Secure Distributed Computing on Untrusted Workers

Rawad Bitar, Parimal Parag and Salim El Rouayheb

Illinois Institute of Technology

Indian Institute of Science

Distributed Computing

In the setting of distributed computing, a Master server M possesses big amounts of data and wants to perform intensive computations on it. M wants to divide these computations into smaller computational tasks and distribute them to n worker machines that can perform these smaller tasks in parallel.

The process could be iterative requiring the workers to run several computations on the same data, e.g., machine learning algorithms. At each iteration, the workers return their results to the master, who can process them to obtain the result of its original task.

Distributed computing is found in several applications ranging from managing user requests in Data Centers, e.g. MapReduce, to peer-to-peer machine learning algorithms, e.g., folding@home.

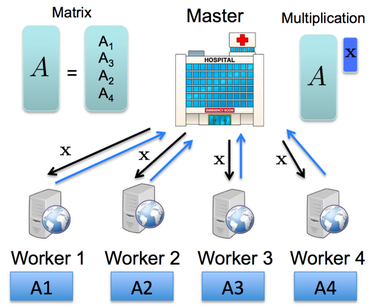
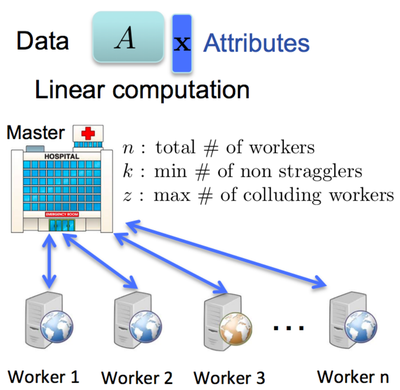


Fig. 1: An example of distributed computing. The Master divides the data matrix A among the workers and send them a vector $\mathbf{x}$. Each worker multiplies his matrix by x and sends the result to the Master who decodes $A\mathbf{x}$.

Problem Formulation



Master possesses *confidential* data on which it wants to perform distributed iterative *linear computations*.

The Master M sends the data to n workers. At each iteration, M sends the attribute vector $\mathbf{x}$ to the workers who multiply it by the data they have and send back the result to the Master.

Main challenges:

Data confidentiality: The data must remain confidential. We assume that any $z < k$ workers can collude.

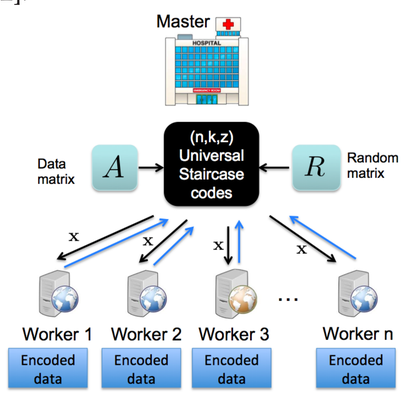
System latency: At each iteration, at most $n - k$ workers might be slow (stragglers) [1].

Goal: Design codes that ensure data confidentiality and account for any number of stragglers between 0 and $n - k$.

Staircase vs Classical Codes

As a proxy for latency, we use communication cost.

Model the workers service time by a random variable drawn according to shifted exponential distribution [2].



An (n, k, z) classical code ensures data confidentiality against any z colluding workers and allows the master to decode $A\mathbf{x}$ from any k responses.

An (n, k, z) Staircase code [3] ensures data confidentiality against any z colluding workers and allows the master to decode $A\mathbf{x}$ by receiving a fraction of the responses of any d workers, $k \leq d \leq n$.

Universal Staircase codes *minimize delays*, because they achieve *minimum communication cost* [4] for *any number* of non-stragglers between k and n.

Results: Distribution of the waiting time at the Master

Theorem 1 (Bounds on mean waiting time [5]) *The mean waiting time $\mathbb{E}[T_{SC}]$ of an (n, k) system using Staircase codes is upper bounded by*

$$\mathbb{E}[T_{SC}] \leq \min_{d\in\{k,\ldots,n\}} \left(\frac{H_n - H_{n-d}}{\lambda(d-z)} + \frac{c}{d-z} \right), \tag{1}$$

where H_n is the n^{th} harmonic sum defined as $H_n \triangleq \sum_{i=1}^{n} \frac{1}{i}$, and $H_0 \triangleq 0$. The mean waiting time is lower bounded by

$$\mathbb{E}[T_{SC}] \geq \frac{c}{n-z} + \max_{d\in\{k,\ldots,n\}} \sum_{i=0}^{k-1} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} \frac{2(-1)^j}{\lambda\left(2(n-i+j)(d-z)+(n-d)(n-d+1)\right)}. \tag{2}$$

Theorem 2 (Exact CDF of mean waiting time [5]) *Let $t_i \triangleq t(i-z)/(k-z)$, the CDF of the waiting time T_{SC} of an (n, k) system using Staircase codes is given by*

$$F_{T_{SC}}(t) = 1 - n! \int_{y\in A(t)} \frac{F(y_k)^{k-1}}{(k-1)!} dF(y_n) \cdots dF(y_k), \tag{3}$$

where $A(t) = \cap_{i\geq k}\{y_i \in (t_i, y_{i+1}]\}$ and $F(y_i) = F_{T_i}(y_i)$.

Example and Implementation on Amazon cloud

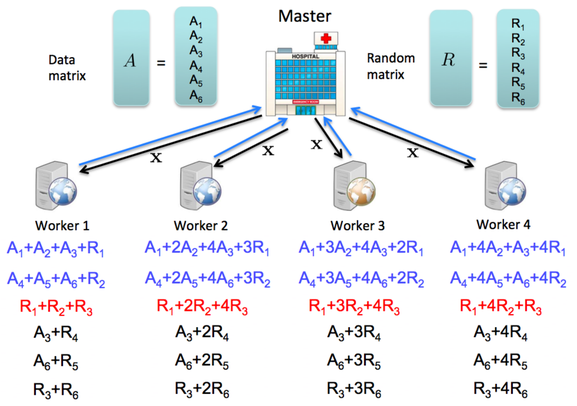


Fig. 2: Encoding of the $(4, 2, 1)$ Universal Staircase code. When there are no stragglers, the Master can decode $A\mathbf{x}$ after receiving the multiplication of the 4 blue parts by $\mathbf{x}$. If one worker is a straggler, the Master decodes $A\mathbf{x}$ after receiving the multiplication of 3 blue and 3 red parts by $\mathbf{x}$. If two workers are stragglers, the master waits until receiving the multiplication of the blue, red and black parts of any two workers by $\mathbf{x}$ to decode $A\mathbf{x}$.

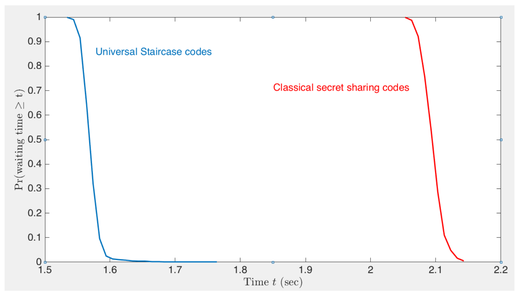


Fig. 3: Implementation of the $(4, 2, 1)$ system on Amazon cloud. The blue curve depicts the distribution of the Master's waiting time when using Universal Staircase codes. The red curve depicts the distribution of the Master's waiting time when using a classical secret sharing code.

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," accepted with minor changes to the *IEEE Transactions on Information Theory*, 2017.

[4] W. Huang, M. Langberg, J. Kliewer, and J. Bruck, "Communication efficient secret sharing," *arXiv preprint arXiv:1602.04496*, 2015.

[5] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure distributed computing," accepted in *IEEE International Symposium on Information Theory (ISIT)*, Aachen, June 2017. *preprint arXiv:1703.01504*.