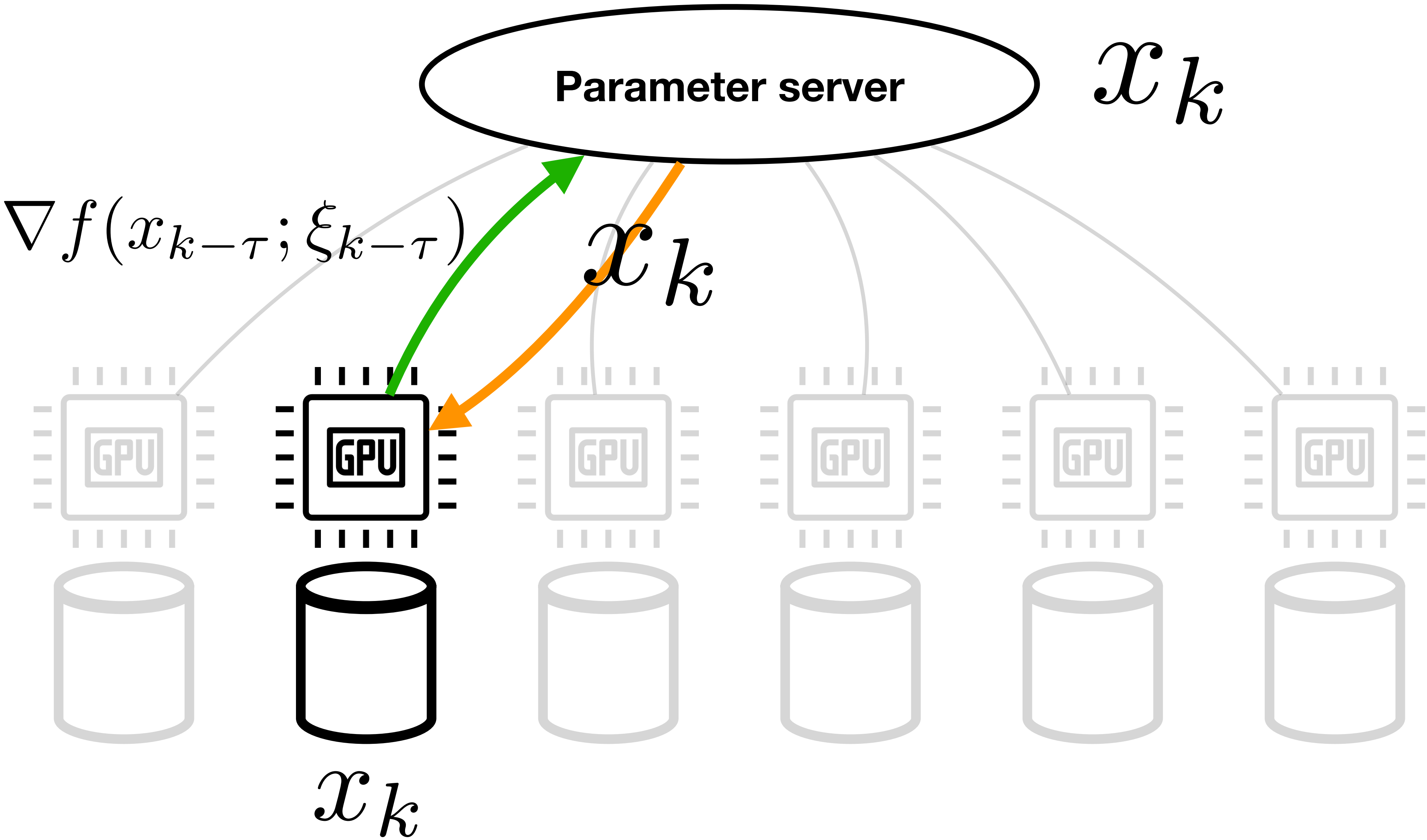
1. **Tight bounds for constant delays**



Is it not solved?

State of the literature:

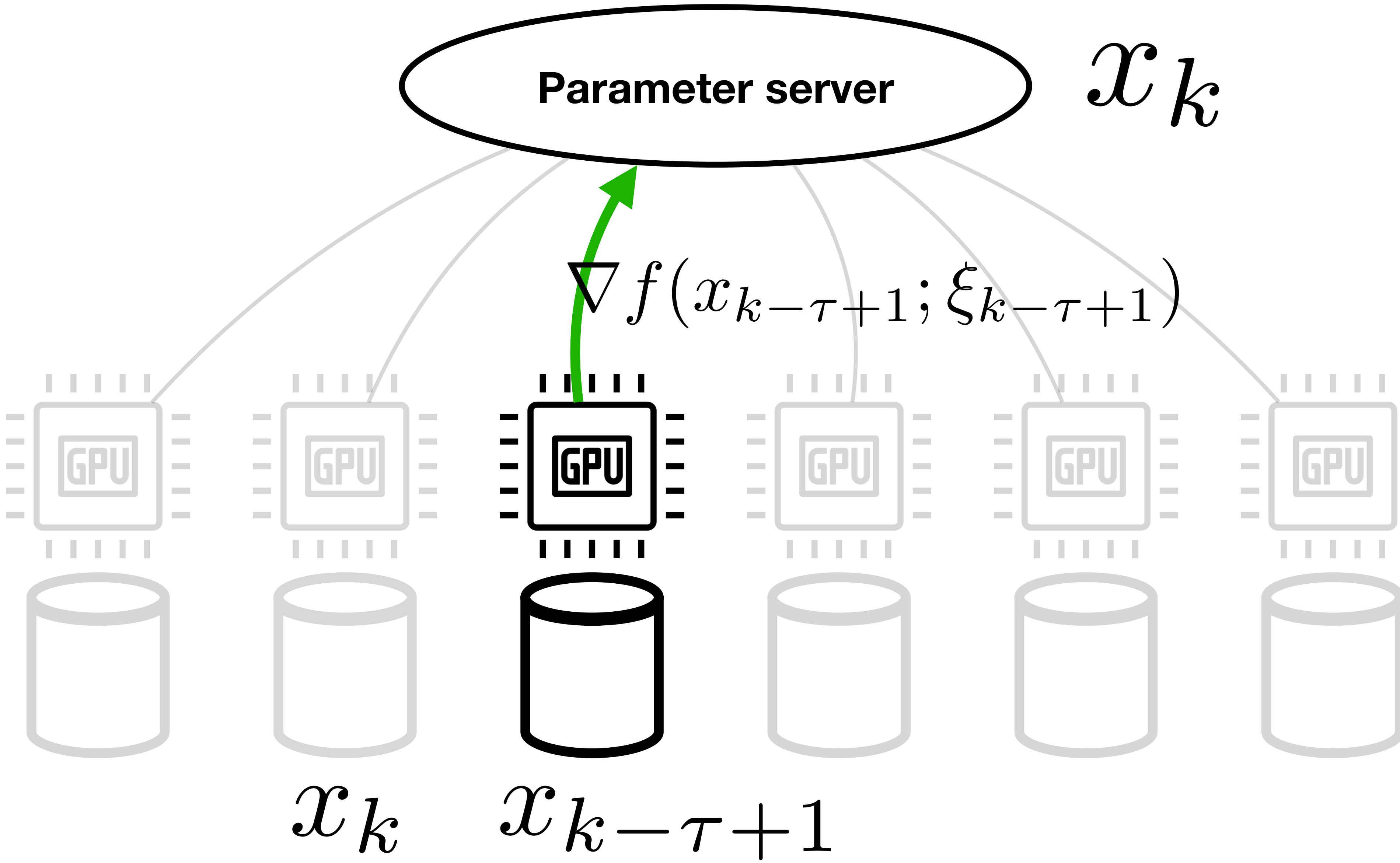
1. **Tight bounds for constant delays**



Is it not solved?

State of the literature:

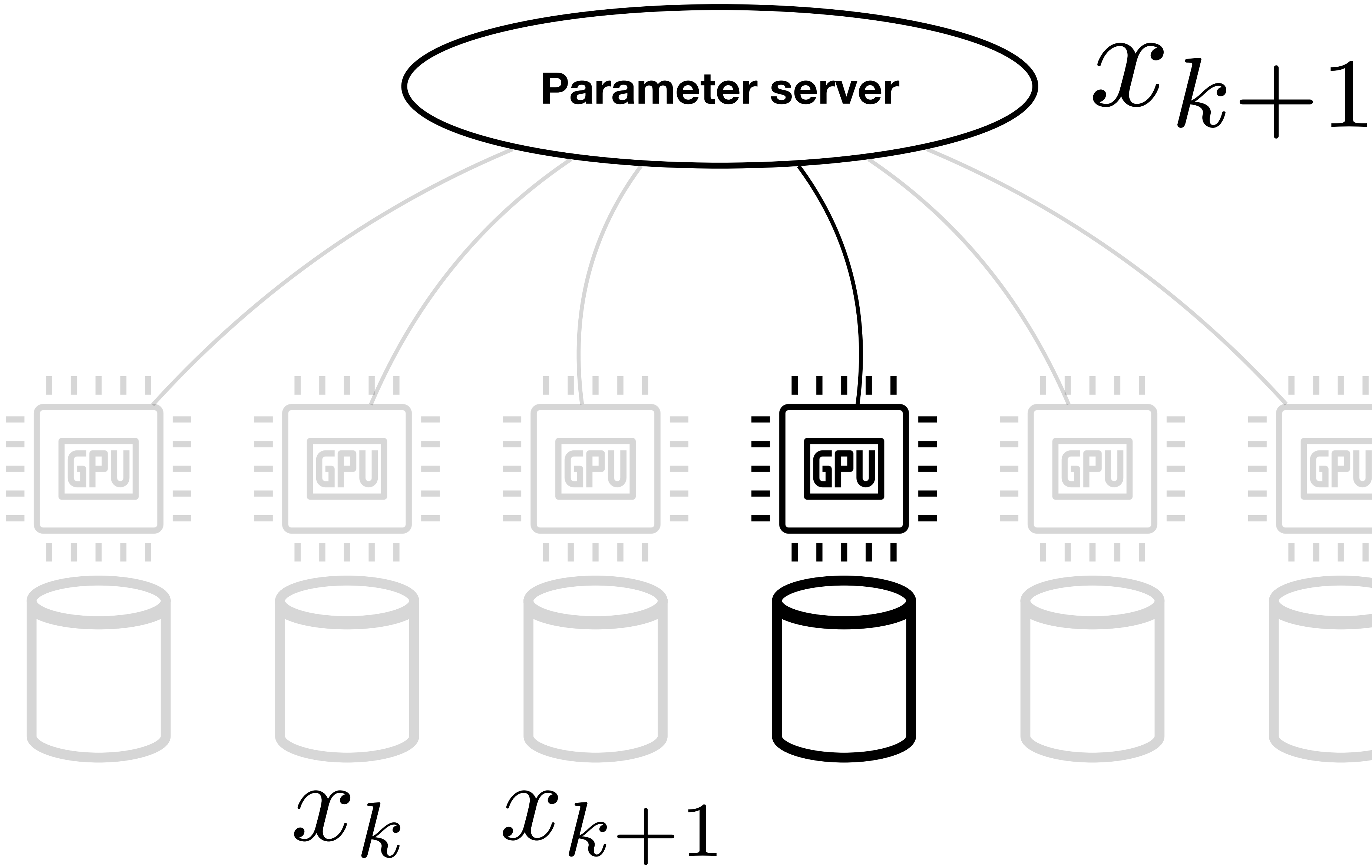
1. **Tight bounds for constant delays**



Is it not solved?

State of the literature:

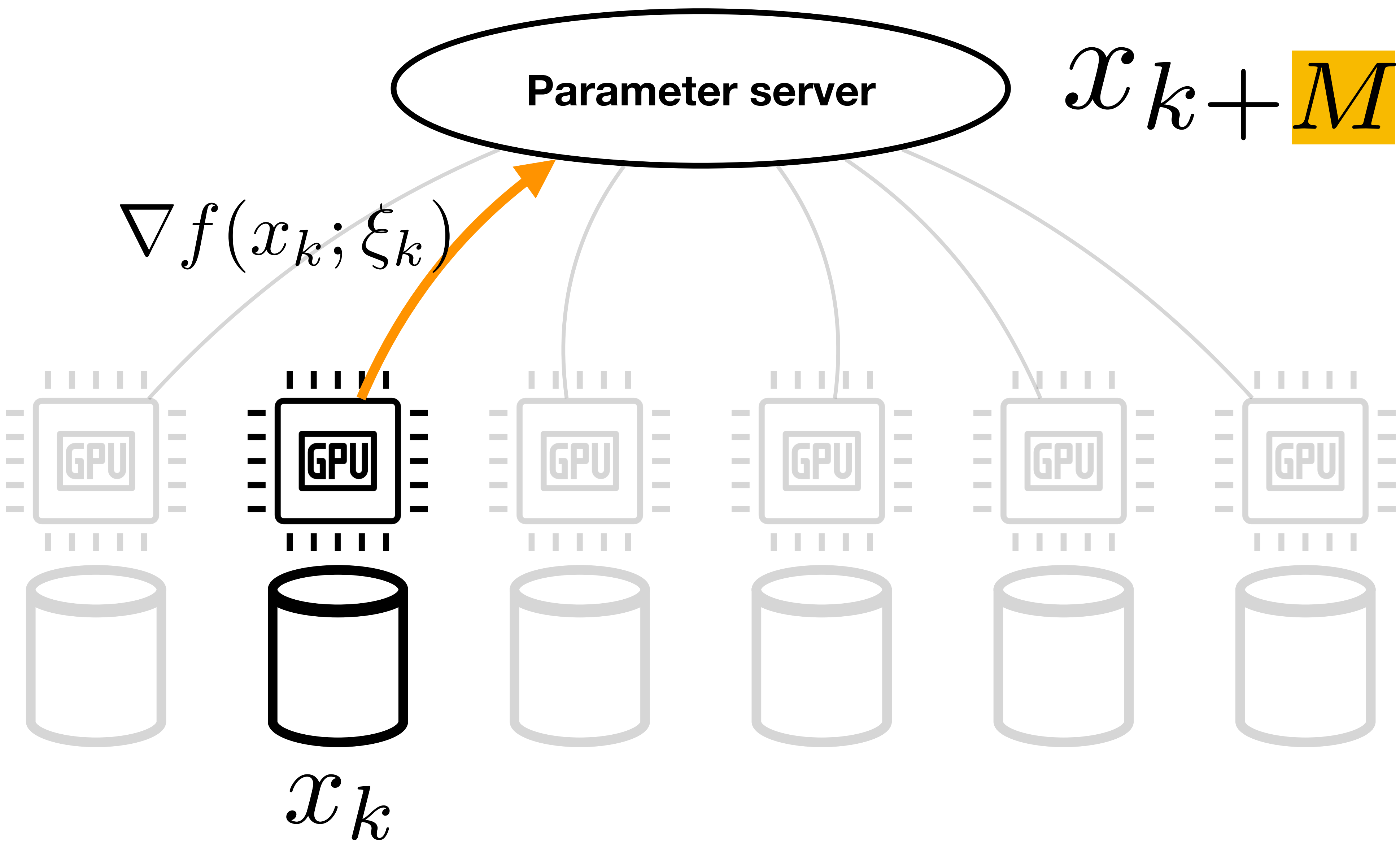
1. **Tight bounds for constant delays**



Is it not solved?

State of the literature:

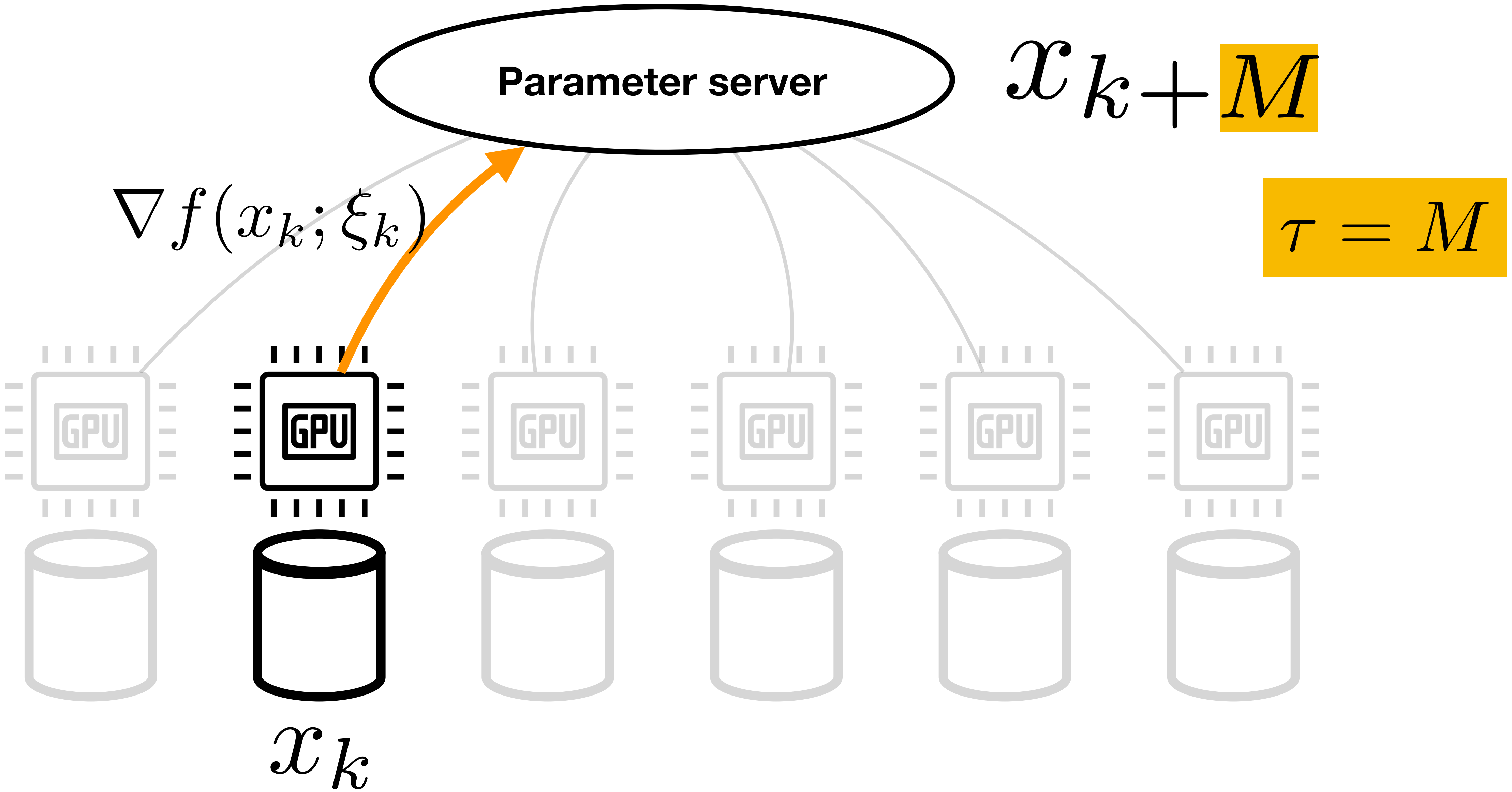
1. **Tight bounds for constant delays**



Is it not solved?

State of the literature:

1. **Tight bounds for constant delays**



Is it not solved?

State of the literature:

1. **Tight bounds for constant delays**
2. **General rates depend on the longest delay**

$$\mathbb{E}[f(x_K) - f_*] = \mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{\tau}{K}\right)$$

Is it not solved?

State of the literature:

1. **Tight bounds for constant delays**
2. **General rates depend on the longest delay**

$$\mathbb{E}[f(x_K) - f_*] = \mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{\tau}{K}\right)$$

$$\tau = \max_{k \le K} \tau(k)$$

Is it not solved?

State of the literature:

1. **Tight bounds for constant delays**
2. **General rates depend on the longest delay**

$$\mathbb{E}[f(x_K) - f_*] = \mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{\tau}{K}\right)$$

$$M \leq \tau = \max_{k \leq K} \tau(k)$$

Is it not solved?

State of the literature:

1. Tight bounds for constant delays
2. General rates depend on the longest delay
3. Assumptions sometimes don’t match those of Minibatch SGD

Is it not solved?

State of the literature:

1. Tight bounds for constant delays
2. General rates depend on the longest delay
3. Assumptions sometimes don't match those of Minibatch SGD
4. No provable speed-up vs. Minibatch SGD

Is it not solved?

A Tight Convergence Analysis for Stochastic Gradient Descent with Delayed Updates

Yossi Arjevani, Ohad Shamir
Weizmann Institute of Science
Rehovot 7610001, Israel
{yossi.arjevani,ohad.shamir}@weizmann.ac.il

Nathan Srebro
TTI Chicago
Chicago, IL 60637
nati@ttic.edu

Abstract

We provide tight finite-time convergence bounds for gradient descent and stochastic gradient descent on quadratic functions, when the gradients are delayed and reflect iterates from τ rounds ago. First, we show that without stochastic noise, delays strongly affect the attainable optimization error: In fact, the error can be as bad as non-delayed gradient descent ran on only $1/\tau$ of the gradients. In sharp contrast, we quantify how stochastic noise makes the effect of delays negligible, improving on previous work which only showed this phenomenon asymptotically or for much smaller delays. Also, in the context of distributed optimization, the results indicate that the performance of gradient descent with delays is competitive with synchronous approaches such as mini-batching. Our results are based on a novel technique for analyzing convergence of optimization algorithms using generating functions.

Is it not solved?

A **Tight** Convergence Analysis for Stochastic Gradient Descent with Delayed Updates

Yossi Arjevani, Ohad Shamir
Weizmann Institute of Science
Rehovot 7610001, Israel
{yossi.arjevani,ohad.shamir}@weizmann.ac.il

Nathan Srebro
TTI Chicago
Chicago, IL 60637
nati@ttic.edu

Abstract

We provide tight finite-time convergence bounds for gradient descent and stochastic gradient descent on quadratic functions, when the gradients are delayed and reflect iterates from τ rounds ago. First, we show that without stochastic noise, delays strongly affect the attainable optimization error: In fact, the **error can be as bad as non-delayed gradient descent ran on only $1/\tau$ of the gradients. In sharp contrast,** we quantify how stochastic noise makes the effect of delays negligible, improving on previous work which only showed this phenomenon asymptotically or for much smaller delays. Also, in the context of distributed optimization, the results indicate that the performance of gradient descent with delays is competitive with synchronous approaches such as mini-batching. Our results are based on a novel technique for analyzing convergence of optimization algorithms using generating functions.

Is it not solved? (notation)

$$\min_{x \in \mathbb{R}^d} f(x) = \mathbb{E}[f(x;\xi)]$$

Smoothness: $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$

Variance: $\mathbb{E}[\|\nabla f(x;\xi) - \nabla f(x)\|^2] \leq \sigma^2$

Is it not solved?

A Tight Convergence Analysis for Stochastic Gradient Descent with Delayed Updates

Yossi Arjevani, Ohad Shamir
Weizmann Institute of Science
Rehovot 7610001, Israel
{yossi.arjevani,ohad.shamir}@weizmann.ac.il

Nathan Srebro
TTI Chicago
Chicago, IL 60637
nati@ttic.edu

Theorem. Let f be L-smooth and convex, then Asynchronous SGD with constant delay τ converges as

$$\mathbb{E}[f(x_K) - f_*] = \mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{\tau}{K}\right)$$

Is it not solved?

A Tight Convergence Analysis for Stochastic Gradient Descent with Delayed Updates

Yossi Arjevani, Ohad Shamir
Weizmann Institute of Science
Rehovot 7610001, Israel
{yossi.arjevani,ohad.shamir}@weizmann.ac.il

Nathan Srebro
TTI Chicago
Chicago, IL 60637
nati@ttic.edu

Theorem. Let f be L-smooth and convex, then Asynchronous SGD with constant delay τ converges as

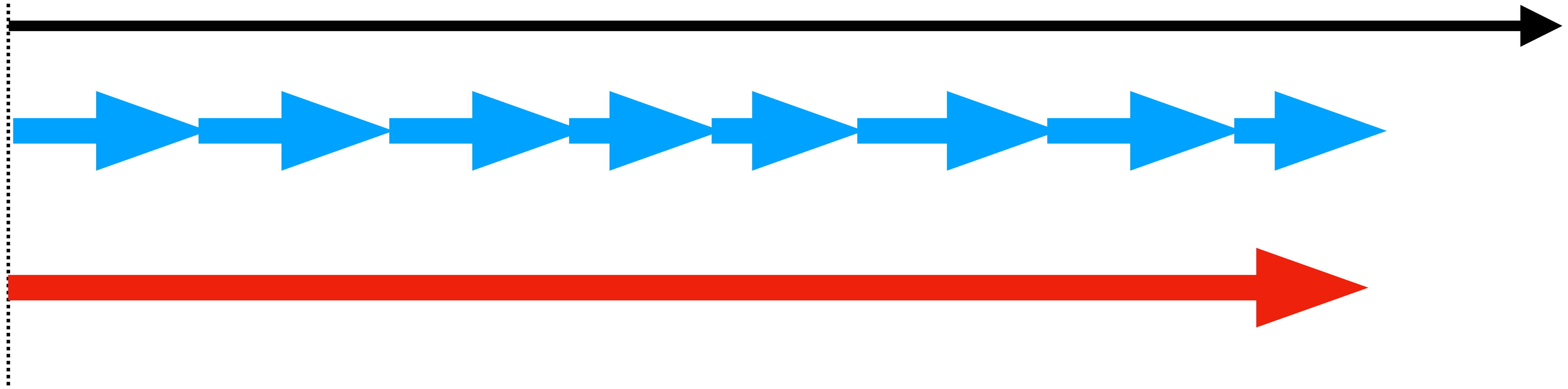
$$\mathbb{E}[f(x_K) - f_*] = \mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{\tau}{K}\right)$$

Theorem. Let f be L-smooth and convex, then *any* Asynchronous Gradient with constant delay τ converges not better than

$$\mathbb{E}[f(x_K) - f_*] = \Omega\left(\frac{\tau^2}{K^2}\right)$$

Motivating example

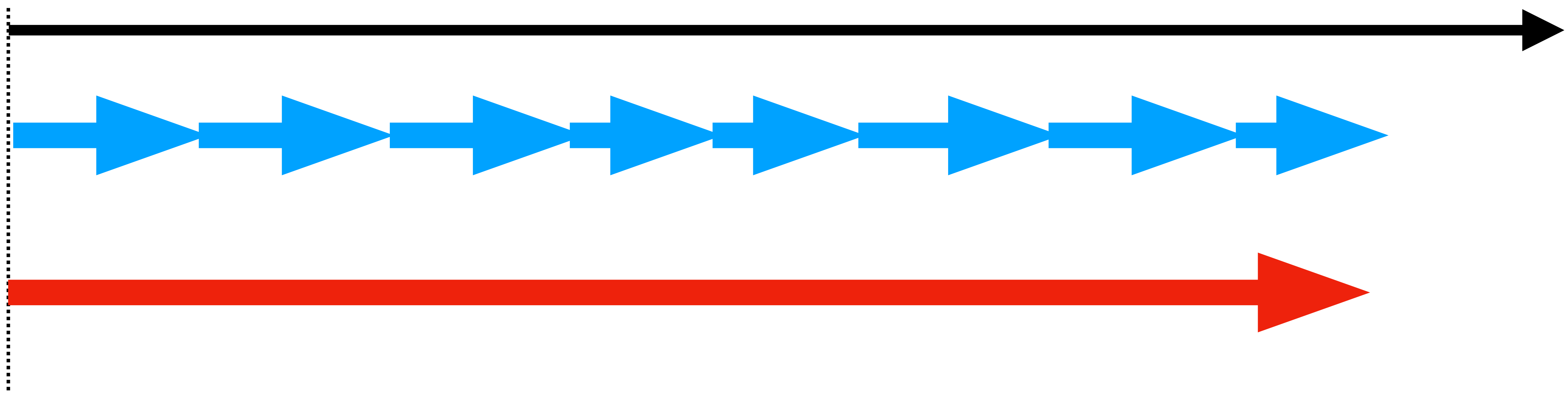
Asynchronous SGD



One extremely slow worker: $\tau = K$

Motivating example

Asynchronous SGD

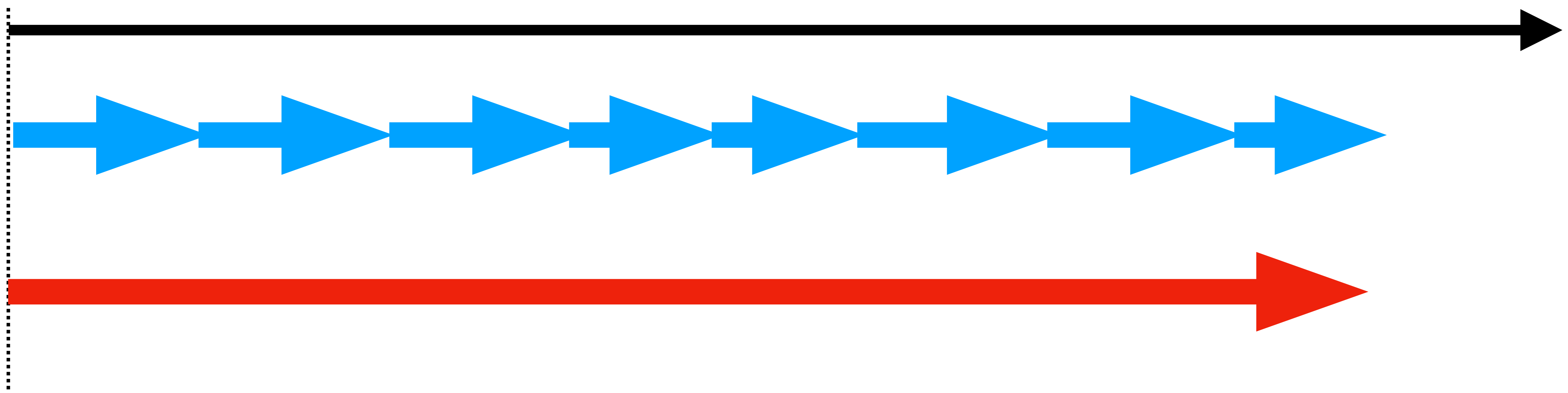


One extremely slow worker: $\tau = K$

$$\mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{\tau}{K}\right) = \mathcal{O}(1)$$

Motivating example

Asynchronous SGD



In general, the delay is $\tau \approx \sum_{m=1}^{M} \frac{\max_j s_j}{s_m}$

s_m: Seconds per gradient by worker m

Asynchronous SGD

Minibatch SGD

$$\frac{SM}{s_{\max}} \text{ gradients}$$

Asynchronous SGD

$$\sum_{m=1}^{M} \frac{S}{s_m} \text{ gradients}$$

Improvement: $\frac{1}{M}\sum_{m=1}^{M} \frac{s_{\max}}{s_m}$

Rate is $\mathcal{O}(M)$

Rate is $\mathcal{O}(\tau)$

Asynchronous SGD

Minibatch SGD

$$\frac{SM}{s_{\max}} \text{ gradients}$$

Asynchronous SGD

$$\sum_{m=1}^{M} \frac{S}{s_m} \text{ gradients}$$

Improvement: $\frac{1}{M} \sum_{m=1}^{M} \frac{s_{\max}}{s_m}$

Rate is $\mathcal{O}(M)$

Rate is $\mathcal{O}(\tau)$

$$\tau \approx \sum_{m=1}^{M} \frac{s_{\max}}{s_m}$$

Asynchronous SGD

Minibatch SGD

$$\frac{SM}{s_{\max}} \text{ gradients}$$

Asynchronous SGD

$$\sum_{m=1}^{M} \frac{S}{s_m} \text{ gradients}$$

$$\text{Improvement: } \frac{1}{M} \sum_{m=1}^{M} \frac{s_{\max}}{s_m}$$

Rate is $\mathcal{O}(M)$

Rate is $\mathcal{O}(\tau)$

$$\tau \approx \sum_{m=1}^{M} \frac{s_{\max}}{s_m}$$

So who is faster?

New results

Theorem.

Smooth nonconvex problems:

$$\min_{k \leq K} \mathbb{E}[\|\nabla f(x_k)\|^2] = \mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{M}{K}\right)$$

New results

Theorem.

Smooth nonconvex problems:

$$\min_{k\leq K} \mathbb{E}[\|\nabla f(x_k)\|^2] = \mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{M}{K}\right)$$

Smooth convex problems:

$$\min_{k\leq K} \mathbb{E}[f(x_k) - f_*] = \mathcal{O}\left(\frac{\sigma}{K} + \frac{M}{K}\right)$$

New results

Theorem.

Smooth nonconvex problems:

$$\min_{k \leq K} \mathbb{E}[\|\nabla f(x_k)\|^2] = \mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{M}{K}\right)$$

Smooth convex problems:

$$\min_{k \leq K} \mathbb{E}[f(x_k) - f_*] = \mathcal{O}\left(\frac{\sigma}{\sqrt{K}} + \frac{M}{K}\right)$$

Smooth strongly convex problems:

$$\min_{k \leq K} \mathbb{E}[f(x_k) - f_*] = \mathcal{O}\left(\frac{\sigma^2}{\mu K} + \exp\left(-\frac{K}{\kappa M}\right)\right)$$

Asynchronous SGD

Minibatch SGD

$$\frac{SM}{s_{\max}} \text{ gradients}$$

Asynchronous SGD

$$\sum_{m=1}^{M} \frac{S}{s_m} \text{ gradients}$$

$$\text{Improvement: } \frac{1}{M}\sum_{m=1}^{M}\frac{s_{\max}}{s_m}$$

Rate is $\mathcal{O}(M)$

Rate is $\mathcal{O}(M)$

Asynchronous SGD is faster

Contradiction with lower bounds

A Tight Convergence Analysis for Stochastic Gradient Descent with Delayed Updates

Yossi Arjevani, Ohad Shamir
Weizmann Institute of Science
Rehovot 7610001, Israel
{yossi.arjevani,ohad.shamir}@weizmann.ac.il

Nathan Srebro
TTI Chicago
Chicago, IL 60637
nati@ttic.edu

Theorem. Let f be L-smooth and convex, then *any* Asynchronous Gradient with constant delay τ converges not better than

$$\mathbb{E}[f(x_K) - f_*] = \Omega\left(\frac{\tau^2}{K^2}\right)$$

Contradiction with lower bounds

A Tight Convergence Analysis for Stochastic Gradient Descent
with Delayed Updates

Yossi Arjevani, Ohad Shamir
Weizmann Institute of Science
Rehovot 7610001, Israel
{yossi.arjevani,ohad.shamir}@weizmann.ac.il

Nathan Srebro
TTI Chicago
Chicago, IL 60637
nati@ttic.edu

Theorem. Let f be L-smooth and convex, then *any* Asynchronous Gradient with constant delay τ converges not better than

$$\mathbb{E}[f(x_K) - f_*] = \Omega\left(\frac{\tau^2}{K^2}\right)$$

Our rate: $\mathcal{O}\left(\frac{M}{K}\right)$

If $\tau = K$, then $\frac{M}{K} \ll 1 = \frac{\tau^2}{K^2}$

Contradiction with lower bounds

A Tight Convergence Analysis for Stochastic Gradient Descent with Delayed Updates

Yossi Arjevani, Ohad Shamir
Weizmann Institute of Science
Rehovot 7610001, Israel
{yossi.arjevani,ohad.shamir}@weizmann.ac.il

Nathan Srebro
TTI Chicago
Chicago, IL 60637
nati@ttic.edu

Theorem. Let f be L-smooth and convex, then *any* Asynchronous Gradient with constant delay τ converges not better than

$$\mathbb{E}[f(x_K) - f_*] = \Omega\left(\frac{\tau^2}{K^2}\right)$$

Our rate: $\mathcal{O}\left(\frac{M}{K}\right)$

If $\tau = K$, then $\frac{M}{K} \ll 1 = \frac{\tau^2}{K^2}$

Contradiction! (and their derivation is correct!)

Contradiction with lower bounds

A Tight Convergence Analysis for Stochastic Gradient Descent with Delayed Updates

Yossi Arjevani, Ohad Shamir
Weizmann Institute of Science
Rehovot 7610001, Israel
{yossi.arjevani,ohad.shamir}@weizmann.ac.il

Nathan Srebro
TTI Chicago
Chicago, IL 60637
nati@ttic.edu

Theorem. Let f be L-smooth and convex, then *any* Asynchronous Gradient with constant delay τ converges not better than

$$\mathbb{E}[f(x_K) - f_*] = \Omega\left(\frac{\tau^2}{K^2}\right)$$

Our rate: $\mathcal{O}\left(\frac{M}{K}\right)$

If $\tau = K$, then $\frac{M}{K} \ll 1 = \frac{\tau^2}{K^2}$

But in their counterexample, $\tau = M$

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

Define without delay

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

Future stepsize associated with future gradient

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

Amazingly powerful idea

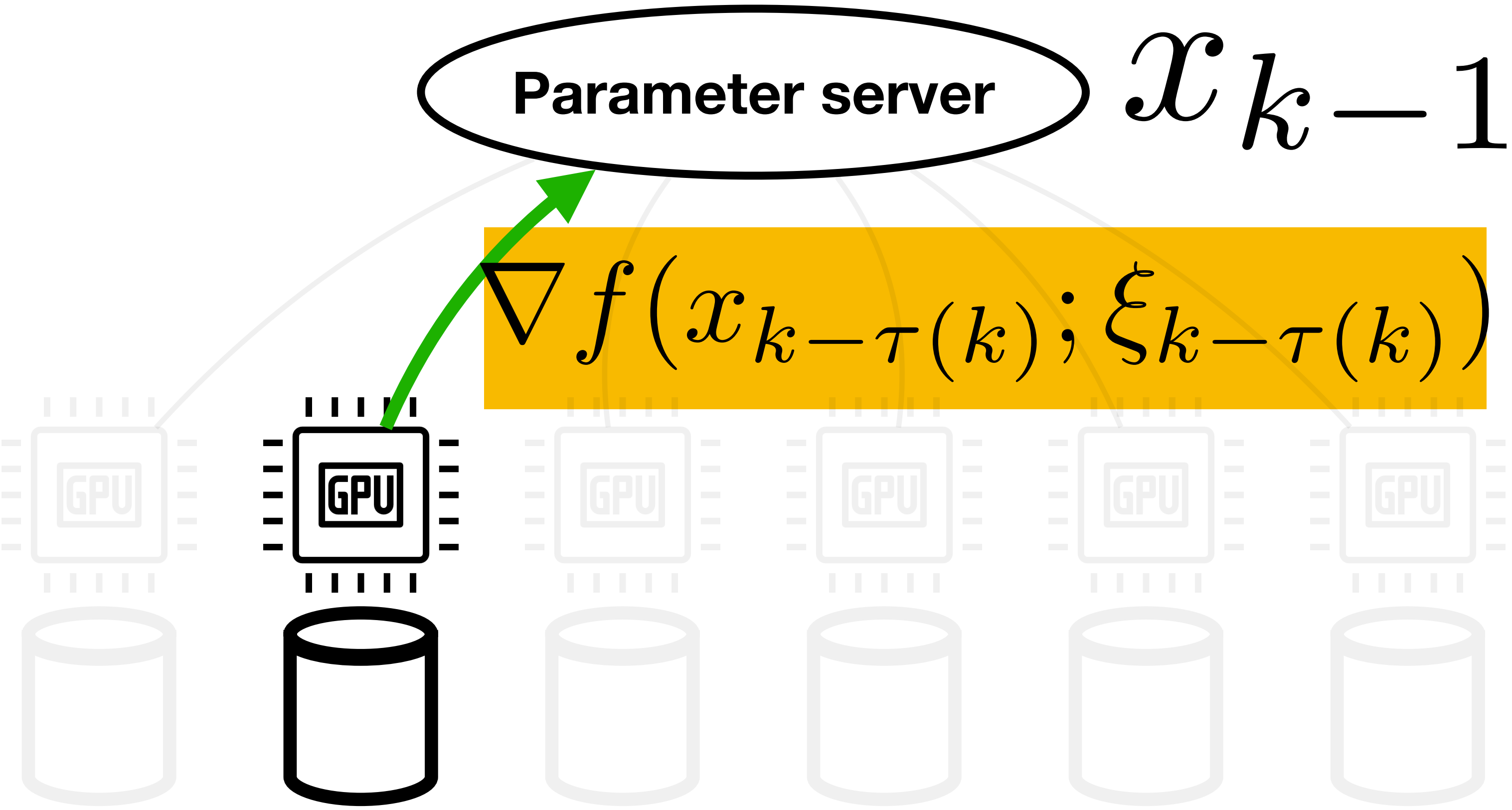
(Mania et al., 2017)

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

$$x_k = x_{k-1} - \gamma_k \nabla f(x_{k-\tau(k)}; \xi_{k-\tau(k)})$$

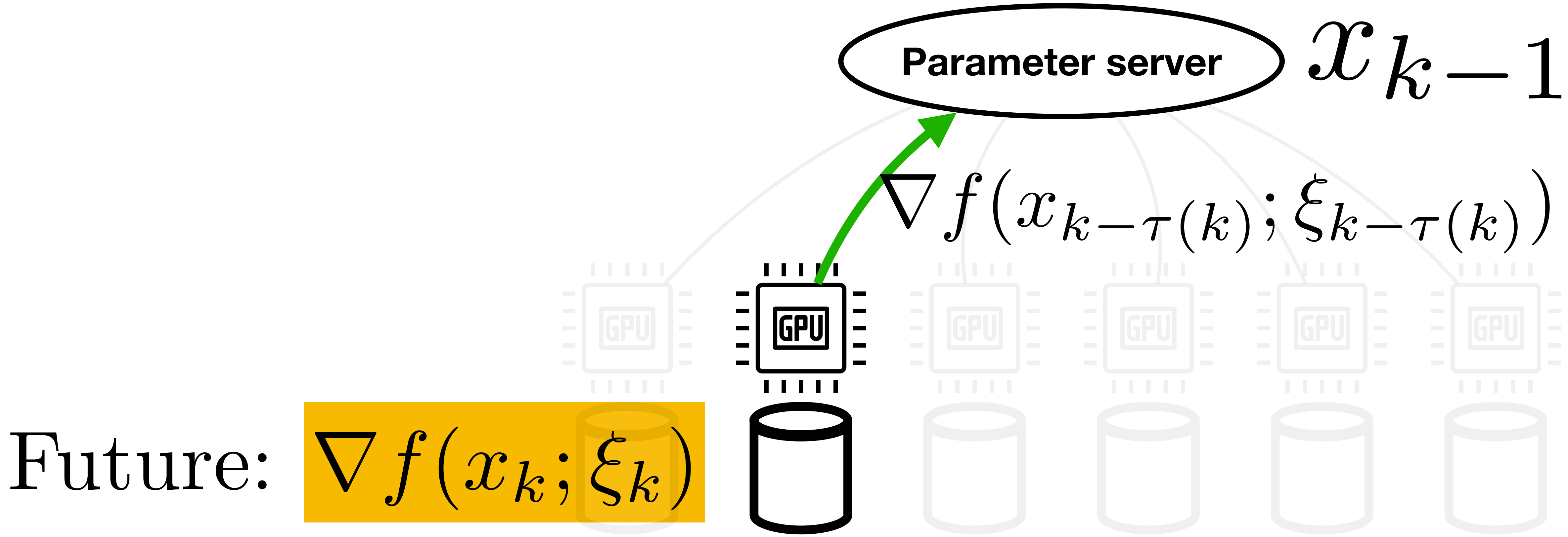


Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

$$x_k = x_{k-1} - \gamma_k \nabla f(x_{k-\tau(k)}; \xi_{k-\tau(k)})$$



Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

$$x_k = x_{k-1} - \gamma_k \nabla f(x_{k-\tau(k)}; \xi_{k-\tau(k)})$$

$x_k - \hat{x}_k \sim$ all promised gradients

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

$$x_k = x_{k-1} - \gamma_k \nabla f(x_{k-\tau(k)}; \xi_{k-\tau(k)})$$

$$x_k - \hat{x}_k = \sum_{m=1}^{M} \gamma_{\mathrm{next}(k,m)} \nabla f(x_{\mathrm{prev}(k,m)}; \xi_{\mathrm{prev}(k,m)})$$

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

$$x_k = x_{k-1} - \gamma_k \nabla f(x_{k-\tau(k)}; \xi_{k-\tau(k)})$$

$$x_k - \hat{x}_k = \sum_{m=1}^{M} \gamma_{\text{next}(k,m)} \nabla f(x_{\text{prev}(k,m)}; \xi_{\text{prev}(k,m)})$$

Gradients that are being computed

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

Difficulty: $\mathbb{E}[\nabla f(x_k; \xi_k)] = \nabla f(x_k) \neq \nabla f(\hat{x}_k)$

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

Difficulty: $\mathbb{E}[\nabla f(x_k; \xi_k)] = \nabla f(x_k) \neq \nabla f(\hat{x}_k)$

Resolution: $\mathbb{E}[\|\hat{x}_k - x_k\|^2] \leq M^2(G^2 + \sigma^2)$

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

Difficulty: $\mathbb{E}[\nabla f(x_k; \xi_k)] = \nabla f(x_k) \neq \nabla f(\hat{x}_k)$

Resolution: $\mathbb{E}[\|\hat{x}_k - x_k\|^2] \leq M^2(G^2 + \sigma^2)$

Upper bound on gradient $\|\nabla f(x)\| \leq G$

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

Difficulty: $\mathbb{E}[\nabla f(x_k; \xi_k)] = \nabla f(x_k) \neq \nabla f(\hat{x}_k)$

If gradient not bounded:

$$\mathbb{E}[\|\hat{x}_k - x_k\|^2] \leq 2 \sum_{m=1}^{M} \gamma^2_{\text{next}(k,m)} (\sigma^2 + \|\nabla f(x_{\text{prev}(k,m)}; \xi_{\text{prev}(k,m)})\|^2)$$

Main ideas

Virtual iterates:

$$\hat{x}_{k+1} = \hat{x}_k - \hat{\gamma}_k \nabla f(x_k; \xi_k)$$

Difficulty: $\mathbb{E}[\nabla f(x_k; \xi_k)] = \nabla f(x_k) \neq \nabla f(\hat{x}_k)$

If gradient not bounded:

$$\mathbb{E}[\|\hat{x}_k - x_k\|^2] \leq 2 \sum_{m=1}^{M} \gamma^2_{\mathrm{next}(k,m)} (\sigma^2 + \|\nabla f(x_{\mathrm{prev}(k,m)}; \xi_{\mathrm{prev}(k,m)})\|^2)$$

$$\gamma_k = \mathcal{O}\left(\frac{1}{L\tau(k)}\right)$$

Limitations

1. Constant stepsize only if bounded gradients

Limitations

1. Constant stepsize only if bounded gradients
2. Delay-dependent stepsize only if noise and delay are independent

Limitations

1. Constant stepsize only if bounded gradients
2. Delay-dependent stepsize only if noise and delay are independent
3. Workers must have same data

Limitations

1. Constant stepsize only if bounded gradients
2. Delay-dependent stepsize only if noise and delay are independent
3. Workers must have same data
4. Still centralized