Minimizing Latency for Secure Distributed Computing

Rawad Bitar, Parimal Parag, and Salim El Rouayheb

***Abstract*—We consider the setting of a master server who possesses confidential data (genomic, medical data, etc.) and wants to run intensive computations on it, as part of a machine learning algorithm for example. The master wants to distribute these computations to untrusted workers who have volunteered or are incentivized to help with this task. However, the data must be kept private (in an information theoretic sense) and not revealed to the individual workers. The workers may be busy and will take a random time to finish the task assigned to them. We are interested in reducing the aggregate delay experienced by the master. We focus on linear computations as an essential operation in many iterative algorithms. A known solution is to use a linear secret sharing scheme to divide the data into secret shares on which the workers can compute. We propose to use instead new secure codes, called Staircase codes, introduced previously by two of the authors. We study the delay induced by Staircase codes which is always less than that of secret sharing. The reason is that secret sharing schemes need to wait for the responses of a fixed fraction of the workers, whereas Staircase codes offer more flexibility in this respect. For instance, for codes with rate $R = 1/2$ Staircase codes can lead to up to 40% reduction in delay compared to secret sharing.**

I. Introduction

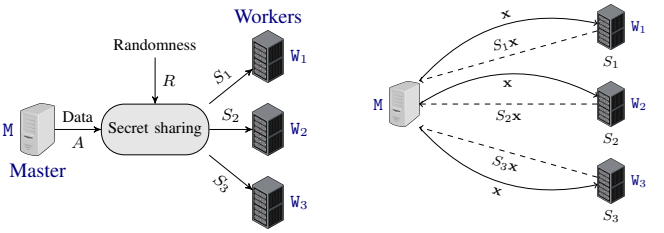
We consider the setting of distributed computing in which a server M, referred to as Master, possesses *confidential* data and wants to perform intensive computations on it. M wants to divide these computations into smaller computational tasks and distribute them to n *untrusted* worker machines that can perform these smaller tasks in parallel. The workers then return their results to the master, who can process them to obtain the result of its original task.

In this paper, we are interested in applications in which the worker machines do not belong to the same system or cluster as the master. Rather, the workers are online computing machines that can be hired or can volunteer to help the master in its computations, e.g., crowdsourcing platforms [1], [2]. The additional constraint that we worry about here, is that the workers cannot be trusted with the sensitive data, which must remain hidden from them. Our privacy constraint is information theoretic, meaning that each worker must obtain zero information about the data irrespective of its computational power. This is in contrast to computational privacy, achieved here by homomorphic encryption algorithms, which rely on the assumed hardness of certain mathematical problems. We choose information theoretic privacy instead of homomorphic encryption, due to the high computation and memory overheads of the latter [3].

We focus on linear computations (matrix multiplication) since they form a basic building block of many iterative algorithms. The workers introduce random delays due to the difference of their workloads or network congestion. This causes the Master to wait for the slowest workers, referred to as stragglers in the distributed computing community [4]–[6]. Our goal is to reduce the delay at the Master.

Privacy can be achieved by encoding the data using linear secret sharing codes [7] as illustrated in Example 1. However, these codes are not specifically designed to minimize latency as we will highlight later.

Example 1. *Let the matrix A denote the data set owned by M and let $\mathbf{x}$ be a given vector. M wants to compute $A\mathbf{x}$. Suppose that M gets the help of $n = 3$ workers out of which at most $n - k = 1$ may be a straggler. M generates a random matrix R of same dimensions as A and over the same field. M encodes A and R into 3 shares $S_1 = R$, $S_2 = R+A$ and $S_3 = R+2A$ using a secret sharing scheme [8], [9]. M sends share S_i to worker W_i (Figure 1a) and then sends $\mathbf{x}$ to all the workers. Each worker computes $S_i\mathbf{x}$ and sends it back to M (Figure 1b). M can decode $A\mathbf{x}$ after receiving any $k = 2$ responses. For instance, if the first two workers respond, M can obtain $A\mathbf{x} = S_2\mathbf{x}-S_1\mathbf{x}$. No information about A is revealed to the workers, because A is one-time padded by R.*



(a) M encodes A into 3 secret shares S_1, S_2, S_3 and sends them to the workers.

(b) M sends $\mathbf{x}$ to the workers. Each worker W_i computes $S_i\mathbf{x}$ and sends the result to M.

Fig. 1: Secure distributed matrix multiplication with 3 workers.

In the previous example, even if there were no stragglers, M still has to wait for the full responses of two workers, and the response of the third one will not be used for decoding. This is due to the fact that classical secret sharing codes are designed for the worst-case scenario. We overcome this limitation by using Staircase codes introduced in [10] and which allow more flexibility in decoding, as explained in the next example.

R. Bitar and S. El Rouayheb are with the ECE department of Illinois Institute of Technology. P. Parag is with the ECE department of the Indian Institute of Science. Emails: rbitar@hawk.iit.edu, parimal@ece.iisc.ernet.in, salim@iit.edu.

This work was supported in parts by ARL Grant W911NF-17-1-0032.

Worker 1	Worker 2	Worker 3
$A_1 + A_2 + R_1$	$A_1 + 2A_2 + 4R_1$	$A_1 + 3A_2 + 4R_1$
$R_1 + R_2$	$R_1 + 2R_2$	$R_1 + 3R_2$

TABLE I: The shares sent by M to each worker. All operations are in $GF(5)$.

Example 2 (Staircase codes)**.** *Consider the same setting as Example 1. Instead of using a classical secret sharing code,* M *now encodes A and R using the Staircase code given in Table I. The Staircase code requires* M *to divide the matrices A and R into $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}^T$ and $R = \begin{bmatrix} R_1 & R_2 \end{bmatrix}^T$. In this setting,* M *sends two subshares to each worker, hence each task consists of 2 subtasks. The master sends $\mathbf{x}$ to all the workers. Each worker multiplies the subshares by $\mathbf{x}$ (going top to bottom) and sends each multiplication back to* M *independently. Now,* M *has two possibilities for decoding:* 1) M *receives the first subtask from all the workers, i.e., receives $(A_1+A_2+R_1)\mathbf{x}$, $(A_1+2A_2+4R_1)\mathbf{x}$ and $(A_1+2A_2+4R_1)\mathbf{x}$ and decodes $A\mathbf{x}$ which is the concatenation of $A_1\mathbf{x}$ and $A_2\mathbf{x}$. Note that* M *decodes only $R_1\mathbf{x}$ and does not need to decode $R_2\mathbf{x}$.* 2) M *receives all the subtasks from any 2 workers and decodes $A\mathbf{x}$. Here* M *has to decode $R_1\mathbf{x}$ and $R_2\mathbf{x}$. One can check that no information about A is revealed to the workers.*

Under an exponential delay model for each worker, we show that the Staircase code given in Example 2 can lead to a 25% improvement in delay over the secret sharing code given in Example 1. Our goal is to give a general systematic study of the delay incurred by Staircase codes and compare it to classical secret sharing codes.

Related work: Recently, there has been a growing research interest in studying codes for delay minimization and straggler mitigation. The early body of work focused on minimizing latency of content download in distributed storage systems. For instance, Huang et al. [11] proposed the use of MDS codes to reduce latency. Joshi et al. studied in [12] the trade-off between storage cost and content download time. Liang and Kozat [13] adaptively encoded the tasks depending on the workload at the workers end. Kadhe et al. [14] proposed the use of availability codes instead of MDS codes to account for straggler mitigation.

For distributed computing systems, Lee et al. [6] studied the use of MDS codes for straggler mitigation in *linear* distributed machine learning algorithms. Tandon et al. [15] introduced a gradient coding framework for straggler mitigation for distributed gradient descent algorithms. In [16], Dutta et al. proposed new coding techniques that reduce the computation time at the workers side and that accounts for stragglers. In a related context, Li et al. [17] studied the effect of the workers' computation load on the communication complexity.

To the extent of our knowledge, this paper is the first to consider straggler mitigation in distributed computing systems under privacy constraints. The work that is closest to our work is the problem of distributively multiplying two private matrices under information theoretic privacy constraints in [7]. Our work can also be related to the work on privacy-preserving algorithms, e.g., [18], [19]. However, the privacy constraint in this line of work is computational privacy, and the proposed algorithms are not designed for straggler mitigation.

Contributions: We consider the distributed computing setting described above in which we require the workers to learn no information about the Master's data. We study the waiting time of the Master, i.e., the aggregate delays caused by the workers. The novelty in our work is in the use of Staircase codes that allow decoding flexibility at the Master, which translates into delay reduction. Assuming an exponential model for the workers response time, we make the following contributions: *(i)* we derive an upper and a lower bound on the mean waiting time (Theorem 1) ; *(ii)* we derive an integral expression leading to the CDF of the waiting time (Theorem 2) and use this expression to find the exact mean waiting time for the cases when $k = n - 1$ and $k = n - 2$ (Corollary 1); and *(iii)* we compare our approach to the approach using secret sharing and show that for high rates, k/n, and small number of workers our approach can save around 40% of the waiting time. Moreover, we ran simulations to check the tightness of the theoretical bounds.

II. System Model

We consider a server M which wants to perform intensive computations on confidential data represented by an $m \times \ell$ matrix A (typically $m >> \ell$). M divides these computations into smaller computational tasks and assigns them to n workers $\mathtt{W}_i$, $i = 1, \ldots, n$, that can perform these tasks in parallel.

Computations model: We focus on linear computations. The motivation is that a building block in several iterative machine learning algorithms, such as gradient descent, is the multiplication of A by a sequence of $\ell \times 1$ attribute vectors $\mathbf{x}^1, \mathbf{x}^2, \ldots$. In the sequel, we focus on the multiplication $A\mathbf{x}$ with one attribute vector $\mathbf{x}$.

Workers model: The workers have the following properties: *1)* The workers incur random delays while executing the task assigned to them by M resulting in what is known as the straggler problem [4]–[6]. We model all the delays incurred by each worker by an independent and identical exponential random variable. *2)* The workers do not collude, i.e., they do not share with each other the data they receive from M. This has implications on the privacy constraint described later.

General scheme: M encodes A, using randomness, into n shares S_i sent to worker $\mathtt{W}_i$, $i = 1, \ldots, n$. Any k or more shares can decode A. The workers obtain zero information about A, i.e., $H(A|S_i) = H(A)$ for all $i \in \{1, \ldots, n\}$.
At each iteration, the master sends $\mathbf{x}$ to all the workers. Then, each worker computes $S_i\mathbf{x}$ and sends it back to the master. Since the scheme and the computations are linear, the master can decode $A\mathbf{x}$ after receiving enough responses[1]. We refer to such scheme as an (n, k) system.

Encoding: We consider classical secret sharing codes [8], [9] and universal Staircase codes [10]. Due to lack of space we

[1]In some cases the attribute vectors $\mathbf{x}^j$ contain information about A, and therefore need to be hidden from the workers. We describe in [20] how our scheme can be generalized to such cases.

only describe their properties that are necessary for performing the delay analysis. Secret sharing codes require the division of A into $k-1$ row blocks and encodes them into n shares of dimension $m/(k-1) \times \ell$ each. Any k shares can decode A. Whereas, Staircase codes require the division of A into $(k-1)\alpha$ row blocks, $\alpha = \text{LCM}\{k, \dots, n-1\}$, and encodes them into n shares. Each share consists of α subshares and is of dimension $m/(k-1) \times \ell$. Any $(k-1)/(d-1)$ fraction of any d shares can decode A, where $d \in \{k, \dots, n\}$. We show that Staircase codes outperform classical codes in terms of incurred delays.

Delay model: Let T_A be the random variable representing the time spent to compute $A\mathbf{x}$ at one worker. We assume a mother runtime distribution $F_{T_A}(t)$ that is exponential[2] with rate λ. Due to the encoding, each task given to a worker is $k-1$ times smaller than A. Let $T_i,\ i \in \{1, \dots, n\}$ denote the time spent by worker $\mathtt{W}_i$ to execute its task, then we assume that F_{T_i} is a scaled distribution of F_{T_A}, i.e.,

$$F_{T_i}(t) \triangleq F_{T_A}((k-1)t) = 1 - e^{-(k-1)\lambda t}.$$

For an (n, k) system using Staircase codes, we assume that T_i is evenly distributed between the subshares, i.e., the time spent by a worker $\mathtt{W}_i$ on one subshare is equal to T_i/α. Let $T_{(i)}$ be the i^{th} order statistic of the T_i's and T_{SC} be the time the master waits until it can decode $A\mathbf{x}$. We can write

$$T_{\text{SC}} = \min_{d \in \{k,\dots,n\}} \left\{ \frac{k-1}{d-1} T_{(d)} \right\} \triangleq \min_{d \in \{k,\dots,n\}} \alpha_d T_{(d)},$$

where $\alpha_i \triangleq (k-1)/(i-1)$. For an (n, k) system using classical secret sharing codes, we can write $T_{\text{SS}} = T_{(k)}$.

III. Main Results

Our main results are summarized as follows. We provide an upper bound and a lower bound on the mean waiting time of $\mathtt{M}$ in Theorem 1.

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

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

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

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

Discussion: Our extensive simulations show that (1) is a good approximation of the mean waiting time. Moreover, by taking $d = k$ in (1), the upper bound on the mean waiting time of Staircase codes becomes the one of classical secret sharing, i.e.,

[2]Our analysis can be generalized to the shifted exponential model used in [6], [13] as we detail in [20].

$$\mathbb{E}[T_{\text{SC}}] \leq \mathbb{E}[T_{\text{SS}}] = \frac{H_n - H_{n-k}}{\lambda(k-1)}. \tag{3}$$

While finding the exact expression of the mean waiting time for any (n, k) system remains open, we derive in Corollary 1 an expression for systems with 1 and 2 parities, i.e. $(k+1, k)$ and $(k+2, k)$ systems, using the result of Theorem 2. Using Corollary 1 one can compare the performance of Staircase codes an secret sharing codes. For instance, in a $(4, 2)$ system Staircase codes reduce the mean waiting time by 40%.

Theorem 2. *Let $t_i \triangleq t(i-1)/(k-1)$, the CDF of the waiting time T_{SC} of an (n, k) system using Staircase codes is given by*

$$F_{T_{SC}}(t) = 1 - n! \int_{y \in A(t)} \frac{F(y_k)^{k-1}}{(k-1)!} dF(y_n) \cdots dF(y_k), \tag{4}$$

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

To check the tightness of the bounds we plot in Figure 2 the upper bound in (1), lower bound in (2) and the exact mean waiting time in (17) for $(k+2, k)$ systems.

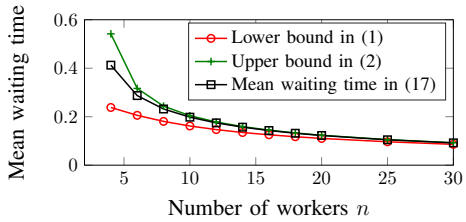


Fig. 2: Bounds on mean waiting time $\mathbb{E}[T_{\text{SC}}]$ for $(k+2, k)$ systems with $\lambda = 1$.

Asymptotics: To better understand the above results, we look at the asymptotic behavior of the lower and upper bounds when n goes to infinity in two regimes: 1) For a constant number of parities $r = n - k$. The mean waiting time of the system is given by $\lim_{n\to\infty} \mathbb{E}[T_{SC}] = \mathbb{E}[T_{SS}]$. Meaning, in this regime there is no advantage in using Staircase codes (Figure 2). 2) For a fixed rate $R = k/n$. The mean waiting time can be bounded by $\mathbb{E}[T_{SC}] \leq \log\left(1/(1-c)\right)/(\lambda(nc-1))$, where c is a constant satisfying $R \leq c < 1$. In this regime, the mean waiting time of systems using Staircase codes is smaller by a constant factor s, $s < 1/R$, than systems using classical secret sharing codes (Figure 3b).

IV. Proof of Theorem 1

We will need the following characterization of order statistics for *iid* exponential random variables.

Theorem 3 (Renyi [21])**.** *The d^{th} order statistic $T_{(d)}$ of n iid exponential random variables T_i, with distribution function $F(t) = 1 - e^{-\lambda t}$, is equal to a random variable Z in the distribution, where*

$$T_{(d)} \triangleq \sum_{j=0}^{d-1} \frac{Z_j}{n-j},$$

and Z_j are iid random variables with distribution $F(t)$.

A. Upper bound on the mean waiting time

We use Jensen's inequality to upper bound the mean waiting time $\mathbb{E}[T_{\text{SC}}]$. The exact mean waiting time is given by

$$\mathbb{E}[T_{\text{SC}}] = \mathbb{E}\left[\min_{d\in\{k,\ldots,n\}}\left\{\frac{k-1}{d-1}T_{(d)}\right\}\right].$$

Since $\min$ is a convex function, we can use Jensen's inequality to write

$$\mathbb{E}\left[\min_{d\in\{k,\ldots,n\}}\left\{\frac{k-1}{d-1}T_{(d)}\right\}\right] \leq \min_{d\in\{k,\ldots,n\}}\left\{\mathbb{E}\left[\frac{k-1}{d-1}T_{(d)}\right]\right\}. \tag{5}$$

The average of the d^{th} order statistic $\mathbb{E}[T_{(d)}]$ can be written as

$$\mathbb{E}[T_{(d)}] = \mathbb{E}[Z_i]\sum_{j=0}^{d-1}\frac{1}{n-j} = \frac{H_n - H_{n-d}}{\lambda(k-1)} \tag{6}$$

Equations (5) and (6) conclude the proof. We give an intuitive behavior of the upper bound. The harmonic number can be approximated by $H_n \approx \log(n) + \gamma$, where $\gamma \approx 0.577218$ is called the Euler-Mascheroni constant. Therefore, $\log(n) < H_n < \log(n+1)$. Hence, we can write

$$\mathbb{E}[T_{\text{SC}}] < \min\left\{\begin{array}{l}\min\limits_{d\in\{k,\ldots,n-1\}}\left\{\frac{1}{\lambda(d-1)}\log\left(\frac{n+1}{n-d}\right)\right\},\\ \frac{1}{\lambda(n-1)}\log(n+1)\end{array}\right\}. \tag{7}$$

B. Lower bound on the mean waiting time

To lower bound the mean waiting time $\mathbb{E}[T_{\text{SC}}]$, we find the probability distribution of a small (sufficient) set of conditions that result in $T_{\text{SC}} > t$. This distribution serves as a lower bound on the exact distribution of T_{SC}. For a given $d \in \{k,\ldots,n\}$, consider the following set of conditions

$$\mathcal{C} \triangleq \left\{T_{(k)} > \frac{t}{\alpha_d}\right\} \bigcap_{j=d+1}^{n}\left\{T_{(j)} - T_{(j-1)} > \frac{t}{\alpha_j} - \frac{t}{\alpha_{j-1}}\right\},$$

where $\alpha_j \triangleq (k-1)/(j-1)$. For T_{SC} to be greater than t, all the j^{th} order statistic $T_{(j)}$'s must be greater than t/α_j for $j \in \{k,\ldots,n\}$. We show that if $\mathcal{C}$ is satisfied, then the previous condition is satisfied. If $T_{(k)} > t/\alpha_d$, then $T_{(i)} > t/\alpha_i$ for all $i \in \{k,\ldots,d\}$, because $T_{(i)} \geq T_{(k)} > t/\alpha_d > t/\alpha_i$. It follows that if for all $j \in \{d+1,\ldots,n\}$, $T_{(j)} - T_{(j-1)} > t/\alpha_j - t/\alpha_{j-1}$, then $T_{(j)} > t/\alpha_j$. Therefore, $\Pr(T_{\text{SC}} > t) \geq \Pr(\mathcal{C} \text{ is satisfied}) \triangleq \Pr(\mathcal{C})$. Furthermore,

$$\mathbb{E}[T_{\text{SC}}] = \int_0^\infty \Pr(T_{\text{SC}} > t)\,dt \geq \int_0^\infty \Pr(\mathcal{C})dt. \tag{8}$$

Next we derive an expression of $\int_0^\infty \Pr(\mathcal{C})\,dt$. Note that $1/\alpha_j - 1/\alpha_{j-1} = 1/(k-1)$, using Theorem 3 we can write

$$\Pr\left\{T_{(j)} - T_{(j-1)} > \frac{t}{k-1}\right\} = \bar{F}_{Z_j}\left(\frac{(n-j+1)t}{k-1}\right), \tag{9}$$

where $\bar{F}_{Z_j}(t) \triangleq \Pr(Z_j > t)$. From (9) we get

$$\Pr(\mathcal{C}) = \bar{F}_{T_{(k)}}\left(\frac{t}{\alpha_d}\right)\prod_{j=d+1}^{n}\bar{F}_{Z_j}\left(\frac{(n-j+1)}{(k-1)}t\right) \tag{10}$$

Since $\bar{F}_{Z_j}(t) = e^{-(k-1)\lambda t}$, we can write

$$\begin{aligned}\prod_{j=d+1}^{n}\bar{F}_{Z_j}\left(\frac{(n-j+1)}{(k-1)}t\right) &= \bar{F}_{Z_j}\left(\sum_{j=d+1}^{n}\frac{(n-j+1)}{(k-1)}t\right)\\ &= \bar{F}_{Z_j}\left(t\frac{(n-d)(n-d+1)}{2(k-1)}\right).\end{aligned} \tag{11}$$

On the other hand, $\bar{F}_{T_{(k)}}(t/\alpha_d)$ is the probability that there are at most $k-1$ T_i's less than t/α_d, therefore

$$\bar{F}_{T_{(k)}}\left(\frac{t}{\alpha_d}\right) = \sum_{i=0}^{k-1}\binom{n}{i}F_{T_i}\left(\frac{t}{\alpha_d}\right)^i\bar{F}_{T_i}\left(\frac{t}{\alpha_d}\right)^{n-i}. \tag{12}$$

Recall that $F_{T_i}(t) = 1 - e^{(k-1)\lambda t} = 1 - \bar{F}_{T_i}(t)$, therefore by using the binomial expansion we can write

$$F_{T_i}\left(\frac{t}{\alpha_d}\right)^i = \sum_{j=0}^{i}\binom{i}{j}(-1)^j\bar{F}_{T_i}\left(\frac{t}{\alpha_d}\right)^j. \tag{13}$$

Using (13) and the fact that $F_{T_i}(t) = e^{-(k-1)\lambda t}$, (12) becomes

$$\bar{F}_{T_{(k)}}\left(\frac{t}{\alpha_d}\right) = \sum_{i=0}^{k-1}\binom{n}{i}\sum_{j=0}^{i}\binom{i}{j}(-1)^j\bar{F}_{T_i}\left(t\frac{(n-i+j)(d-1)}{(k-1)}\right). \tag{14}$$

Combining (11) and (14) and noting that $\bar{F}_{T_i}(t) = \bar{F}_{Z_j}(t) = e^{-\lambda(k-1)t}$, (10) becomes

$$\begin{aligned}\Pr(\mathcal{C}) = \sum_{i=0}^{k-1}\binom{n}{i}\sum_{j=0}^{i}\binom{i}{j}(-1)^j\exp\big(&-\lambda t(n-i+j)(d-1)\\ &-\lambda t(n-d)(n-d+1)/2\big).\end{aligned} \tag{15}$$

Note that $\int_0^\infty e^{-xt}dt = 1/x$ and that the integral of a sum is equal to the sum of the integrals. Therefore, integrating (15) from 0 to ∞ and maximizing it over all values of d, $d \in \{k,\ldots,n\}$, concludes the proof.

V. Proof of Theorem 2

We derive an integral expression leading to the probability distribution of the waiting time T_{SC}. Since the delays at the workers' side T_i's are independent and are absolutely continuous with respect to the Lebesgue measure (i.e. the probability density exists), we have

$$\begin{aligned}f_{T_{(1)},\ldots,T_{(n)}}(t_1,\ldots,t_n) &= n!\prod_{i=1}^{n}f_{T_i}(t_i)\\ &= n!\lambda^n(k-1)^n\exp\left(-\lambda(k-1)\sum_{i=1}^{n}t_i\right),\end{aligned}$$

where t_i denotes t/α_i and $0 \leq t_1 \leq \ldots \leq t_n$. Therefore we can write the distribution of T_{SC} as

$$\Pr\{T_{\text{SC}} > t\} = \Pr\bigcap_{d=k}^{n}\{T_{(d)} > t_d\} = \int_{A(t)}f_{T_{(1)},\ldots,T_{(n)}}(y)dy,$$

$$\mathbb{E}\left[T_{\text{SC}}\right] = \frac{1}{\lambda}\sum_{i=2}^{k+1}(-1)^i\binom{k+1}{i}\left[\frac{i}{k+(k-1)(i-1)} - \frac{1}{ki}\right]. \tag{16}$$

$$\mathbb{E}[T_{\text{SC}}] = \sum_{i=2}^{k+2}\frac{(-1)^i\binom{k+2}{i}}{\lambda}\left[\frac{i}{(k+1)+k(i-1)} - \frac{1}{(k+1)i} + \frac{i(i-1)}{4(k+1)+2(k-1)(i-2)} - \frac{i(i-1)}{(2k+1)+(k-1)(i-2)}\right]. \tag{17}$$

where $y_{n+1} = \infty$ and

$$\begin{aligned} A(t) &= \{0 \le y_1 \le \ldots \le y_n : y_d > t_d, \text{ for } k \le d \le n\} \\ &= \cap_{i\ge k}\{y_i \in (t_i, y_{i+1}]\} \cap_{i<k} \{y_i \in [0, y_{i+1}]\}. \end{aligned}$$

That is, we can re-write $\Pr\{T_{\text{SC}} > t\}$ as

$$n!\int_{t_n}^{\infty}\cdots\int_{t_k}^{y_{k+1}}\prod_{i=k}^{n} dF_{T_i}(y_i)\left(\int_0^{y_k}\cdots\int_0^{y_2}\prod_{i=1}^{k-1} dF_{T_i}(y_i)\right).$$

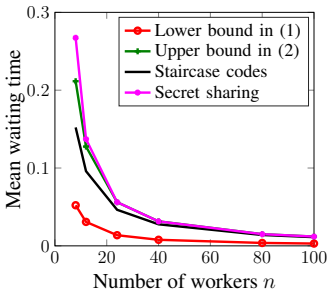
Claim 1. $\int_0^{y_k}\cdots\int_0^{y_2}\prod_{i=1}^{k-1} dF_{T_i}(y_i) = \frac{F(y_k)^{k-1}}{(k-1)!}$.

The result of Claim 1 is straightforward, it follows from integrating $k-1$ times the complementary CDF of an exponential random variable in respect to its derivative. This completes the proof. A more detailed proof of Claim 1 can be found in [20]. We state the mean waiting time for the $(k+2,k)$ and $(k+1,k)$ systems in Corollary 1.

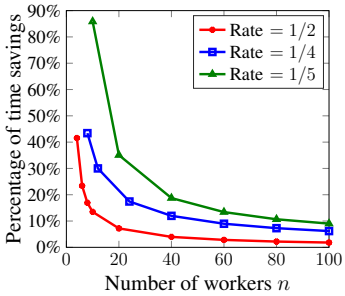
Corollary 1. *The mean waiting time* $\mathbb{E}[T_{SC}]$ *for* $(k+1,k)$ *and* $(k+2,k)$ *systems are given by* (16) *and* (17)*, respectively.*

VI. Simulations

We check the tightness of the bounds of Theorem 1 and measure the improvement, in terms of delays, of Staircase codes over classical secret sharing codes for systems with fixed rate $R \triangleq k/n$. In Figure 3 (a) we plot the upper bound (1), lower bound (2) and the simulated mean waiting time for $R = 1/4$. Our extensive simulations show that the upper bound is a good approximation of the exact mean waiting time, whereas the lower bound might be loose.



(a) Waiting time for systems with rate $R = 1/4$.



(b) Delay reduction using Staircase codes.

Fig. 3: Simulations for (n,k) systems with fixed rate.

Figure 3 (b) aims to better understand the comparison between Staircase codes and classical codes. We plot the normalized difference between the mean waiting times, i.e., $\left(\mathbb{E}[T_{\text{SS}}] - \mathbb{E}[T_{\text{SC}}]\right) / \left(\mathbb{E}[T_{\text{SS}}]\right)$, for different rates. For high rates, Staircase codes offer high savings for small values of n, whereas for low rates Staircase codes offer high savings for all values of n.

References

[1] https://setiathome.berkeley.edu.

[2] https://foldingathome.stanford.edu.

[3] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

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

[5] G. Ananthanarayanan, S. Kandula, A. G. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris, "Reining in the outliers in map-reduce clusters using mantri.," in *OSDI*, vol. 10, p. 24, 2010.

[6] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *arXiv preprint arXiv:1512.02673*, 2015.

[7] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, (New York, NY, USA), pp. 48–59, ACM, 2010.

[8] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[9] R. J. McEliece and D. V. Sarwate, "On sharing secrets and reed-solomon codes," *Communications of the ACM*, vol. 24, no. 9, pp. 583–584, 1981.

[10] R. Bitar and S. El Rouayheb, "Staircase codes for secret sharing with optimal communication and read overheads," in *IEEE International Symposium on Information Theory (ISIT)*, 2016.

[11] L. Huang, S. Pawar, H. Zhang, and K. Ramchandran, "Codes can reduce queueing delay in data centers," in *IEEE International Symposium on Information Theory (ISIT)*, 2012.

[12] G. Joshi, Y. Liu, and E. Soljanin, "Coding for fast content download," in *50th Annual Allerton Conference on Communication, Control, and Computing*, 2012.

[13] 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.

[14] S. Kadhe, E. Soljanin, and A. Sprintson, "Analyzing the download time of availability codes," in *IEEE International Symposium on Information Theory (ISIT)*, 2015.

[15] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding," in 29*th Conference on Neural Information Processing Systems (NIPS)*, 2016.

[16] S. Dutta, V. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," in 29*th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 2092–2100, 2016.

[17] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental tradeoff between computation and communication in distributed computing," in *IEEE International Symposium on Information Theory (ISIT)*, 2016.

[18] H. Takabi, E. Hesamifard, and M. Ghasemi, "Privacy preserving multi-party machine learning with homomorphic encryption," in 29*th Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.

[19] R. Hall, S. E. Fienberg, and Y. Nardi, "Secure multiple linear regression based on homomorphic encryption," *Journal of Official Statistics*, vol. 27, no. 4, p. 669, 2011.

[20] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure distributed computing," *arXiv preprint arXiv:1703.01504*, 2017.

[21] A. Rényi, "On the theory of order statistics," *Acta Mathematica Academiae Scientiarum Hungarica*, vol. 4, no. 3-4, pp. 191–231, 1953.