A Group-Type Distributed Coded Computation Scheme Based on a Gabidulin Code

Koki Kazama
Waseda University
Email: kokikazama@aoni.waseda.jp

Toshiyasu Matsushima
Waseda University
Email: toshimat@waseda.jp

***Abstract*—We focus on a distributed coded computation scheme for matrix multiplication. In this system, the product matrix is encoded and decoded through the overall system to correct errors in computation. We propose a group-type distributed coded computation scheme, for one example, a scheme based on a Gabidulin code, and evaluate the computation time complexity and the error-correcting capability of the overall system.**

The full version of this paper is in [1].

I. Introduction

We focus on a distributed computation scheme in which errors are corrected in computing multiplication of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ on a finite field $\mathbb{F}_q$. In the system of a distributed computation scheme, the main computer (master) partitions the matrix and distributes them to multiple computers (workers), and (2) workers perform parallel computing. This system has the advantage of decreasing the computation time complexity, while this has the disadvantage of increasing the possibility of occurring errors in computation. To eliminate the disadvantage, a distributed coded computation scheme (DCC) [2], [3], [4], [5] uses an error-correcting code (ECC) to correct errors in distributed computation. On performance evaluation of DCCs, (a) computation time complexity (CTC) and (b) error-correcting capability of the overall system are important criteria while they are a tradeoff. Considering the reason to construct DCCs, we would like to propose the DCC which can correct some errors and compute $\boldsymbol{AB}$ more efficiently than the stand-alone scheme (SA), of which the system computes $\boldsymbol{AB}$ solely. Here all errors form a matrix $\boldsymbol{E}$, which is called an error matrix. The DCCG (DCCG) corrects $\boldsymbol{E}$ whose all entries are nonzero if it satisfies some conditions of column-dependency, while the other previous schemes cannot correct them.

In this paper, we propose a new distributed coded computation scheme called *Group-Type Distributed Coded Computation scheme* (GDCC) and, as one example, a *GDCC based on a Gabidulin code* (GDCCG). In these schemes, workers are equally partitioned into multiple groups and the parallel computations of matrix multiplications are performed within each group. A Gabidulin code encodes a matrix over $\mathbb{F}_q$ and correct an error matrix $\boldsymbol{E}$ if $\mathrm{rank}(\boldsymbol{E}) \le t$, where t is a constant defined later. The key idea is that computing an inner product of two vectors over an extension field $\mathbb{F}_{q^m}$ can be decomposed to computations over $\mathbb{F}_q$, which can be parallelly performed. Using these facts, this distributed computing system performs a process over $\mathbb{F}_q$, which is equivalent to encoding a vector over $\mathbb{F}_{q^m}$ corresponding $\boldsymbol{AB}$ to a codeword over $\mathbb{F}_{q^m}$. Thus the encoder in the GDCCG system encodes all columns of$\boldsymbol{AB}$ together while the encoders in previous DCC systems encode each column of $\boldsymbol{AB}$. This enables to correct error matrices whose columns depend on each other and enables to decrease the overall CTC of the system simultaneously. The GDCCG is a little different from the DCCG because more workers are used for parallel computations. Thus the computation time complexity of the GDCCG is less than that of the DCCG, while the error-correcting capability of the GDCCG is the same as that of the DCCG.

Moreover, we evaluate (a) the CTC and (b) the error-correcting capability of the GDCC in detail and show the advantages as follows. In the evaluation of (a), we evaluate the CTC of the GDCCG and the SA and we show the condition of parameters (the number of workers and groups) in which the GDCGC is superior to the SA. We also explain that the GDCCG is also superior to the DCCG just a little. We cannot generally evaluate (a) of the GDCC because the CTC of the decoding algorithm is depending on the code. Thus we evaluate (a) the CTC only of the GDCCG. We define the CTC of a DCC system as the number of four arithmetic operations (an addition, a subtraction, a multiplication, and an inversion[1])over $\mathbb{F}_q$ in parallel computing of each worker and decoding of the master. To evaluate the number of operations of the encoder and the decoder using a Gabidulin code over $\mathbb{F}_{q^m}$ in the GDCCG system, we first show how many times it is necessary when the operation of $\mathbb{F}_{q^m}$ is decomposed into the operation of $\mathbb{F}_q$, and then we enumerate them. In the evaluation of (b), we show what error matrices the GDCC system and the GDCCG system can correct, respectively. Specifically, we show that the GDCCG system can correct an error matrix if $\mathrm{rank}(\boldsymbol{E}) \le t$, where t is a certain constant.

II. The Purpose of Constructing a Distributed Coded Computation Scheme

As a preliminary, we define notations and explain the purpose of distributed coded computation scheme. Let $\mathbb{N}$

This research is supported in part by Grant-in-Aid JP17K06446 for Scientific Research (C).

[1]An inversion is an operation of computing a^{-1} from a. This indicates that an operation of computing ab^{-1} is a combination of an inversion b^{-1} and a multiplication ab^{-1}.

denote the set of all positive integers. We define $[m,n] := \{ m, m+1, \ldots, n \}$ for two integers m and n with $m \leq n$. $[n]$ denotes $[1,n]$ for $n \in \mathbb{N}$. All vectors are column vectors except specifically noted. $\boldsymbol{E}^\top$ is the transposeof a matrix $\boldsymbol{E}$. $\mathbb{F}_q$ is a finite field with q elements, where q is a power of 2. $\mathbb{F}_q^{n\times m}$ denotes the set of all $n\times m$ matrices over $\mathbb{F}_q$, and $\mathbb{F}_q^n := \mathbb{F}_q^{n\times 1}$. $\boldsymbol{e}_{\cdot j} \in \mathbb{F}_q^n$ denotes the j-th column of a matrix $\boldsymbol{E} \in \mathbb{F}_q^{n\times m}$. $\boldsymbol{e}_{i\cdot} \in \mathbb{F}_q^m$ denotes the transpose of the i-th row vector. Thus $\boldsymbol{E} = (\boldsymbol{e}_{\cdot 1}, \ldots, \boldsymbol{e}_{\cdot b}) = (\boldsymbol{e}_{1\cdot}, \ldots, \boldsymbol{e}_{n\cdot})^\top$. For $m, n, k_{\mathrm{A}} \in \mathbb{N}, A \subset [n], \boldsymbol{G} \in \mathbb{F}_{q^m}^{n\times k_{\mathrm{A}}}$, we define $\boldsymbol{G}_{A\cdot}^\top \in \mathbb{F}_{q^m}^{|A|\times k_{\mathrm{A}}}$ as a matrix constructed from all $i(\in A)$-th row $\boldsymbol{g}_{i\cdot}^\top \in \mathbb{F}_{q^m}^{1\times k_{\mathrm{A}}}$. ? denotes the symbol of an erasure or decoding failure. We define the sum and difference of any $a \in \mathbb{F}_q$ and ? as ?.

Definition 2.1 ($\boldsymbol{v}$, $\boldsymbol{f}^s$): Let $v_1, \ldots, v_m \in \mathbb{F}_{q^m}$ be linearly independent over $\mathbb{F}_q$ and $v_i = v_1^{q^{i-1}}$ for any $i \in [m]$. $\{v_1, \ldots, v_m\}$ is a normal basis [6] of a linear space $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Let $\boldsymbol{v}$ denote a vector $(v_1, \ldots, v_m)$. For any $s \in \mathbb{N}$, let $\boldsymbol{f}^s \colon \mathbb{F}_{q^m}^s \to \mathbb{F}_q^{s\times m}$ be a linear homomorphism over $\mathbb{F}_q$ such that, from the input $\boldsymbol{x} \in \mathbb{F}_{q^m}^s$, $\boldsymbol{f}^s$ outputs a unique matrix $\boldsymbol{X} \in \mathbb{F}_q^{s\times m}$ such that $\boldsymbol{x} = \boldsymbol{X}\boldsymbol{v}$. $f^1 := \boldsymbol{f}^1$.

We explain the definition of computation time complexity.

Assumption 2.1: Through this paper, q is a power of 2. Positive integers $n, k_{\mathrm{A}}, k_{\mathrm{B}}, l, m$ satisfy $2 \leq k_{\mathrm{A}} < n, 2 \leq k_{\mathrm{B}}$, $m = \max\{k_{\mathrm{B}}, n\}$, and $2 \leq l$.

In this paper, we consider schemes for computing a multiplication of two matrices, $\boldsymbol{A} \in \mathbb{F}_q^{k_{\mathrm{A}}\times l}$ and $\boldsymbol{B} \in \mathbb{F}_q^{l\times k_{\mathrm{B}}}$. One value of $\boldsymbol{A}$ is input to the master only once, and the master store this value. On the other hand, many values of $\boldsymbol{B}$ are input to the master many times, and each time the master attempts to compute the value of the matrix $\boldsymbol{AB}$.

The most simple scheme is as follows.

Definition 2.2: We define a *stand-alone scheme* (SA) as a scheme in which the master computes $\boldsymbol{AB}$ solely from the input $\boldsymbol{A}, \boldsymbol{B}$.

Definition 2.3: We define *computation time complexity* (CTC) of a process as the number of four arithmetic operations over $\mathbb{F}_q$ which the system performs in the process from input to output. A subtraction, an addiction, a multiplication, and an inversion are equally treated as one operation in the evaluation of the CTC. *CTC of the system* is the overall CTC from input $\boldsymbol{B}$ to output $\hat{\boldsymbol{AB}}$. [2]

Proposition 2.1: The CTC of the SA system is $k_{\mathrm{A}} k_{\mathrm{B}}(2l-1)$.

We would like to compute $\boldsymbol{AB}$ more efficiently than the SA system, i.e. to construct a computing system whose CTC is less than that of the SA system. For this purpose, we focus on a distributed coded computation scheme. In this system, however, the possibility of errors increases. Thus we use a distributed coded computation scheme to correct errors.

III. Redefinitions of Previous Schemes

Based on [5], we redefine distributed coded computation schemes (DCCs) for computing $\boldsymbol{AB}$ from $\boldsymbol{A}$ and $\boldsymbol{B}$. As examples, we redefine the previous ones [5] and [2].

[2] $\hat{\boldsymbol{AB}}$ may not be $\boldsymbol{AB}$ by errors.

Let $q, k_{\mathrm{A}}, k_{\mathrm{B}}, l, n$ be given. Let $\tilde{\mathcal{C}} \subset \mathbb{F}_{q^m}^n$ be an (n, k_{A}) linear code over $\mathbb{F}_{q^m}$. This code has a generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{n\times k_{\mathrm{A}}}$ of a canonical systematic encoder. $\mathcal{C} := \boldsymbol{f}^n(\tilde{\mathcal{C}})(\subset \mathbb{F}_q^{n\times m})$ is a linear code over $\mathbb{F}_q$. Let $\psi \colon \mathbb{F}_q^{n\times m} \to \mathcal{C} \cup \{?\}$ be the decoder of $\mathcal{C}$. Let a set $\mathcal{E}$ $(\subset \mathbb{F}_q^{n\times m})$ be the set of all error matrices which can be corrected rightly, i.e. for any $\boldsymbol{E} \in \mathcal{E}$ and $\boldsymbol{C} \in \mathcal{C}$, $\psi(\boldsymbol{C} + \boldsymbol{E}) = \boldsymbol{C}$.

Lemma 3.1: Any codeword of a linear code $\mathcal{C} = \boldsymbol{f}^n(\tilde{\mathcal{C}})$ over $\mathbb{F}_q$ is $\boldsymbol{f}^n(\boldsymbol{GMv})$ for some matrix $\boldsymbol{M} \in \mathbb{F}_q^{k_{\mathrm{A}}\times m}$.

Proof : A codeword an (n, k_{A}) linear code $\tilde{\mathcal{C}} \subset \mathbb{F}_{q^m}^n$ over $\mathbb{F}_{q^m}$ is $\boldsymbol{GMv}$ for some matrix $\boldsymbol{M} \in \mathbb{F}_q^{k_{\mathrm{A}}\times m}$. □

Definition 3.1 (Gabidulin code [7]): Let $h_1, \ldots, h_n \in \mathbb{F}_{q^m}$ be $h_i = v_i$ for any $i \in [n]$, where $\{v_1, \ldots, v_m\}$ is in Definition 2.1. Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{n\times(n-k_{\mathrm{A}})}$ is a matrix whose (i,j)-th entry is $h_i^{q^{j-1}}$ for any $(i,j) \in [n] \times [k_{\mathrm{A}}]$. An (n, k_{A}) linear code $\tilde{\mathcal{C}}_{\mathrm{G}} \subset \mathbb{F}_{q^m}^n$ over $\mathbb{F}_{q^m}$ which has a parity check matrix $\boldsymbol{H}$ is called an (n, k_{A}) *Gabidulin code* over $\mathbb{F}_{q^m}$.

A DCC $(\pi, \boldsymbol{G}, \psi)$ is defined as follows. In the system of a DCC, the master and workers $1, \ldots, n$ are used. The master has a full rank matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{n\times k_{\mathrm{A}}}$ and a function $\psi \colon \mathbb{F}_q^{n\times m} \to \mathcal{C} \cup \{?\}$. All workers have a function $\pi \colon \mathbb{F}_{q^m}^l \times \mathbb{F}_q^{l\times k_{\mathrm{B}}} \to \mathbb{F}_q^m$. The flow is as follows.

⟨*Preprocess*⟩ When the matrix $\boldsymbol{A}$ is input to the master, the master encodes $\boldsymbol{A}$ to the matrix $\boldsymbol{GA} \in \mathbb{F}_{q^m}^{n\times l}$. $\boldsymbol{g}_{i\cdot}^\top \boldsymbol{A} \in \mathbb{F}_{q^m}^{1\times l}$, where $\boldsymbol{g}_{i\cdot}^\top$ is the i-th row of $\boldsymbol{G}$, is stored in each worker $i \in [n]$.

⟨*Computing Process*⟩ When the matrix $\boldsymbol{B}$ is input to the master, the master sends $\boldsymbol{B}$ to all workers. We define $\boldsymbol{B}' = \boldsymbol{B}$ if $n \leq k_{\mathrm{B}}$, and $\boldsymbol{B}' = (\boldsymbol{B}, \boldsymbol{0})$ if $n > k_{\mathrm{B}}$, where $\boldsymbol{0}$ is an $l \times (m - k_{\mathrm{B}})$ zero matrix over $\mathbb{F}_q$. Each worker i computes $\pi(\boldsymbol{g}_{i\cdot}^\top \boldsymbol{A}, \boldsymbol{B}) := f^1(\boldsymbol{g}_{i\cdot}^\top \boldsymbol{AB}'\boldsymbol{v}) \in \mathbb{F}_q^{1\times m}$ from $\boldsymbol{g}_{i\cdot}^\top \boldsymbol{A}$ and $\boldsymbol{B}$. In the computation of each worker i, the error $\boldsymbol{e}_{i\cdot} \in \mathbb{F}_q^m$ occurs and the result is $\boldsymbol{y}_{i\cdot} := \pi\left(\boldsymbol{g}_{i\cdot}^\top \boldsymbol{A}, \boldsymbol{B}\right) + \boldsymbol{e}_{i\cdot}$.

⟨*Decoding Process*⟩ The master receives the results $\boldsymbol{y}_{1\cdot}, \ldots, \boldsymbol{y}_{n\cdot}$ of all workers and computes $\psi(\boldsymbol{Y})$ from the matrix $\boldsymbol{Y} := (\boldsymbol{y}_{1\cdot}, \ldots, \boldsymbol{y}_{n\cdot})^\top \in \mathbb{F}_q^{n\times m}$ by a function ψ. A bijection exists betwee two sets $\mathcal{C}(\subset \mathbb{F}_q^{n\times m})$ and $\mathbb{F}_q^{k_{\mathrm{A}}\times l}$. If $\psi(\boldsymbol{Y}) \in \mathcal{C}$, then the master $\hat{\boldsymbol{AB}}$ from the matrix $\psi(\boldsymbol{Y})$ by this bijection.

Definition 3.2: Let $\Pi \colon \mathbb{F}_q^{k_{\mathrm{A}}\times k_{\mathrm{B}}} \to \mathcal{C}$ be a function such that $\Pi(\boldsymbol{AB}) := \boldsymbol{f}^n(\boldsymbol{GAB}'\boldsymbol{v})$. Let $\boldsymbol{E} := (\boldsymbol{e}_{1\cdot}, \ldots, \boldsymbol{e}_{n\cdot})^\top$. $\Pi(\boldsymbol{AB})$, $\boldsymbol{E}, \boldsymbol{Y}, \boldsymbol{G}, \Pi, \psi$ and $\mathcal{C}$ are called a *codeword (matrix),* an *error (matrix),* a *received matrix,* a *generator matrix,* an *encoder*, a *decoder* and a *code*. Clearly $\boldsymbol{Y} = \boldsymbol{f}^n(\boldsymbol{GAB}'\boldsymbol{v}) + \boldsymbol{E}$.

If $\boldsymbol{G} \in \mathbb{F}_q^{n\times k_{\mathrm{A}}}$ is a generator matrix of a Reed Solomon code over $\mathbb{F}_q$, this DCC is the same as the scheme of [2] since $\boldsymbol{f}^n(\boldsymbol{GAB}'\boldsymbol{v}) = \boldsymbol{GAB}'$ and $f^1(\boldsymbol{g}_{i\cdot}^\top \boldsymbol{AB}'\boldsymbol{v}) = \boldsymbol{g}_{i\cdot}^\top \boldsymbol{AB}'$. If $\boldsymbol{G} \in \mathbb{F}_{q^m}^{n\times k_{\mathrm{A}}}$ is a generator matrix of a Gabidulin code over $\mathbb{F}_{q^m}$, this DCC is the same as the DCCG.

The system of the DCCG can correct an error matrix $\boldsymbol{E}$ with $\mathrm{rank}(\boldsymbol{E}) \leq t$. However, this system needs at least $O(n^2)$ operations over $\mathbb{F}_{q^m}$. We propose a scheme to compute $\boldsymbol{AB}$ more efficiently than the SA scheme and this scheme.

IV. The Proposed Scheme

In this section, we propose a new scheme called a *group-type distributed coded computation scheme* (GDCCG). More-

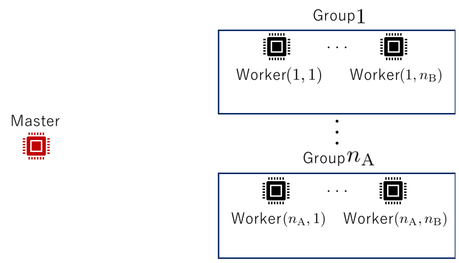

Fig. 1. the master and the workers

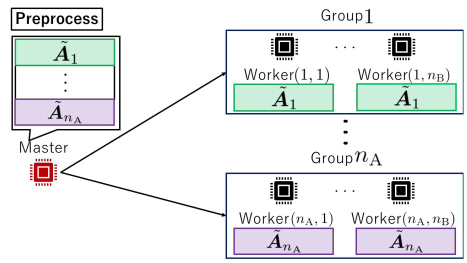

Fig. 3. Preprocess

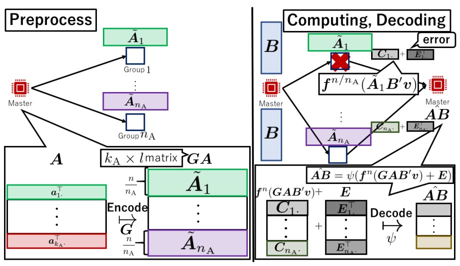

Fig. 2. the flow of the DCC

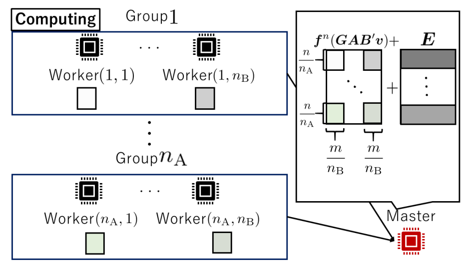

Fig. 4. Computing Process

over, we propose a scheme called *GDCC with a Gabidulin code* (GDCCG) as an example of a GDCC.

This scheme is based on an idea of decomposing operations over an extension field $\mathbb{F}_{q^m}$ to operations over $\mathbb{F}_q$. The system can compute $\boldsymbol{AB}$ efficiently and correct errors by this idea. (The decomposition is used for computing a matrix $\Pi(\boldsymbol{AB})$ from $\boldsymbol{A}$ and $\boldsymbol{B}$ by operations over $\mathbb{F}_q$ instead of computing a vector $\boldsymbol{GAB'v}$ over an extension field $\mathbb{F}_{q^m}$. This computation can be performed by distributed computing.

A. GDCC

We propose a GDCC $((\pi_{ij}|i \in [n_\mathrm{A}], j \in [n_\mathrm{B}]), \boldsymbol{G}, \psi)$ when $q, k_\mathrm{A}, k_\mathrm{B}, l, n, n_\mathrm{A}$ and n_B are given. In this scheme, we use the master and $n_\mathrm{A} n_\mathrm{B}$ workers which are partitioned into multiple groups, where $n_\mathrm{A} \in \mathbb{N}$ devides n and $n_\mathrm{B} \in \mathbb{N}$ devides m. All workers are equally partitioned into n groups. The j-th worker of the i-th group is called *a worker* $(i,j) \in [n_\mathrm{A}] \times [n_\mathrm{B}]$ (Figure 1). We assume that errors occur only when workers compute something. The master has the matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{n \times k_\mathrm{A}}$ mentioned above and a function ψ. A function π_{ij} are stored in each worker $(i,j) \in [n_\mathrm{A}] \times [n_\mathrm{B}]$.

In GDCC, when the matrix $\boldsymbol{A}$ is input to the master, then the master and all workers perform *preprocess*. For any time when the matrix $\boldsymbol{B}$ is input to the master, all workers perform *Computing Process*, and then the master performs *Decoding Process*. The result of Decoding Process of the master is $\hat{\boldsymbol{AB}} \in \mathbb{F}_q^{k_\mathrm{A} \times k_\mathrm{B}}$, which is an estimated results of $\boldsymbol{AB}$. See the details in below (Figure 2).

$\langle$*Preprocess*$\rangle$ (Figure 3) We define $\langle i \rangle_\mathrm{r} := [(i-1)(n/n_\mathrm{A}) + 1, i(n/n_\mathrm{A})]$, $\langle j \rangle_\mathrm{c} := [(j-1)(m/n_\mathrm{B}) + 1, j(m/n_\mathrm{B})]$, and $\tilde{\boldsymbol{A}}_i := \boldsymbol{G}_{\langle i \rangle_\mathrm{r} \cdot}^\top \boldsymbol{A} \in \mathbb{F}_{q^m}^{(n/n_\mathrm{A}) \times l}$. The master encodes $\boldsymbol{A}$ to $\boldsymbol{GA} = (\tilde{\boldsymbol{A}}_1^\top, \ldots, \tilde{\boldsymbol{A}}_{n_\mathrm{A}}^\top)^\top \in \mathbb{F}_{q^m}^{n \times l}$. The master store $\tilde{\boldsymbol{A}}_i$ in all workers of each group i. Let $\tilde{a}_{i'l'm'j'} \in \mathbb{F}_q$ denote the j'-th symbol of $f^1(\boldsymbol{g}_{i' \cdot}^\top \boldsymbol{a}_{\cdot l'} v_{m'}) \in \mathbb{F}_q^{1 \times m}$ for any $l' \in [l]$, $m' \in [m]$, $i' \in \langle i \rangle_\mathrm{r}$ and $j' \in \langle j \rangle_\mathrm{c}$. Then, each worker (i,j) stores the set $\{(i', l', m', j', \tilde{a}_{i'l'm'j'}) \mid l' \in [l], m' \in [m], i' \in \langle i \rangle_\mathrm{r}, j' \in \langle j \rangle_\mathrm{c}\}$. It is clear that $\tilde{a}_{i'l'm'j'}$ can be computed from $\boldsymbol{g}_{i' \cdot}^\top \boldsymbol{A} = (\boldsymbol{g}_{i' \cdot}^\top \boldsymbol{a}_{\cdot 1}, \ldots, \boldsymbol{g}_{i' \cdot}^\top \boldsymbol{a}_{\cdot l})$, where $\boldsymbol{a}_{\cdot l'} \in \mathbb{F}_q^{k_\mathrm{A}}$ is the l'-th column of the matrix $\boldsymbol{A}$.

$\langle$*Computing Process*$\rangle$ (Figure 4) The master sends the matrix $\boldsymbol{B}$ to all workers. we define the j*-th block* of a $(n/n_\mathrm{A}) \times m$ matrix as the $(j-1)(m/n_\mathrm{B}) + 1, j(m/n_\mathrm{B})$-th columns. Each worker (i,j) computes the j-th block of an $(n/n_\mathrm{A}) \times m$ matrix $\boldsymbol{C}_{i \cdot} := \boldsymbol{f}^{n/n_\mathrm{A}} \left(\boldsymbol{G}_{\langle i \rangle_\mathrm{r} \cdot} \boldsymbol{AB'v} \right)$. This computation can be done by computing $\sum_{l' \in [l]} \sum_{m' \in [m]} \tilde{a}_{i'l'm'j'} b_{l'm'}$ from $\boldsymbol{B}$ for any $(i', j') \in \langle i \rangle_\mathrm{r} \times \langle j \rangle_\mathrm{c}$. See proposition 4.1.

For any $(i,j) \in [n_\mathrm{A}] \times [n_\mathrm{B}]$, we define the correct computing result of any worker (i,j) as $\pi_{ij}(\boldsymbol{g}_{i \cdot}^\top \boldsymbol{A}, \boldsymbol{B})$. we define *product function* as the function $\pi_{ij} \colon \mathbb{F}_{q^m}^{(n/n_\mathrm{A}) \times l} \times \mathbb{F}_q^{l \times k_\mathrm{B}} \to \mathbb{F}_q$ which computes the j-th block of the matrix $\boldsymbol{C}_{i \cdot}$ from $\boldsymbol{G}_{\langle i \rangle_\mathrm{r} \cdot}^\top \boldsymbol{A} \in \mathbb{F}_{q^m}^{(n/n_\mathrm{A}) \times l}$ and $\boldsymbol{B}$. We assume that the error $e_{i'j'} \in \mathbb{F}_q$ occurs in computing the j'-th symbol of $f^1 \left(\boldsymbol{g}_{i' \cdot}^\top \boldsymbol{AB'v} \right) \in \mathbb{F}_q^{1 \times m}$. The master recieves a matrix $\boldsymbol{Y} := \boldsymbol{f}^n(\boldsymbol{GAB'v}) + \boldsymbol{E}$, where $\boldsymbol{E}$ is a matrix whose (i', j')-th entry is $e_{i'j'}$ for any $(i', j') \in [n] \times [m]$. This matrix is constructed from all output results of all workers.

$\langle$*Decoding Process*$\rangle$ The master gets $\psi(\boldsymbol{Y})$ from the matrix $\boldsymbol{Y}$ with a function $\psi : \mathbb{F}_q^{n \times m} \to \mathcal{C} \cup \{?\}$, where a symbol $? \notin \mathcal{C}$ represents a fact that the master cannot get an estimated matrix $\hat{\boldsymbol{AB}}$. Since the generator matrix is a generator matrix of a canonical systematic encoder, if $\psi(\boldsymbol{Y}) \in \mathcal{C}$, the master gets $\hat{\boldsymbol{AB}}$ from $\psi(\boldsymbol{Y})$.

Proposition 4.1: For any $(i,j) \in [n_\mathrm{A}] \times [n_\mathrm{B}]$ and for any $(i', j') \in \langle i \rangle_\mathrm{r} \times \langle j \rangle_\mathrm{c}$, the (i', j')-th entry of a matrix $\boldsymbol{f}^n(\boldsymbol{GABv})$ is $\sum_{l' \in [l]} \sum_{m' \in [k_\mathrm{B}]} \tilde{a}_{i'l'm'j'} b_{l'm'}$.
See Appendix A in [1].

Corollary 4.1: Any worker $(i,j) \in [n_{\mathrm{A}}] \times [n_{\mathrm{B}}]$ performs $(lk_{\mathrm{B}}-1)(mn/n_{\mathrm{A}}n_{\mathrm{B}})$ additions over $\mathbb{F}_q$ and $lk_{\mathrm{B}}(mn/n_{\mathrm{A}}n_{\mathrm{B}})$ multiplications over $\mathbb{F}_q$ in Computing Process.
See Appendix B in [1].

If $n_{\mathrm{A}} = n$ and $n_{\mathrm{B}} = 1$, then the GDCC is the DCC in Section III.

B. Example: GDCC Based on a Gabidulin Code

We propose a GDCC based on a Gabidulin code as an example of GDCCs. Before the definition, we assume some assumptions on q and m to construct the proposed scheme.

Definition 4.1 (Multiplication Table [8]): For any $i \in [m]$, we define $T_{i1}, \ldots, T_{im} \in \mathbb{F}_q$ as the elements uniquely determined by $v_1 v_i = \sum_{j\in[m]} T_{ij} v_j$. $\boldsymbol{T} := (T_{ij})_{(i,j)\in[m]^2} \in \mathbb{F}_q^{m\times m}$ is called *multiplication table*. We define $C(\boldsymbol{T}) \in \mathbb{Z}$ as the number of nonzero entries of $\boldsymbol{T}$. $C(\boldsymbol{T})$ is called the *complexity of the normal basis* $\{v_1, \ldots, v_m\}$.

Lemma 4.1 ([9]): An addition over $\mathbb{F}_{q^m}$ can be done with m additions over $\mathbb{F}_q$. Moreover, an multiplication over $\mathbb{F}_{q^m}$ can be done with $m(C(\boldsymbol{T})+1) - m^2 - 1$ additions over $\mathbb{F}_q$ and $m(C(\boldsymbol{T}) + m)$ multiplications over $\mathbb{F}_q$

Definition 4.2 (Optimal Normal Basis [10]): For any normal basis $\{v_1, \ldots, v_m\}$ and multicative table $\boldsymbol{T}$, $C(\boldsymbol{T}) \geq 2m-1$. The normal basis $\{v_1, \ldots, v_m\}$ is called an *optimal* normal basis if it achieves this lower bound.
Hereafter let a normal basis $\{v_1, \ldots, v_m\} \in \mathbb{F}_{q^m}$ over $\mathbb{F}_q$ be an *optimal* normal basis.

Lemma 4.2 ([10]): An optimal normal basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ exists if and only if q and m satisfies the following. $\log_2 q$ and m are prime with each other. $2m+1$ is a prime number. A multicative group $(\mathbb{Z}/(2m+1)\mathbb{Z})^\star = (\mathbb{Z}/(2m+1)\mathbb{Z}) \setminus \{0\}$ is generated from 2 and -1.

Assumption 4.1 (Assumption for the parameters): We assume the condition of Lemma 4.2.

Definition 4.3 (GDCCG): Let $\boldsymbol{G} \in \mathbb{F}_{q^m}^{n\times k_{\mathrm{A}}}$ is a canonical systematic generator matrix of a (n, k_{A}) Gabidulin code over $\mathbb{F}_{q^m}$. ψ is a bounded rank distance decoder of this code. This GDCC is called a *GDCCG*.

V. Performance Evaluations on the Proposed Schemes

We evaluate (a) the CTC of the GDCCG system and (b) the error-correcting capability of the GDCC system and the GDCCG system. We evaluate (b) of the GDCC and GDCCG and show that GDCCG corrects an error matrix that previous schemes cannot correct. Moreover, we compare (a) the CTC of the stand-alone scheme (SA) and that of GDCCG and give the parameter condition when GDCCG is superior to the SA.

A. Evaluations of the Computation Time Complexities

The CTC of the GDCCG system is the sum of the number of the four arithmetic operations in the Computing Process of each worker and the Decoding Process of the master over $\mathbb{F}_q$. To include operations over $\mathbb{F}_q^m$ in the evaluation of the CTC, we decompose operations over $\mathbb{F}_q^m$ to operations over $\mathbb{F}_q$ and enumerate them.

1) Assumptions on Computation Time Complexities: We assume some assumptions in this paper to evaluate the CTC of the GDCCG.

Assumption 5.1: We do not include CTC of the preprocess in the evaluation of (a) since $\boldsymbol{A}$ is input only once. Moreover, we do not include any communication time between workers and the master.
We use corresponding between $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$ and $\boldsymbol{f}^n(\boldsymbol{a})$ in the proposed scheme. We assume Assumption 5.2, also assumed in [8] [11] [12].

Assumption 5.2: We do not include the CTC of computing $\boldsymbol{f}^n(\boldsymbol{a})$ from $\boldsymbol{a} \in \mathbb{F}_{q^m}^n$ or that of computing $\left(\boldsymbol{f}^n\right)^{-1}(\boldsymbol{A})$ from $\boldsymbol{A} \in \mathbb{F}_q^{n\times m}$ in the evaluation of (a).

Proposition 5.1: For any q, m and $a \in \mathbb{F}_{q^m}$, if $f^1(a) = (a_1, \ldots, a_m) \in \mathbb{F}_q^{1\times m}$, then $f^1(a^{q^i}) = (a_{m-i+1}, \ldots, a_m, a_1, a_2, \ldots, a_{m-i})$.

Definition 5.1 (cyclic shift [8]): For any q, m and a, we define i-th*cyclic shift up* as $\boldsymbol{a}^{\uparrow i} := f(a^{q^i})$ and *cyclic shift down* $\boldsymbol{a}^{\downarrow i} := f(a^{q^i})$.
We assume Assumption 5.3, also assumed in [8] [11] [12].

Assumption 5.3: We do not include computation time complexities of cyclic shifts up or down in the evaluation.

From these assumptions, the below facts hold.

Proposition 5.2: Let q and m satisfy Assumption 4.1. An addition, which is also a subtraction, over $\mathbb{F}_{q^m}$ can be done with m additions over $\mathbb{F}_q$. A multiplication over $\mathbb{F}_{q^m}$ can be done with m^2-1 additions and m^2 multiplication over $\mathbb{F}_q$. An inversion over $\mathbb{F}_{q^m}$ can be done with $(m^2-1)(2\lfloor\log_2(m\log_2 q - 1)\rfloor + 2)$ additions and $m^2(2\lfloor\log_2(m\log_2 q - 1)\rfloor + 2)$ multiplications over $\mathbb{F}_q$.

Form Proposition 5.2, Corollary 5.1 holds. The value D is a little modified from the upper bound of the number of operations, derived in [11]. See Appendix C in [1].

Corollary 5.1: Let q and m satisfy the condition of Lemma 4.2. Let $t = \lfloor(n-k_{\mathrm{A}})/2\rfloor$ and $d = n - k_{\mathrm{A}} + 1$. Then D is an upper bound of the CTC of the decoding algorithm [11] of an (n, k_{A}) Gabidulin code over $\mathbb{F}_{q^m}$. D is

$$\begin{aligned}&2mnt+m^2+m-1+\frac{2}{3}(m(m-1)(2m-1)-t(t-1)(2t-1))\\&-(t-2)(m-t)-(t-1)(m^2-m-t^2+t)\\&+(dn+d^2-t^2+2dt+mt-n-4d-t-1)m\\&+(dn+3d^2+3t^2-4dt+mt-n-9d+9t+5)(2m^2-1)\\&+2t(2m^2-1)(2\lfloor\log_2(m\log_2 q-1)\rfloor).\end{aligned} \tag{1}$$

2) Evaluations and A Comparison to the Stand-Alone System: We evaluate (a) the CTC of the GDCCG system in case $\mathrm{rank}\boldsymbol{E} \leq t$. The CTC of the GDCC system is the sum of the number of four arithmetic operations over $\mathbb{F}_q$ of each worker and that of Decoding Process of the master.

Theorem 5.1 (evaluation of (a) of GDCCG): D is defined in Eq. (1). Under Assumption 4.1, the sum of CTC of Computation Process of each worker and that of Decoding Process of the master in the system of the GDCCG is at most

$$(2k_{\mathrm{B}}l-1)(mn/n_{\mathrm{A}}n_{\mathrm{B}}) + D. \tag{2}$$

Thus Eq.(2) is an upper bound of the CTC of this system.

Proof : We showed from Corollary 4.1 that CTC of Computation Process of each worker $(i,j) \in [n_{\mathrm{A}}] \times [n_{\mathrm{B}}]$ is at most $(2k_{\mathrm{B}}l-1)(mn/n_{\mathrm{A}}n_{\mathrm{B}})$. The master computes $\boldsymbol{f}^n(\boldsymbol{GAB'v})$ from $\boldsymbol{f}^n(\boldsymbol{GAB'v})+\boldsymbol{E}$ with the bounded rank-distance decoder in Decoding Process. We do not include time to compute $\boldsymbol{AB}$ from $\boldsymbol{f}^n(\boldsymbol{GAB'v})$ since $\boldsymbol{G}$ is a generator matrix of a canonical systematic encoder. Thus the CTC is at most D. $\square$

We compare the CTC of the stand-alone scheme.

Corollary 5.2 (Comparison (a) of the GDCCG with that of the SA system): Under Assumption 4.1, if $n, n_{\mathrm{A}}, n_{\mathrm{B}}$ satisfies the below, the value of Eq.(2) is less than the CTC of the SA system.

$$l > \frac{1}{2k_{\mathrm{B}}(k_{\mathrm{A}}-(mn/n_{\mathrm{A}}n_{\mathrm{B}}))}(k_{\mathrm{A}}k_{\mathrm{B}}-(mn/n_{\mathrm{A}}n_{\mathrm{B}})+D).$$

Example 5.1: Set $n=n_{\mathrm{A}}, m=n_{\mathrm{B}}$, $q=2$, $k_{\mathrm{A}}=100$, $k_{\mathrm{B}}=293$ and $l=100000$. The value of Eq.(2) is less than the CTC of the SA system if $n(>k_{\mathrm{A}})$ satisfies $101 \leq n \leq 172$. Table 4.1 of [10] shows that $(q,m)=(2,293)$ satisfy Assumption 4.1.

B. Evaluations of the Error-Correcting Capabilities

Theorem 5.2 (Evaluation of (b) of GDCC): The GDCC system computes correctly if the error matrix $\boldsymbol{E}$ is in $\mathcal{E}$.

Proof : GDCC outputs $\boldsymbol{f}^n(\boldsymbol{GAB'v}) \in \mathbb{F}_q^{n\times m}$ from $\boldsymbol{A}$ and $\boldsymbol{B}$ when no error occurs in the computation. If $\boldsymbol{E} \in \mathcal{E}$, then $\psi(\boldsymbol{f}^n(\boldsymbol{GAB'v})+\boldsymbol{E}) = \boldsymbol{f}^n(\boldsymbol{GAB'v})$ since $\boldsymbol{f}^n(\boldsymbol{GAB'v}) \in \mathcal{C}$ from Lemma 3.1. Thus this scheme corrects all error matrices in $\mathcal{E}$. $\square$

Theorem 5.3 (Evaluation of (b) of the GDCCG): The GDCCG system computes correctly if the error matrix $\boldsymbol{E}$ satisfies $\mathrm{rank}\boldsymbol{E} \leq t$.

Proof : An (n, k_{A}) Gabidulin code over $\mathbb{F}_{q^m}$ can correct an error matrix $\boldsymbol{E}$ if $\mathrm{rank}(\boldsymbol{E}) \leq t$. $\square$

This error-correcting capability is the same as that of [5].

Remark 5.1: Theorem 5.3 showed that (b) the error-correcting capability of the GDCCG is the same as that of [5]. However, (a) the CTC of the GDCCG is approximately $n/n_{\mathrm{A}}n_{\mathrm{B}}$ times the CTC of the DCCG. This is because, each worker $(i,j) \in [n_{\mathrm{A}}] \times [n_{\mathrm{B}}]$ computes the j-th block of $\boldsymbol{f}^{n/n_{\mathrm{A}}}(\boldsymbol{G}_{\langle i\rangle_{\mathrm{r}}}.\boldsymbol{AB'v}) \in \mathbb{F}_q^{(n/n_{\mathrm{A}})\times(m/n_{\mathrm{B}})}$ in the GDCCG, while the worker $i \in [n]$ computes all entries of $f^1(\boldsymbol{g}_{i'}.\boldsymbol{AB'v}) \in \mathbb{F}_q^m$ in the DCCG. Since the purpose of this paper is to compare the GDCCG with the SA, we do not compare the GDCCG with the DCCG in detail.

VI. Conclusion and Future Works

In this paper, we proposed and evaluated a new distributed coded computation scheme called GDCC and, as one example, GDCCG. First, we evaluated the GDCCG with (a) the computation time complexity and showed the parameter condition in which the GDCCG system is superior to the SA system. Next, we evaluated the GDCC and the GDCCG with (b) the error-correcting capability of the overall system and showed that the GDCCG system can correct $\boldsymbol{E}$ which satisfies $\mathrm{rank}(\boldsymbol{E}) \leq t$.

In future works, we would like to improve the proposed schemes. For example, if we use more efficient decoding algorithms such as [13], the GDCCG may be better concerning the CTC. For another example, the GDCCG uses more workers than the DCCG. Thus we would like to propose new DCCs considering not only (a) and (b) but also communication load and the number of workers.

References

[1] K. Kazama and T Matsushima. A group-type distributed coded computation scheme based on a gabidulin code. https://onl.bz/w2NxCX5, 2022.

[2] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran. Speeding up distributed machine learning using codes. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 1143–1147, 2016.

[3] K.H. Huang and J. A. Abraham. Algorithm-based fault tolerance for matrix operations. *IEEE Transactions on Computers*, Vol. C-33, No. 6, pp. 518–528, 1984.

[4] S. Dutta, V. Cadambe, and P. Grover. "short-dot": Computing large linear transforms distributedly using coded short dot products. *IEEE Transactions on Information Theory*, Vol. 65, No. 10, pp. 6171–6193, 2019.

[5] K. Kazama and T Matsushima. A coded computation method based on a gabidulin code and the evaluation of its error-correcting capability (in japanese). *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences A*, Vol. J104-A, No. 6, pp. 156–159, 2021.

[6] H. Niederreiter. *Finite Fields*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1984.

[7] E. M. Gabidulin. Theory of codes with maximum rank distance. *Problems of Information Transmission (English translation of Problemy Peredachi Informatsii)*, Vol. 21, No. 1, pp. 1–12, 1985.

[8] S. Gao, D. Panario, V. Shoup, et al. Algorithms for exponentiation in finite fields. *Journal of Symbolic Computation*, Vol. 29, No. 6, pp. 879–889, 2000.

[9] D. Silva and F. R. Kschischang. Fast encoding and decoding of gabidulin codes. In *2009 IEEE International Symposium on Information Theory (ISIT)*, pp. 2858–2862, 2009.

[10] S. Gao. *Normal bases over finite fields.* University of Waterloo Waterloo, 1993.

[11] M. Gadouleau and Zhiyuan Yan. Complexity of decoding gabidulin codes. In *2008 42nd Annual Conference on Information Sciences and Systems*, pp. 1081–1085, 2008.

[12] R. B. Venturelli and D. Silva. An evaluation of erasure decoding algorithms for gabidulin codes. In *2014 International Telecommunications Symposium (ITS)*, pp. 1–5, 2014.

[13] H. Bartz, T. Jerkovits, S. Puchinger, and J. Rosenkilde. Fast decoding of codes in the rank, subspace, and sum-rank metric. *IEEE Transactions on Information Theory*, Vol. 67, No. 8, pp. 5026–5050, 2021.