MindFlayer SGD: Efficient Parallel SGD in the Presence of Heterogeneous and Random Worker Compute Times

Artavazd Maranjyan, Omar Shaikh Omar, Peter Richtárik

uai

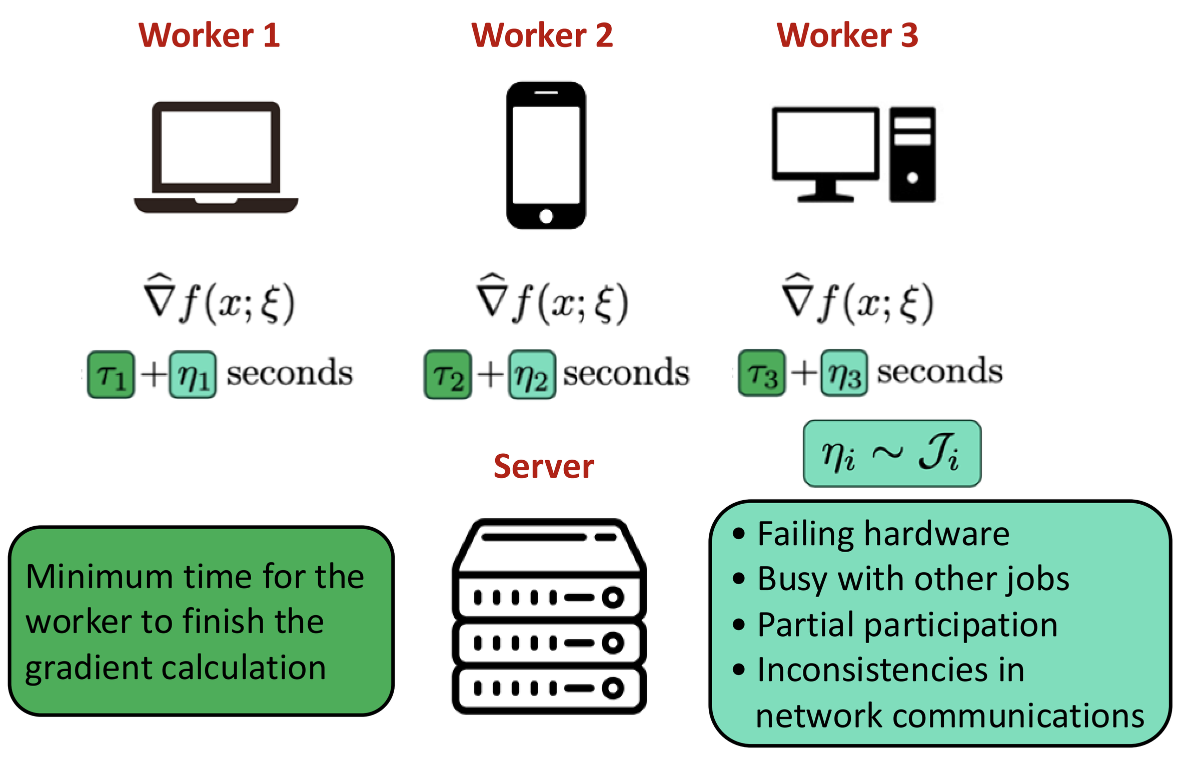
جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Problem Setup: Distributed Training

We address the nonconvex optimization problem:

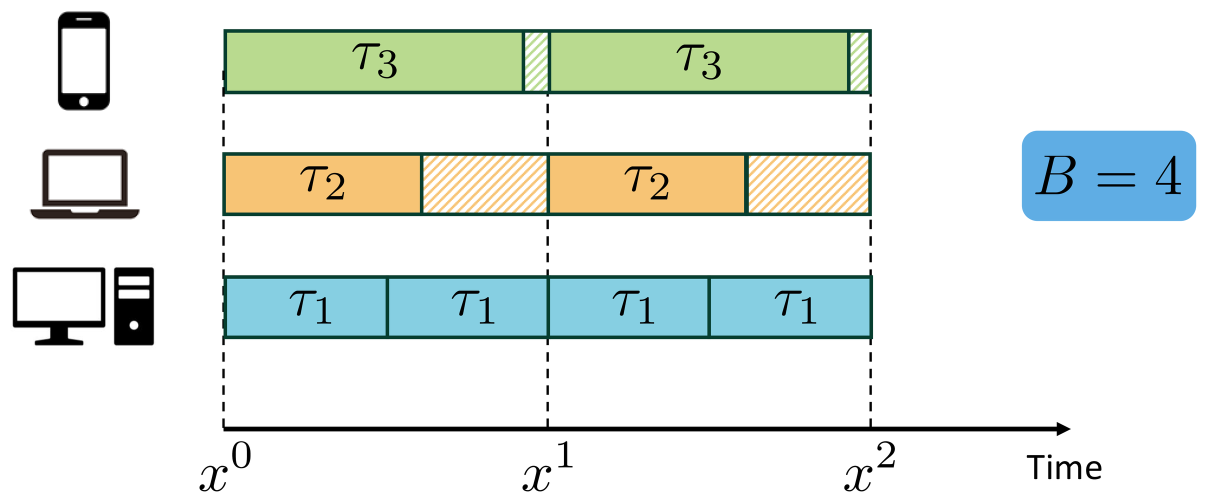
$$\min_{x\in\mathbb{R}^d} \left\{ f(x) := \mathbb{E}_{\xi\sim\mathcal{D}}\left[f(x;\xi)\right] \right\},$$

We assume we have access to n parallel workers that compute stochastic gradients independently



Optimal Method in Deterministic Regime

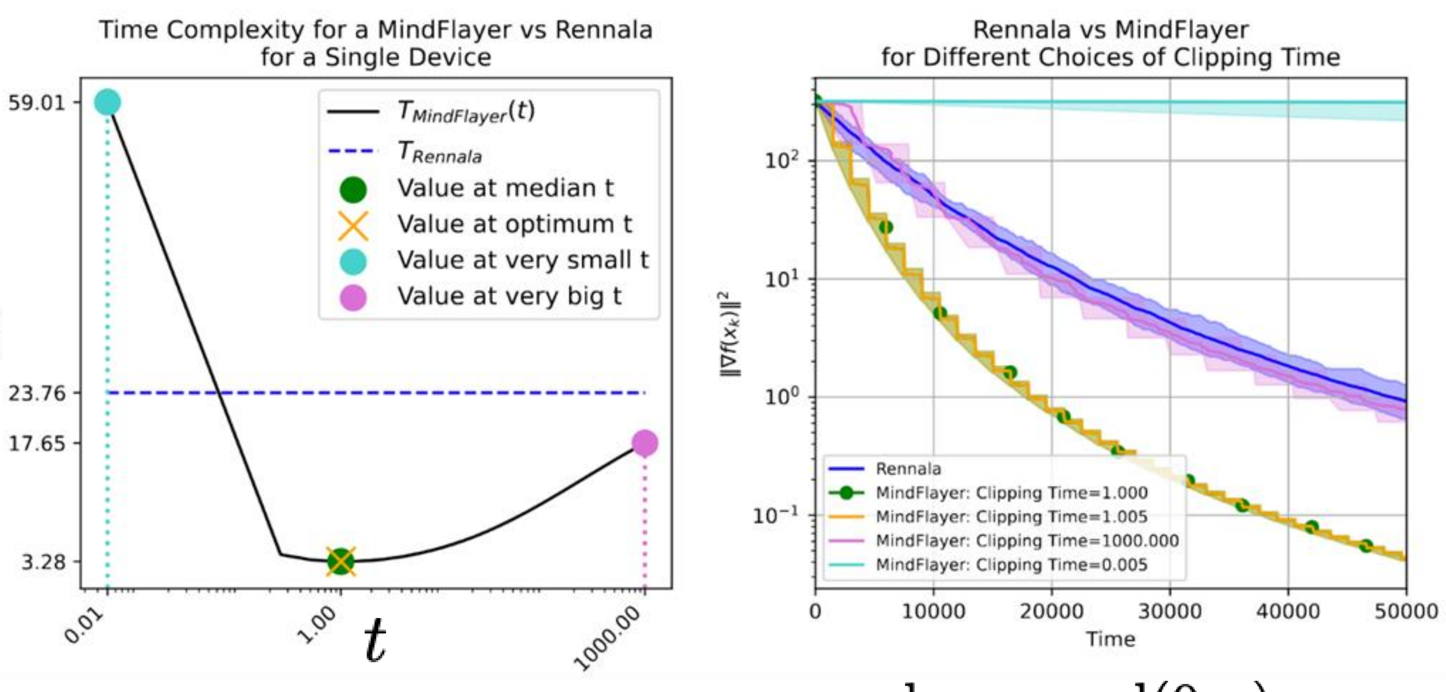
Rennala SGD (Tyurin & Richtárik, 2024)



$$x^{k+1} = x^k - \gamma \frac{1}{B} \sum_{j=1}^{B} \nabla f(x^k;\xi_j^k)$$

Motivation and the Single Device Case

MindFlayer SGD:
- Wait t seconds for gradient response
- If the timeout expires, resend the request



$$\eta \sim \mathrm{lognormal}(0, s)$$

MindFlayer SGD

Worker 1

Try computing $\widehat{\nabla} f(x^k;\xi)$ within t_1 seconds

Try computing $\widehat{\nabla} f(x^k;\xi)$ within t_1 seconds

Try computing $\widehat{\nabla} f(x^k;\xi)$ within t_1 seconds

B_1 trials

Worker 2

Try computing $\widehat{\nabla} f(x^k;\xi)$ within t_2 seconds

Try computing $\widehat{\nabla} f(x^k;\xi)$ within t_2 seconds

Try computing $\widehat{\nabla} f(x^k;\xi)$ within t_2 seconds

Try computing $\widehat{\nabla} f(x^k;\xi)$ within t_2 seconds

B_2 trials

Worker 3

Try computing $\widehat{\nabla} f(x^k;\xi)$ within t_3 seconds

Try computing $\widehat{\nabla} f(x^k;\xi)$ within t_3 seconds

B_3 trials

MindFlayer (Server)

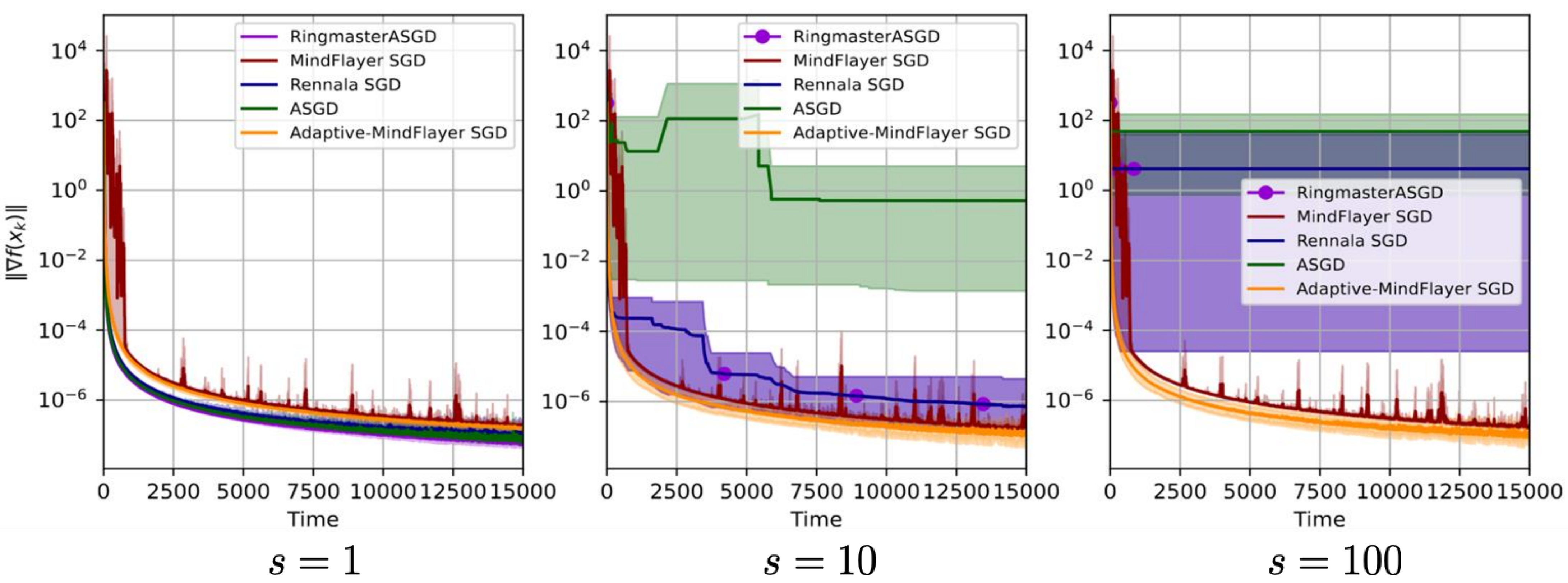
$$x^{k+1} = x^k - \gamma \frac{1}{B} \sum_{i=1}^{n} \sum_{j=1}^{B_i} I\left(\eta_i^j \le t_i\right) \nabla f\left(x^k;\xi_i^j\right)$$

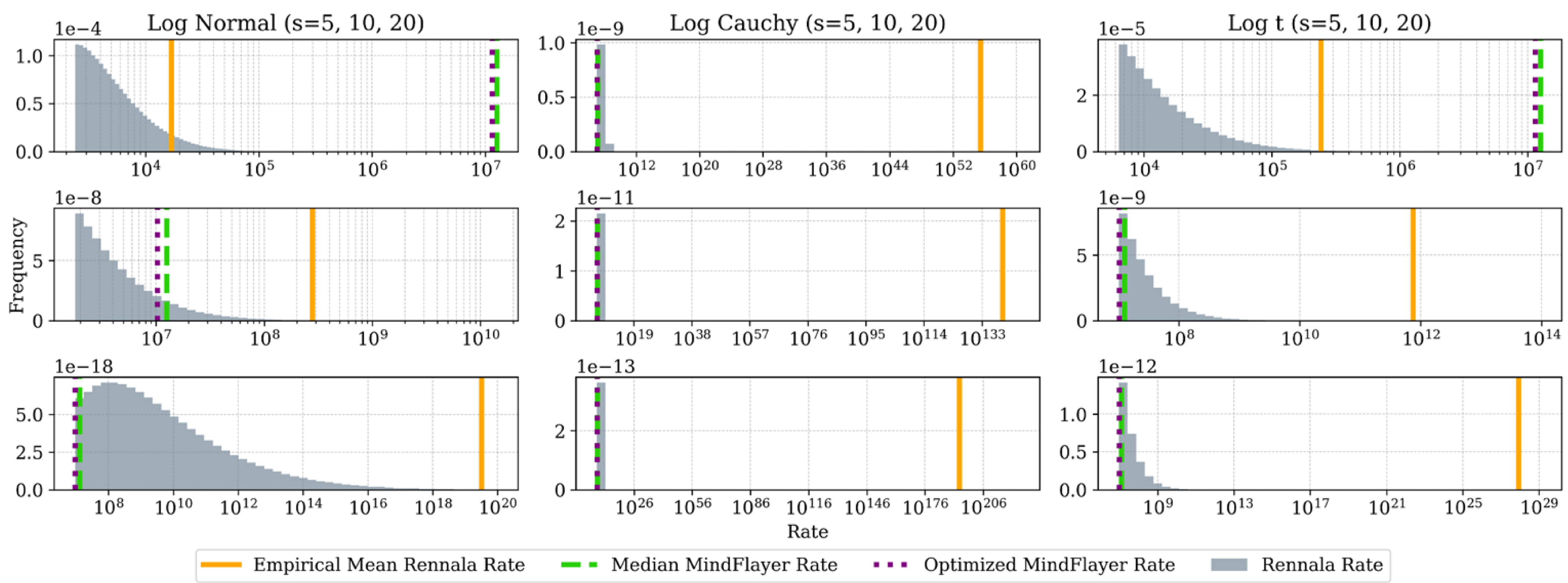
$$B = \sum_{i=1}^{n} p_i B_i \;\text{ and }\; p_i = F_i(t_i) = P(\eta_i \le t_i)$$

Experiments

$$f(x) = \frac{1}{2} x^\top A x - b^\top x \qquad \forall x \in \mathbb{R}^d$$

$$\eta \sim \mathrm{lognormal}(0, s)$$





Theoretical Results

Assumptions

Assumption 1. Function f is differentiable, and its gradient is L–Lipschitz continuous, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x-y\|, \text{ for all } x, y \in \mathbb{R}^d.$$

Assumption 2. The function $f(x)$ is bounded below, and we denote its infimum by $f^{\inf} \in \mathbb{R}$. Let x^0 be the initial point of the optimization method, define $\Delta := f(x^0) - f^{\inf}$.

Assumption 3. For all $x \in \mathbb{R}^d$, stochastic gradients $\nabla f(x;\xi)$ are unbiased and σ^2-variance-bounded, i.e.,

$$\mathbb{E}_\xi\left[\nabla f(x;\xi)\right] = \nabla f(x),$$
$$\mathbb{E}_\xi\left[\|\nabla f(x;\xi) - \nabla f(x)\|^2\right] \le \sigma^2,$$

where $\sigma^2 \ge 0$.

Iteration Complexity

Theorem 1. *Let*

$$B = \sum_{i=1}^{n} p_i B_i \quad \textit{and} \quad \gamma = \frac{1}{2L}\min\left\{1, \frac{\varepsilon B}{\sigma^2}\right\}$$

Then, the method guarantees that $\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla f(x^k)\|^2\right] \le \varepsilon$ *if*

$$K \ge \max\left\{1, \frac{\sigma^2}{\varepsilon B}\right\} \frac{8L\left(f(x^0) - f^{\inf}\right)}{\varepsilon}$$

Time Complexity

$$p_j = F_j(t_j) = P(\eta_j \le t_j) \qquad \Delta = f(x^0) - f^{\inf}$$

$$\min_{m\in[n]}\left\{\left(\frac{1}{m}\sum_{j=1}^{m}\frac{p_j}{\tau_j + t_j}\right)^{-1}\left(\frac{S}{m} + \frac{1}{m}\sum_{j=1}^{m} p_j\right)\frac{\Delta L}{\varepsilon}\right\}$$

$$S = \max\left\{\frac{\sigma^2}{\varepsilon}, 1\right\}$$