Codes for Straggler Mitigation in Secure Distributed Linear Regression

Rutgers, The State University of New Jersey

Rawad Bitar

Advisor: Prof. Salim El Rouayheb

Distributed Linear Regression

Linear regression is an algorithm aiming to represent a data matrix A, labeled with a vector $\mathbf{y}$, by a vector of attributes $\mathbf{x}^*$ such that $A\mathbf{x}^* \approx \mathbf{y}$.

A possible solution is to run the following iterative computation

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha^{(t)} A^T (A\mathbf{x}^{(t)} - \mathbf{y}). \tag{1}$$

Here $\alpha^{(t)}$ is called the learning rate and $\mathbf{x}^*$ is found when $\mathbf{x}^{(t+1)} \approx \mathbf{x}^{(t)}$.

In a distributed setting, one wants to divide the main computation into smaller tasks and distribute them to n worker machines that can perform these smaller tasks in parallel.

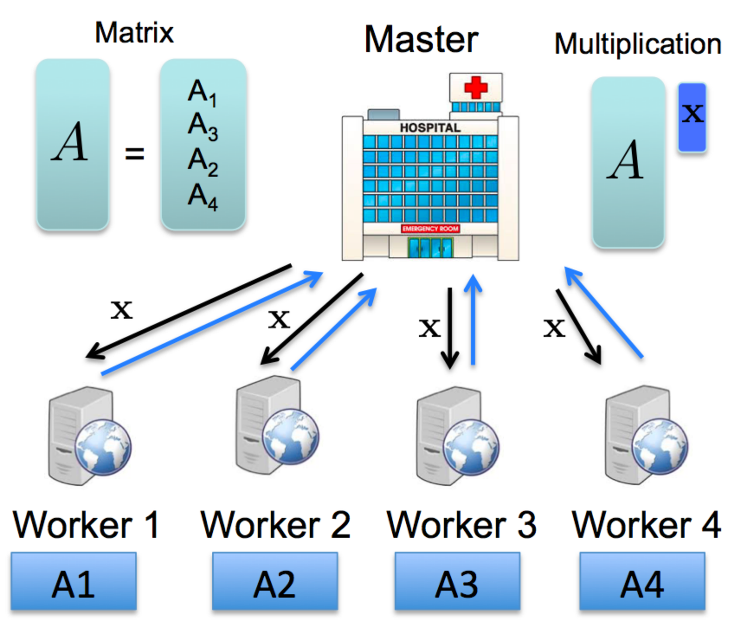
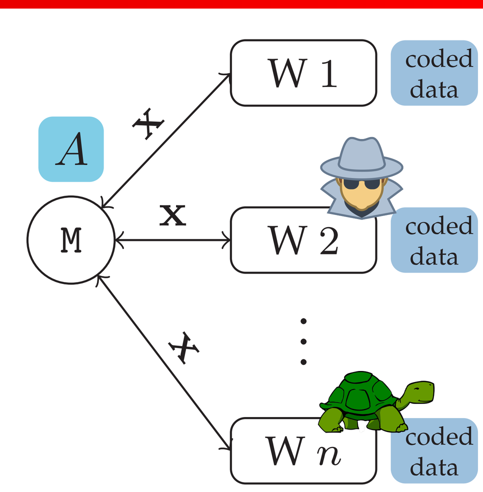


Fig. 1: An example of distributed linear regression. The computation given in (1) is divided into two multiplications: $A\mathbf{x}$ and $A^T\mathbf{b}$, with $b \triangleq A\mathbf{x} - \mathbf{y}$. Each worker multiplies his matrix by the vector sent to them and sends the result back to the Master.

Secrecy and Stragglers

Master `M` owns A and wants $A\mathbf{x}$.

- n workers are available.
- $n-k$ workers might be stragglers ($k < n$).
- z workers can collude to learn A ($z < k$).



Main challenges:

Secrecy: maintain the confidentiality of A, in an information theoretic sense.

Stragglers: Should `M` wait for the slow workers? [1]

Assumptions:

Communication cost as proxy for latency.

Delays are shifted exponential random variables [2],

$$F(delay = t) \triangleq 1 - e^{-\lambda(t-c)}, \quad \text{for } t \geq c.$$

Goal:

Design codes that ensure secrecy, mitigate any number of stragglers up to a given threshold and minimize the communication cost.

Staircase Codes

Theorem 1 ([3]) *The (n,k,z) Staircase code constructed as follows mitigates any number of stragglers $n-d$, $k \leq d \leq n$, and achieves minimum download cost given by*

$$\textit{download cost} = \frac{d(k-z)}{d-z}$$

for all values of d.

$$\begin{bmatrix} \text{Worker } 1 \\ \vdots \\ \text{Worker } n \end{bmatrix} = \begin{bmatrix} \text{Vandermonde} \end{bmatrix} \begin{bmatrix} \mathcal{S} & \mathcal{D}_1 & \mathcal{D}_2 & \cdots & \mathcal{D}_{h-1} \\ & & \mathcal{R}_3 & \cdots & \mathcal{R}_h \\ & \mathcal{R}_2 & & & \\ \mathcal{R}_1 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

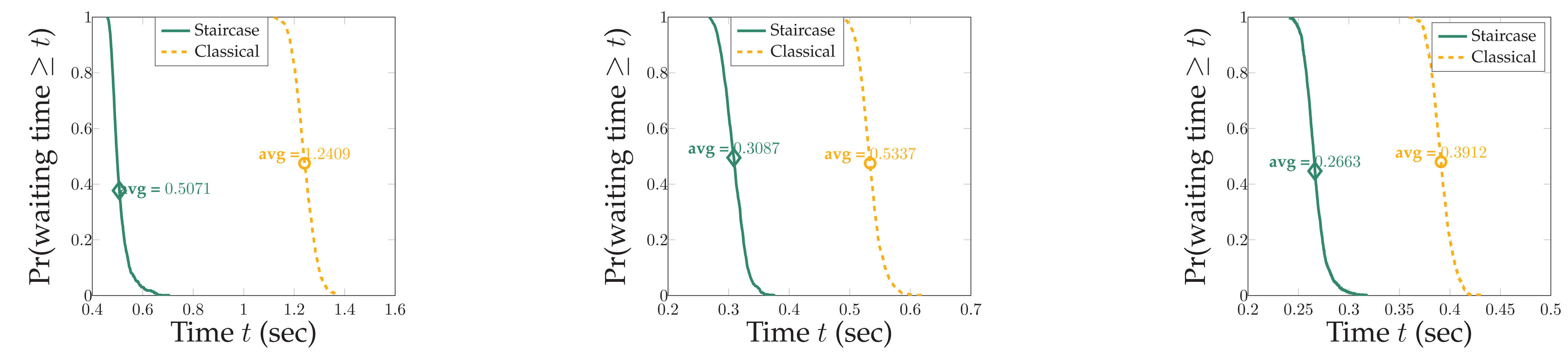
Example: An $(n,k,z) = (4,2,1)$ Staircase code constructed over $GF(5)$.

Worker 1	Worker 2	Worker 3	Worker 4
$A_1 + A_2 + A_3 + R_1$	$A_1 + 2A_2 + 4A_3 + 3R_1$	$A_1 + 3A_2 + 4A_3 + 2R_1$	$A_1 + 4A_2 + A_3 + 4R_1$
$A_4 + A_5 + A_6 + R_2$	$A_4 + 2A_5 + 4A_6 + 3R_2$	$A_4 + 3A_5 + 4A_6 + 2R_2$	$A_4 + 4A_5 + A_6 + 4R_2$
$R_1 + R_2 + R_3$	$R_1 + 2R_2 + 4R_3$	$R_1 + 3R_2 + 4R_3$	$R_1 + 4R_2 + R_3$
$A_3 + R_4$	$A_3 + 2R_4$	$A_3 + 3R_4$	$A_3 + 4R_4$
$A_6 + R_5$	$A_6 + 2R_5$	$A_6 + 3R_5$	$A_6 + 4R_5$
$R_3 + R_6$	$R_3 + 2R_6$	$R_3 + 3R_6$	$R_3 + 4R_6$

Ongoing Work and Future Directions

- ⋄ *Heterogeneous system:* Workers have different properties.
- ⋄ *Malicious workers:* Workers might send wrong results to `M`.
- ⋄ *Beyond linear regression:* Flexible straggler mitigation for gradient descent and stochastic gradient descent algorithms.
- ⋄ *Dynamic environment:* Workers can join and leave the system at any time.
- ⋄ *Delay analysis:* Study the delay model for small amount of data.

Implementation on Amazon Cloud



$(a) : (n,k,z) = (4,2,1)$. $(b) : (n,k,z) = (10,5,1)$. $(c) : (n,k,z) = (20,10,1)$.

Figure 2: Empirical complementary CDF of the Master's waiting time (and its average) observed on Amazon EC2 clusters for systems with rate $k/n = 1/2$. The data matrix A is a 378000×250 matrix with entries generated uniformly at random from $\{1,\dots,255\}$. Staircase codes bring 59% reduction in the mean waiting time for $n = 4$. Those numbers were obtained by repeating the multiplication process 1000 times.

Savings in Waiting Time at the Master

Theorem 2 (Lower bound on savings in mean waiting time [4]) *The savings on mean waiting time $\mathbb{E}[T_{SC}]$ of an (n,k,z) system using Staircase codes compared to k-out-of-n classical codes is lower bounded by*

$$\textit{Savings} \geq 1 - \min_{d \in \{k,\dots,n\}} \left\{ \frac{(k-z)(\lambda c + H_n - H_{n-d})}{(d-z)(\lambda c + H_n - H_{n-k})} \right\}. \tag{2}$$

where H_n is the n^{th} harmonic sum defined as $H_n \triangleq \sum_{i=1}^{n} \frac{1}{i}$, and $H_0 \triangleq 0$.

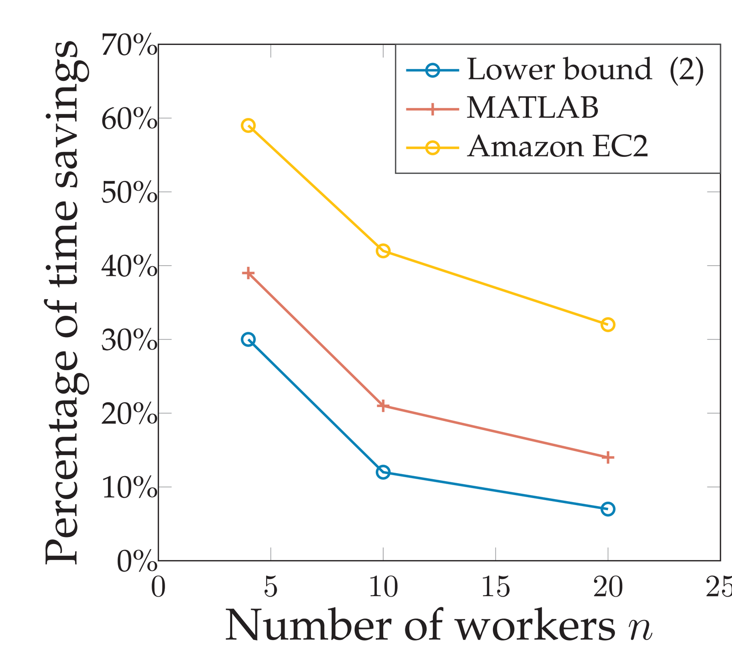


Figure 3: The savings brought by $(n, n/2, 1)$ Staircase codes on Amazon EC2 instances compared to theoretical results for large amounts of data. The data is a matrix of size 387000×250. The savings obtained on Amazon match the theoretical expectations.

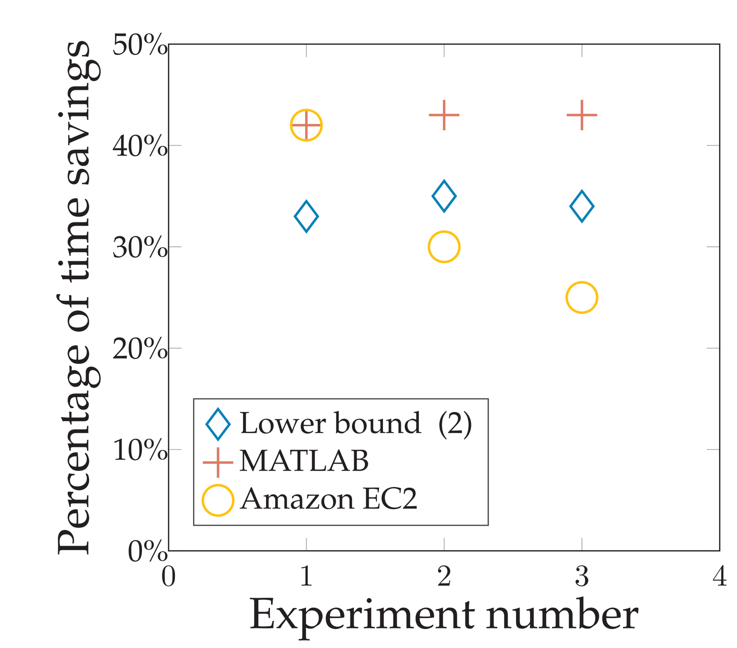


Figure 4: The savings brought by $(4,2,1)$ Staircase codes on Amazon EC2 instances compared to theoretical results for small amounts of data. The data is a matrix of size 42000×250. The savings obtained on Amazon EC2 do not match theoretical expectations.

References

[1] J. Dean and L. A. Barroso, "The tail at scale," *Communications of the ACM*, vol. 56, no. 2, pp. 74–80, 2013.

[2] G. Liang and U. C. Kozat, "TOFEC: Achieving optimal throughput-delay trade-off of cloud storage using erasure codes," in *IEEE International Conference on Computer Communications*, 2014.

[3] **R. Bitar** and S. El Rouayheb, "Staircase codes for secret sharing with optimal communication and read overheads," *IEEE Transactions on Information Theory*, Vol. 64, No. 2, February 2018.

[4] **R. Bitar**, P. Parag, and S. El Rouayheb, "Minimizing Latency in Secure Distributed Computing via Staircase Codes," submitted to *IEEE Transactions on Information Theory*.