New Results on Communication-Computation Tradeoffs for Heterogeneous Coded Distributed Computing

Fan Xu, Shuo Shao and Meixia Tao

Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China
Emails: {xxiaof, shuoshao, mxtao}@sjtu.edu.cn

***Abstract*—This paper considers a heterogeneous MapReduce-based coded distributed computing framework. While the optimal communication-computation tradeoffs have been well studied for the homogeneous systems, they remain largely unknown for the heterogeneous systems. A new achievable scheme that takes into account the heterogeneity in both the Map phase and the Reduce phase is proposed. By using an improved cut-set bound and a peeling method, a new converse result is also obtained. The obtained upper and lower bounds of the optimal communication load coincide in some non-trivial cases, and are tighter than existing results. They also degenerate to the existing optimal results in homogeneous systems.**

I. Introduction

Communication overhead has been a major bottleneck in large-scale distributed computing. Coded distributed computing (CDC), introduced in [1], offers a promising solution to reduce the communication load in the MapReduce framework. In the MapReduce-based distributed computing, firstly each worker is assigned to compute some designed intermediate values (IVs) in the Map phase. Then in the Shuffle phase, workers exchange their IVs via a broadcast channel according to each worker's target function assignment. Lastly, in the Reduce phase, each worker can compute the final results of its assigned target functions. In [1], Li *et. al.* show that the communication load in the Shuffle phase can be substantially reduced by exploiting coded multicasting opportunities, at the expense of assigning redundant computation tasks at each worker in the Map phase.

Many existing studies on communication-computation tradeoffs in CDC focus on homogeneous systems [1]–[6], where each worker in the system has an even computation load in the Map phase and an even target function assignment load in the Reduce phase. These results cannot be directly generalized to heterogeneous CDC (HetCDC) systems where each work can be assigned with different computation load and different target function assignment load. Recently, several attempts have been made to study the communication-computation tradeoffs in the HetCDC [7]–[12], however the optimal tradeoff still remains largely unknown. It is difficult on both ends of this problem, either the achievable upper bound or the theoretical lower bound.

Achievable schemes of HetCDC have been studied in [7]–[12]. The authors in [7] obtain the optimal communication-computation tradeoff for the systems with $K = 3$ workers, but their achievable result is not optimal in general when $K > 3$. The authors in [8] also obtain an achievable upper bound, which is optimal with $K \leq 3$ workers, but is only optimal in two extreme cases when $K > 3$. Note that [7], [8] only consider heterogeneous computation load of workers, but still assume homogeneous target function assignment load. As for heterogeneous function assignment load, the authors in [9], [10] obtain achievable upper bounds for cascaded and non-cascaded computing systems, respectively, which are proved to be within a constant multiplicative gap to their obtained lower bounds. However, their achievable scheme is limited to specific function assignment load, and their considered systems are simplified, which consist of multiple worker groups where workers in the same group have the same computation load and function assignment load.

Establishing optimal lower bound for the optimal communication-computation tradeoff is also a non-trivial problem. Intuitively, the method in [1] can be extended to the heterogeneous systems by iteratively calculating the communication load needed in the data shuffling of each worker subset. However, this method is generally not tight for heterogeneous systems. Another possible method is using cut-set bound, as in [8], [12]. However, just like in many other network coding problems, cut-set bound is not optimal in general, since it cannot subtly discribe the mutual information between nodes within a same set.

In this work, we consider a general HetCDC system. Different to previous HetCDC systems, our model is in a more general sense since not only the number of input files computed in each worker is different, but the number of target functions assigned to each worker is also different. Both the achievable upper bound and the theoretical lower bound for the optimal tradeoff are obtained in this work.

Though our achievable scheme also relies on an optimization problem like [7], the fundamental strategy which the optimization problem is based on is more delicately designed in our paper. As a natural idea, the more IVs are encoded in one multicast message, the higher communication

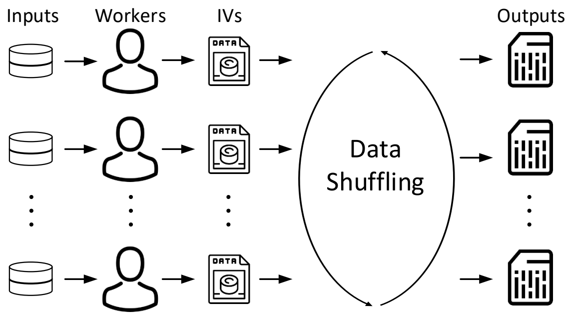

Fig. 1: CDC system.

efficiency this message will have. However in our strategy, we choose to slightly reduce the number of multicast messages with the highest efficiency, so that the number of uncoded messages, thus the messages with the lowest efficiency, can be largely decreased. Therefore, the overall communication load of our scheme is lower than those schemes which are only greedy on the number of messages with highest efficiency. Numerical results show that our scheme achieves a smaller communication load than existing works in some non-trivial cases. Details are shown in Section III and IV.

As for the lower bound, two methods are proposed after carefully partitioning the workers into two sets. The first method is improved from the conventional cut-set bound by jointly calculating the least number of IVs need to be shuffled within each set and between these two sets, while the second method calculates the least number of IVs need to be shuffled to each worker in one set and IVs need to be shuffled within the other set. Numerical results show that our achievable upper bound coincides with the lower bound in some non-trivial cases, and both bounds degenerate to the optimal tradeoff in the homogeneous case obtained by [1].

Notations: For $K \in \mathbb{Z}^+$, $[K]$ denotes the set $\{1, 2, \ldots, K\}$. $X_{\mathcal{S}}$ denotes the set $\{X_k : k \in \mathcal{S}\}$.

II. System Model

A. MapReduce Protocol

We consider a CDC system consisting of K distributed workers, which aims to compute Q target functions from N input files, with $K, Q, N \in \mathbb{Z}^+$. The input files are denoted by $\{f_1, \ldots, f_N\}$, each with F bits, while the target functions are denoted by $\{\phi_1, \ldots, \phi_Q\}$, each mapping all the input files into the output value Φ_q with length B bits for $q \in [Q]$.

Each worker k, for $k \in [K]$, can store $\lfloor m_k N \rfloor$ input files whose indices are denoted by $\mathcal{M}_k$, where $m_k \in [0, 1] \cap \mathbb{Q}$ is called its local computation load and $\sum_{k\in[K]} m_k \geq 1$. It is also assigned to compute $\lfloor w_k Q \rfloor$ target functions whose indices are denoted by $\mathcal{W}_k$, where $w_k \in [0, 1] \cap \mathbb{Q}$ is called its local function assignment load. We assume that $\mathcal{W}_j \cap \mathcal{W}_k = \emptyset$ for any $j, k \in [K]$ and $j \neq k$, which implies that $\sum_{k\in[K]} w_k = 1$. The overall computation load and the overall function assignment load of workers is defined as $\mathbf{m} \triangleq [m_1, \ldots, m_K]$ and $\mathbf{w} \triangleq [w_1, \ldots, w_K]$, respectively.

The CDC system follows the MapReduce framework as in Fig. 1, where the computation of each target function ϕ_q can be decomposed as

$$\phi_q(f_1, \ldots, f_N) = h_q(g_{q,1}(f_1), \ldots, g_{q,N}(f_N)). \tag{1}$$

Here, $\mathbf{g}_n \triangleq (g_{1,n}, \ldots, g_{Q,n})$ is the Map function of input file f_n that maps f_n into Q IVs $\{v_{q,n} \triangleq g_{q,n}(f_n) : q \in [Q]\}$, for $n \in [N]$. Each IV $v_{q,n}$ has T bits, and corresponds to target function ϕ_q, for $n \in [N], q \in [Q]$. h_q is the Reduce function of target function ϕ_q that maps its IVs computed from all input files into the output value $\Phi_q = h_q(v_{q,1}, \ldots, v_{q,N})$, for $q \in [Q]$. Following this decomposition, the considered CDC system consists of three phases: *Map*, *Shuffle* and *Reduce*.

Map phase: Each worker k, for $k \in [K]$, computes the Map functions of its stored input files to obtain the IVs $\{v_{q,n} : q \in [Q], n \in \mathcal{M}_k\}$.

Shuffle phase: Each worker k, for $k \in [K]$, uses its local IVs to generate a ℓ_k-bit message X_k, and broadcasts it to the other workers. After receiving messages from other workers, each worker k, for $k \in [K]$, uses its received messages and its local IVs to obtain the needed IVs $\{v_{q,n} : q \in \mathcal{W}_k, n \in [N]\}$. The overall communication load L in the Shuffle phase is defined as $L \triangleq \frac{\sum_{k=1}^{K} \ell_k}{QNT}$ which is normalized by the total number of bits of all the QN IVs computed in the Map phase.

Reduce phase: Each worker k, for $k \in [K]$, uses its obtained IVs $\{v_{q,n} : q \in \mathcal{W}_k, n \in [N]\}$ to compute the Reduce functions of its assigned target functions $\{\phi_q : q \in \mathcal{W}_k\}$ and obtain the output values $\{\Phi_q : q \in \mathcal{W}_k\}$.

B. Information Theoretic Definition

In the considered system, our work focuses on the design of input file allocation in the Map phase and the encoding and decoding functions of workers in the Shuffle phase so as to reduce the communication load, similar to [1]–[12].

In the following, we give the formal information theoretic model for the achievable communication load. We claim an overall communication load L is achievable in a K-worker CDC system with computation load $\mathbf{m}$ and function assignment load $\mathbf{w}$, if we can find following functions:

1) A file allocation function
$$c : [K] \to [2]^N$$
that associates each worker k to a set of indices
$$\mathcal{M}_k = c(k) \subset [N],$$
where $|\mathcal{M}_k| \leq \lfloor N m_k \rfloor$, for each $k \in [K]$.
2) K Shuffle phase encoding functions
$$e_k : [2^T]^{\lfloor m_k N \rfloor Q} \to [2^{\ell_k}],$$
for $k \in [K]$. Each e_k maps the IVs in worker k to a ℓ_k-bit message
$$X_k = e_k(\{v_{q,n} : q \in [Q], n \in \mathcal{M}_k\}).$$
for each $k \in [K]$.
3) K Shuffle phase decoding functions
$$d_k : [2^{\ell_1}] \times \cdots \times [2^{\ell_K}] \times [2^T]^{\lfloor m_k N \rfloor Q} \to [2^T]^{N \lfloor w_k Q \rfloor},$$

for $k \in [K]$. Each d_k maps all the broadcast messages and the IVs in worker k to decode its demanded IVs

$$\{v_{q,n} : q \in \mathcal{W}_k, n \in [N]\} \\ = d_k(X_1, \ldots, X_k, \{v_{q,n} : q \in [Q], n \in \mathcal{M}_k\}).$$

for each $k \in [K]$.

The optimal communication load in the Shuffle phase is defined by

$$L^*(\mathbf{m}, \mathbf{w}) \triangleq \inf\{L(\mathbf{m}, \mathbf{w}) : (L, \mathbf{m}, \mathbf{w}) \text{ is achievable.}\}.$$

In this work, our goal is to characterize the optimal communication load for any given computation load $\mathbf{m}$ and function assignment load $\mathbf{w}$.

III. Main Results

In this section, we first present our achievable upper bound and the theoretical lower bound for the optimal communication load, and then compare our results with existing works.

A. Upper Bound

We start by illustrating the main ideas of our achievable scheme in a general CDC system. In the Map phase, we exclusively allocate $a_{\mathcal{S}}N$ input files to worker set $\mathcal{S}$. In the Shuffle phase, we consider the data shuffling of all possible worker sets. Specifically, each worker in an arbitrary worker set $\mathcal{S}$ multicasts a coded message to the rest workers in this set. Due to heterogeneity, some workers may need more IVs than others. These extra IVs will be multicasted in the data shuffling of smaller worker sets in this set $\mathcal{S}$.

In the data shuffling of an arbitrary worker set $\mathcal{S}$, we use $L_{k,\mathcal{S}}$ to denote the quantity of multicast messages sent by worker $k \in \mathcal{S}$ to workers $\mathcal{S} \setminus \{k\}$, and use $d_{k,\mathcal{S}}^{\mathcal{S}\setminus\{i\}}$ to denote the quantity of IVs needed by worker $k \in \mathcal{S}$ which will be multicasted in the data shuffling of a smaller worker set $\mathcal{S} \setminus \{i\}$. By jointly optimizing the file allocation strategy $\{a_{\mathcal{S}}\}$ and the data shuffling strategy $\{L_{k,\mathcal{S}}, d_{k,\mathcal{S}}^{\mathcal{S}\setminus\{i\}}\}$, we can obtain our achievable communication load, which is given in the following theorem.

Theorem 1. *(Achievable load) For a heterogeneous MapReduce computing system with K workers, computation load $\mathbf{m} = [m_1, \ldots, m_K]$, and function assignment load $\mathbf{w} = [w_1, \ldots, w_K]$, the optimal communication load $L^*(\mathbf{m}, \mathbf{w})$ is upper bounded by $L^*(\mathbf{m}, \mathbf{w}) \le L_{UB}(\mathbf{m}, \mathbf{w})$ where $L_{UB}(\mathbf{m}, \mathbf{w})$ is defined by the following linear optimization problem:*

$$\mathcal{P}_{UB} : L_{UB}(\mathbf{m}, \mathbf{w}) \triangleq$$

$$\min_{\{a_{\mathcal{S}}, L_{k,\mathcal{S}}, d_{k,\mathcal{S}}^{\mathcal{S}\setminus\{i\}}\}} \sum_{\substack{\mathcal{S},k:\mathcal{S}\subseteq[K],\\ |\mathcal{S}|\ge 2, k\in\mathcal{S}}} L_{k,\mathcal{S}} \tag{2a}$$

$$\textit{s.t.} \sum_{\substack{\mathcal{S}:\mathcal{S}\subseteq[K],\\ |\mathcal{S}|\ge 1}} a_{\mathcal{S}} = 1, \tag{2b}$$

$$\sum_{\substack{\mathcal{S}:\mathcal{S}\subseteq[K],\\ |\mathcal{S}|\ge 1, \mathcal{S}\ni k}} a_{\mathcal{S}} \le m_k, \forall k \in [K] \tag{2c}$$

$$w_k a_{\mathcal{S}\setminus\{k\}} + \sum_{i\in[K]\setminus\mathcal{S}} d_{k,\mathcal{S}\cup\{i\}}^{\mathcal{S}} \\ = \sum_{i\in\mathcal{S}\setminus\{k\}} L_{i,\mathcal{S}} + \sum_{i\in\mathcal{S}\setminus\{k\}} d_{k,\mathcal{S}}^{\mathcal{S}\setminus\{i\}}, \\ \forall \mathcal{S} \subseteq [K], |\mathcal{S}| \ge 2, k \in \mathcal{S} \tag{2d}$$

$$a_{\mathcal{S}} \ge 0, \forall \mathcal{S} \subseteq [K], |\mathcal{S}| \ge 1 \tag{2e}$$

$$L_{k,\mathcal{S}} \ge 0, \forall \mathcal{S} \subseteq [K], |\mathcal{S}| \ge 2, k \in \mathcal{S} \tag{2f}$$

$$d_{k,\mathcal{S}}^{\mathcal{S}\setminus\{i\}} \ge 0, \forall \mathcal{S} \subseteq [K], |\mathcal{S}| \ge 3, k \in \mathcal{S} \tag{2g}$$

Our scheme creates more flexibility on coded multicasting than [7] that we allow IVs with different number of copies to be encoded together. By introducing parameters $\{d_{k,\mathcal{S}}^{\mathcal{S}\setminus\{i\}}\}$ in (2d), our scheme can combine those unmatched IVs together to avoid unicasting them.

B. Lower Bound

Theorem 2. *(Lower bound) For a heterogeneous MapReduce computing system with K workers, computation load $\mathbf{m} = [m_1, \ldots, m_K]$, and function assignment load $\mathbf{w} = [w_1, \ldots, w_K]$, the optimal communication load $L^*(\mathbf{m}, \mathbf{w})$ is lower bounded by $L^*(\mathbf{m}, \mathbf{w}) \ge L_{LB}(\mathbf{m}, \mathbf{w})$, where $L_{LB}(\mathbf{m}, \mathbf{w})$ is defined by the following optimization problem:*

$$\mathcal{P}_{LB} : L_{LB}(\mathbf{m}, \mathbf{w}) \triangleq$$

$$\min_{\{a_{\mathcal{S}}\}} \max_{\substack{\mathcal{S},p:\\ \mathcal{S}\subseteq[K], p\in\mathcal{P}_{\mathcal{S}}}} \{L_{\mathcal{S},p}, L_{\mathcal{S}}\} \tag{3a}$$

$$\textit{s.t.} \sum_{\substack{\mathcal{S}:\mathcal{S}\subseteq[K],\\ |\mathcal{S}|\ge 1}} a_{\mathcal{S}} = 1, \tag{3b}$$

$$\sum_{\substack{\mathcal{S}:\mathcal{S}\subseteq[K],\\ |\mathcal{S}|\ge 1, \mathcal{S}\ni k}} a_{\mathcal{S}} \le m_k, \forall k \in [K] \tag{3c}$$

$$a_{\mathcal{S}} \ge 0, \forall \mathcal{S} \subseteq [K], |\mathcal{S}| \ge 1 \tag{3d}$$

where

$$L_{\mathcal{S},p} \triangleq \sum_{k\in|\mathcal{S}|} \sum_{\mathcal{T}\subseteq[K]\setminus\mathcal{S}_p([k])} w_{\mathcal{S}_p(k)} a_{\mathcal{T}} \\ + \sum_{k\in\mathcal{S}^c} \sum_{\mathcal{T}\subseteq\mathcal{S}^c\setminus\{k\}} w_k a_{\mathcal{T}} \frac{1}{|\mathcal{T}|}, \tag{4}$$

$$L_{\mathcal{S}} \triangleq \sum_{k\in\mathcal{S}} \sum_{\mathcal{T}\subseteq\mathcal{S}^c} w_k a_{\mathcal{T}} + \sum_{k\in\mathcal{S}} \sum_{\mathcal{T}\subseteq\mathcal{S}\setminus\{k\}} w_k a_{\mathcal{T}} \frac{1}{|\mathcal{T}|} \\ + \sum_{k\in\mathcal{S}^c} \sum_{\mathcal{T}\subseteq\mathcal{S}} w_k a_{\mathcal{T}} + \sum_{k\in\mathcal{S}^c} \sum_{\mathcal{T}\subseteq\mathcal{S}^c\setminus\{k\}} w_k a_{\mathcal{T}} \frac{1}{|\mathcal{T}|}, \tag{5}$$

and $\mathcal{S}^c \triangleq [K] \setminus \mathcal{S}$, $\mathcal{P}_{\mathcal{S}}$ denotes all permutations of workers in $\mathcal{S}$, $\mathcal{S}_p(k)$ and $\mathcal{S}_p([k])$ denote the k-th worker and the first k workers in permutation $p \in \mathcal{P}_{\mathcal{S}}$, respectively.

Sketch of the proof: We only illustrate the main ideas of the proof, while the complete proof is given in the longer version [13]. We first derive the lower bound for any given input file allocation and then optimize the file allocation to obtain the final lower bound. For any given file allocation

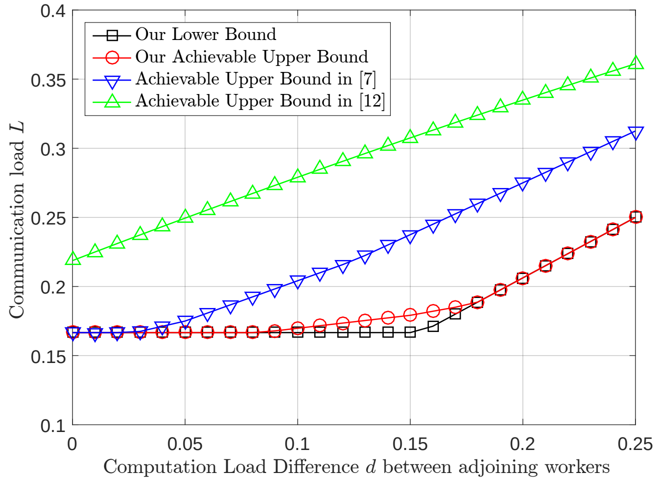

Fig. 2: Communication load in a 4-worker system as a function of the computation load difference d between adjoining workers.

$\{a_{\mathcal{S}}\}$, we first partition the workers into two sets $\mathcal{S}$ and $\mathcal{S}^c$, and then adopt two methods to obtain the lower bounds. The first method is improved from the conventional cut-set bound, which calculates the least number of IVs need to be shuffled within sets $\mathcal{S}$ and $\mathcal{S}^c$ and between these two sets. The second method is called the peeling method. After permuting workers in $\mathcal{S}$ according to permutation $p \in \mathcal{P}_{\mathcal{S}}$, this method calculates the number of IVs needed by worker $\mathcal{S}_p(k)$ and not computed by workers $\mathcal{S}_p([k])$, for each $k \in |\mathcal{S}|$, and the least number of IVs need to be shuffled within set $\mathcal{S}^c$. By considering all possible permutation $p \in \mathcal{P}_{\mathcal{S}}$ and all possible partition $\mathcal{S}$ and $\mathcal{S}^c$, we can obtain the lower bound of file allocation $\{a_{\mathcal{S}}\}$.

C. Comparison with Existing Results

Remark 1. When each worker k has the same computation load $m_k = m$ and the same function assignment load $w_k = \frac{1}{K}$, both the upper and lower bounds degenerate to the optimal communication load in [1, Theorem 1], i.e., $L_{UB} = L_{LB} = L^*$, where L^* is given by the lower convex envelope of points $L^* = \frac{1-m}{Km}$ for $m \in \{\frac{1}{K}, \frac{2}{K}, \dots, 1\}$.

Remark 2. In a general heterogeneous system, our upper bound coincides the lower bound in some non-trivial cases, and is tighter than those achieved by [7], [12].

To illustrate Remark 2, Fig. 2 plots the communication load as a function of the computation load difference d between adjoining workers in a 4-worker system, where the computation load vector $\mathbf{m}$ is assumed to be an arithmetic progression with $\sum_k m_k = 2.5$, and the function assignment load is given by $w_k = \frac{1}{4}$ for $k \in [4]$. Our achievable upper bound coincides with the lower bound when $d \leq 0.08$ or $d \geq 0.18$, which implies that both bounds sufficiently take coded multicasting opportunities into consideration in the Shuffle phase. Compared with existing works, our upper bound is tighter than those achieved in [7] and [12]. Specifically, when $d < 0.03$, both the achievable results in [7] and our scheme coincide with the lower bound. However, when d increases, which implies that the heterogeneity grows, our achievable result outperforms that in [7]. We will compare our scheme with the schemes in [7], [12] in detail in Section IV-B.

IV. Achievable Scheme (Proof of Theorem 1)

In this section, we prove Theorem 1 for the general heterogeneous system, and then illustrate our proposed scheme by using a 4-worker system as an example.

A. General Scheme

Let $\{a^*_{\mathcal{S}}, L^*_{k,\mathcal{S}}, d^{\mathcal{S}\setminus\{i\}*}_{k,\mathcal{S}}\}$ denote the optimal solution of Problem $\mathcal{P}_{UB}$.

1) Map phase: We exclusively assign a fraction $a^*_{\mathcal{S}}$ of the total input files to worker set $\mathcal{S}$[1]. Note that non-negative parameters $\{a^*_{\mathcal{S}} : \mathcal{S} \subseteq [K], |\mathcal{S}| \geq 1\}$ satisfy (2b) and (2c), which are exactly the constraint of the overall number of input files and the constraint of the local computation load of each worker, respectively.

2) Shuffle phase: We illustrate the data shuffling of an arbitrary worker set $\mathcal{S}$ with $|\mathcal{S}| \geq 2$. The number of IVs needed by worker $k \in \mathcal{S}$, and exclusively computed at the rest workers $\mathcal{S}\setminus\{k\}$ is given by $w_k a^*_{\mathcal{S}\setminus\{k\}} QN$. In addition, as shown in Section III-A, there are some IVs needed by worker k which are not sent in the data shuffling of larger worker sets and will be multicasted in the data shuffling of set $\mathcal{S}$. Specifically, there are $d^{\mathcal{S}*}_{k,\mathcal{S}\cup\{i\}} QN$ IVs needed by worker k, which are allocated to this set from set $\mathcal{S}\cup\{i\}$, for each $i \in [K]\setminus\mathcal{S}$. Note that these IVs are also available at workers $\mathcal{S}\setminus\{k\}$.

Thus, each worker $k \in \mathcal{S}$ totally needs $(w_k a^*_{\mathcal{S}\setminus\{k\}} + \sum_{i\in[K]\setminus\mathcal{S}} d^{\mathcal{S}}_{k,\mathcal{S}\cup\{i\}}*)QN$ IVs, which are available at workers $\mathcal{S}\setminus\{k\}$[2]. Among these IVs, we allocate $d^{\mathcal{S}\setminus\{i\}*}_{k,\mathcal{S}} QN$ IVs to the data shuffling of worker set $\mathcal{S}\setminus\{i\}$ for $i \in \mathcal{S}\setminus\{k\}$[3]. Thus, each worker $k \in \mathcal{S}$ actually needs to receive $l_{k,\mathcal{S}} QN$ IVs in the shuffling of set $\mathcal{S}$, with

$$l_{k,\mathcal{S}} \triangleq w_k a_{\mathcal{S}\setminus\{k\}} + \sum_{i\in[K]\setminus\mathcal{S}} d^{\mathcal{S}*}_{k,\mathcal{S}\cup\{i\}} - \sum_{i\in\mathcal{S}\setminus\{k\}} d^{\mathcal{S}\setminus\{i\}*}_{k,\mathcal{S}}. \quad (6)$$

Let $\mathcal{I}_{k,\mathcal{S}}$ denote these IVs needed by each worker $k \in \mathcal{S}$. Then, each worker k, for $k \in \mathcal{S}$, collects $L^*_{k,\mathcal{S}} QN$ distinct IVs from $\mathcal{I}_{i,\mathcal{S}}$ for each worker $i \in \mathcal{S}\setminus\{k\}$ in this set, which is denoted by $\mathcal{I}^k_{i,\mathcal{S}}$, and multicasts a coded message $X^k_{\mathcal{S}} \triangleq \bigoplus_{i\in\mathcal{S}\setminus\{k\}} \mathcal{I}^k_{i,\mathcal{S}}$ to the rest workers in this set. The communication load in this worker set is thus given by

$$L_{\mathcal{S}} = \sum_{k\in\mathcal{S}} L^*_{k,\mathcal{S}}.$$

Given (2d), we have $\sum_{i\in\mathcal{S}\setminus\{k\}} L^*_{i,\mathcal{S}} = l_{k,\mathcal{S}}$ for each worker $k \in \mathcal{S}$, which implies that worker k can successfully obtain its needed IVs $\mathcal{I}_{k,\mathcal{S}}$ from messages $\{X^i_{\mathcal{S}} : i \in \mathcal{S}\setminus\{k\}\}$ in the shuffling of set $\mathcal{S}$ since it has IVs $\{\mathcal{I}_{i,\mathcal{S}} : i \in \mathcal{S}\setminus\{k\}\}$.

[1] We assume that input file number N is large enough so that our achievable scheme avoids rounding down.

[2] If $\mathcal{S} = [K]$, $d^{\mathcal{S}}_{k,\mathcal{S}\cup\{i\}}$ does not exist.

[3] If $|\mathcal{S}| = 2$, we cannot allocate IVs to smaller groups. Hence, $d^{\mathcal{S}\setminus\{i\}}_{k,\mathcal{S}} = 0$.

3) Total Communication Load: By considering all possible worker sets, the total communication load is given by

$$L_{total} = \sum_{\substack{\mathcal{S},k:\mathcal{S}\subseteq[K]\\ |\mathcal{S}|\geq 2, k\in\mathcal{S}}} L^*_{k,\mathcal{S}} = L_{UB}(\mathbf{m},\mathbf{w}). \tag{7}$$

Thus, Theorem 1 is proved.

Remark 3. Problem $\mathcal{P}_{UB}$ is different from that in [7]. In [7], the authors only optimize the file allocation, i.e., $\{a_{\mathcal{S}}\}$, and the quantity of multicast messages in the data shuffling of each worker set, similar to $\{L_{k,\mathcal{S}}\}$ in Problem $\mathcal{P}_{UB}$, but do not exploit coded multicasting opportunities for the extra IVs needed by workers who need more IVs than others. Instead of directly unicasting these extra IVs in [7], our scheme multicasts them in the data shuffling of smaller worker sets, and Problem $\mathcal{P}_{UB}$ additionally optimizes the quantity of these extra IVs multicasted with IVs with less number of copies, i.e., $\{d^{\mathcal{S}\backslash\{i\}}_{k,\mathcal{S}}\}$. Thus, our scheme has more flexibility on coded multicasting, and outperforms that in [7], as shown in Fig. 2.

B. An example

Now we use a 4-worker system as an example to illustrate our data shuffling strategy, and compare our scheme with existing works. We consider a 4-worker system with $\mathbf{m} = [\frac{1}{2}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}]$ and $w_k = \frac{1}{4}$ for each $k \in [4]$.

Fig. 3 shows our data shuffling strategy by solving Problem $\mathcal{P}_{UB}$ and the strategy in [7]. In Fig. 3, $\mathcal{V}^k_{\mathcal{S}}$ denotes the IVs needed by worker k and exclusively computed by set $\mathcal{S}$. In our scheme, after the multicasting of messages $\{\bigoplus_{i\in[4]\backslash\{k\}} \mathcal{V}^i_{[4]\backslash\{i\}} : k \in \{2,3,4\}\}$, each worker k, for $k \in \{2,3,4\}$, obtains its needed $0.028QN$ IVs of $\mathcal{V}^k_{[4]\backslash\{k\}}$, but worker 1 still needs $0.063QN$ IVs of $\mathcal{V}^1_{\{2,3,4\}}$. These IVs will be multicasted with IVs with less number of copies, e.g, $0.014QN$ IVs of $\mathcal{V}^1_{\{2,3,4\}}$ will be multicasted with $\mathcal{V}^3_{\{1,2\}}$ by worker 2 to workers 1 and 3. As a result, unicasting is avoided in our scheme. On the other hand, the scheme in [7] only exploits coded multicasting opportunities for IVs with the same number of copies. Therefore, worker 4 in [7] still needs to unicast $0.047QN$ IVs of $\mathcal{V}^1_{\{2,3,4\}}$ to worker 1.

Table I shows the communication load used for unicasting and multicasting in [7], [12] and our scheme. The scheme in [12] is generalized from decentralized cache placement and content delivery [14]. Therefore, its total communication load is the largest since it cannot efficiently exploit coded multicasting opportunities in the Shuffle phase. Compared with the scheme in [7], though our scheme multicasts slightly less messages to three workers, it multicasts more messages to two workers. As a result, our scheme avoids unicasting, and it achieves the smallest total communication load among the three considered schemes. This implies that, by slightly reducing the quantity of messages multicasted to many workers, our scheme can alleviate the cost of unicasting in the Shuffle phase, so as to the reduce the total communication load.

V. Conclusion

In this paper, we considered a HetCDC system with computation load $\mathbf{m}$ and function assignment load $\mathbf{w}$. We obtain both the achievable upper bound and the theoretical lower bound of the optimal communication load. If the system degenerates to a homogeneous case, our result recovers the result in [1]. When $K = 4$ and $\mathbf{m} = [\frac{5}{8} - \frac{3}{2}d, \frac{5}{8} - \frac{1}{2}d, \frac{5}{8} + \frac{1}{2}d, \frac{5}{8} + \frac{3}{2}d]$, our result is strictly tighter than [7] for $d \in [0.03, 0.25]$, and optimal for $d \in [0, 0.08] \cup [0.18, 0.25]$.

Send \ Need (Worker 1, Worker 2, Worker 3, Worker 4)		Amount of IVs shuffled
Worker 2	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^3_{\{1,2,4\}} \oplus \mathcal{V}^4_{\{1,2,3\}}$	0.014
	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^3_{\{1,2\}}$	0.014
	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^4_{\{1,2\}}$	0.014
	$\mathcal{V}^3_{\{2,4\}} \oplus \mathcal{V}^4_{\{2,3\}}$	0.003
Worker 3	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^2_{\{1,3,4\}} \oplus \mathcal{V}^4_{\{1,2,3\}}$	0.014
	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^2_{\{1,3\}}$	0.007
	$\mathcal{V}^1_{\{2,3\}} \oplus \mathcal{V}^2_{\{1,3\}}$	0.007
	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^4_{\{1,3\}}$	0.014
	$\mathcal{V}^2_{\{3,4\}} \oplus \mathcal{V}^4_{\{2,3\}}$	0.003
Worker 4	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^2_{\{1,3,4\}} \oplus \mathcal{V}^3_{\{1,2,4\}}$	0.014
	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^2_{\{1,4\}}$	0.007
	$\mathcal{V}^1_{\{2,4\}} \oplus \mathcal{V}^2_{\{1,4\}}$	0.007
	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^3_{\{1,4\}}$	0.007
	$\mathcal{V}^1_{\{3,4\}} \oplus \mathcal{V}^3_{\{1,4\}}$	0.007
	$\mathcal{V}^2_{\{3,4\}} \oplus \mathcal{V}^3_{\{2,4\}}$	0.003

(a)

Send \ Need (Worker 1, Worker 2, Worker 3, Worker 4)		Amount of IVs shuffled
Worker 1	$\mathcal{V}^2_{\{1,3\}} \oplus \mathcal{V}^3_{\{1,2\}}$	0.005
	$\mathcal{V}^2_{\{1,4\}} \oplus \mathcal{V}^4_{\{1,2\}}$	0.005
	$\mathcal{V}^3_{\{1,4\}} \oplus \mathcal{V}^4_{\{1,3\}}$	0.005
Worker 2	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^3_{\{1,2,4\}} \oplus \mathcal{V}^4_{\{1,2,3\}}$	0.016
	$\mathcal{V}^1_{\{2,3\}} \oplus \mathcal{V}^3_{\{1,2\}}$	0.005
	$\mathcal{V}^1_{\{2,4\}} \oplus \mathcal{V}^4_{\{1,2\}}$	0.005
	$\mathcal{V}^3_{\{2,4\}} \oplus \mathcal{V}^4_{\{2,3\}}$	0.005
Worker 3	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^2_{\{1,3,4\}} \oplus \mathcal{V}^4_{\{1,2,3\}}$	0.016
	$\mathcal{V}^1_{\{2,3\}} \oplus \mathcal{V}^2_{\{1,3\}}$	0.005
	$\mathcal{V}^1_{\{3,4\}} \oplus \mathcal{V}^4_{\{1,3\}}$	0.005
	$\mathcal{V}^2_{\{3,4\}} \oplus \mathcal{V}^4_{\{2,3\}}$	0.005
Worker 4	$\mathcal{V}^1_{\{2,3,4\}} \oplus \mathcal{V}^2_{\{1,3,4\}} \oplus \mathcal{V}^3_{\{1,2,4\}}$	0.016
	$\mathcal{V}^1_{\{2,4\}} \oplus \mathcal{V}^2_{\{1,4\}}$	0.005
	$\mathcal{V}^1_{\{3,4\}} \oplus \mathcal{V}^3_{\{1,4\}}$	0.005
	$\mathcal{V}^2_{\{3,4\}} \oplus \mathcal{V}^3_{\{2,4\}}$	0.005
	$\mathcal{V}^1_{\{2,3,4\}}$	0.047

(b)

Fig. 3: Data shuffling in (a) our scheme, (b) the scheme in [7]

TABLE I: Communication load of Unicasting and Multicasting

Scheme \ Shuffling	Unicasting	Multicasting to 2 workers	Multicasting to 3 workers	Total
Our Scheme	0	0.094	0.042	0.136
The scheme in [7]	0.047	0.063	0.047	0.157
The scheme in [12]	0.132	0.056	0.023	0.211

References

[1] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, Jan 2018.

[2] N. Woolsey, R. Chen, and M. Ji, "A new combinatorial design of coded distributed computing," in *IEEE ISIT*, June 2018.

[3] K. Konstantinidis and A. Ramamoorthy, "Leveraging coding techniques for speeding up distributed computing," in *IEEE GLOBECOM*, Dec 2018.

[4] Q. Yan, S. Yang, and M. Wigger, "Storage, computation, and communication: A fundamental tradeoff in distributed computing," in *IEEE ITW*, Nov 2018.

[5] M. Zhao, W. Wang, Y. Wang, and Z. Zhang, "Load scheduling for distributed edge computing: A communication-computation tradeoff," *Peer-to-Peer Networking and Applications*, Oct 2018. [Online]. Available: https://doi.org/10.1007/s12083-018-0695-4

[6] F. Li, J. Chen, and Z. Wang, "Wireless MapReduce distributed computing," in *IEEE ISIT*, June 2018.

[7] M. Kiamari, C. Wang, and A. S. Avestimehr, "On heterogeneous coded distributed computing," in *IEEE GLOBECOM*, Dec 2017, pp. 1–7.

[8] N. Shakya, F. Li, and J. Chen, "On distributed computing with heterogeneous communication constraints," in *52nd Asilomar Conf. Signals, Syst. Comput.*, Oct 2018, pp. 1795–1799.

[9] N. Woolsey, R. Chen, and M. Ji, "Cascaded coded distributed computing on heterogeneous networks," in *IEEE ISIT*, July 2019.

[10] ——, "Coded distributed computing with heterogeneous function assignments," 2019. [Online]. Available: http://arxiv.org/abs/1902.10738

[11] M. Kiamari, C. Wang, and A. S. Avestimehr, "Coding for edge-facilitated wireless distributed computing with heterogeneous users," in *51st Asilomar Conf. Signals, Syst. Comput.*, Oct 2017, pp. 536–540.

[12] F. Xu and M. Tao, "Heterogeneous coded distribtued computing: joint design of file allocation and function assignment," in *IEEE GLOBECOM*, Dec 2019.

[13] F. Xu, S. Shao, and M. Tao, "Heterogeneous distributed computation," 2020. [Online]. Available: http://iwct.sjtu.edu.cn/Personal/mxtao/paper/isit20-xu.pdf

[14] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Networking*, vol. 23, no. 4, pp. 1029–1040, Aug 2015.