Optimal Heterogeneous Distributed Storage Regenerating Code at Minimum Remote-Repair Bandwidth Regenerating Point

Jian Xu, Yewen Cao, Deqiang Wang, Changlei Wu, and Guang Yang

Recently, a product-matrix (PM) framework was proposed to construct optimal regenerating codes for homogeneous distributed storage systems (DSSs). In this paper, we propose an extended PM (EPM) framework for coding of heterogeneous DSSs having different repair bandwidths but identical storage capacities. Based on the EPM framework, an explicit construction of minimum remote-repair bandwidth regenerating (MRBR) codes is presented for a specific heterogeneous DSS, where two geographically different datacenters with associated storage nodes are deployed. The data reconstruction and regeneration properties of the MRBR code are proved strictly. For the purpose of demonstration, an example implementation of MRBR code is provided. The presented MRBR code is the first optimal strict-regenerating code for heterogeneous DSSs. In addition, our proposed EPM framework can be applied to homogeneous systems also.

Keywords: Heterogeneous distributed storage, remote-repair bandwidth, regenerating code, extended product-matrix, strict-regenerating code.

Manuscript received May 15, 2015; revised Dec. 14, 2015; accepted Dec. 28, 2015.

This work was supported by the Natural Science Foundation (NSF) of China (61471222) and the NSF of Shandong Province (ZR2015FM003).

Jian Xu (corresponding author, jianxusdu@126.com), Yewen Cao (ycao@sdu.edu.cn), Deqiang Wang (wdq_sdu@sdu.edu.cn), Changlei Wu (wu_chlei@163.com), and Guang Yang (yangguang_1030@163.com) are with the School of Information Science and Engineering, Shandong University, Jinan, China.

I. Introduction

Cloud storage services can relieve the burden of storage resources and avoid huge expenditure on the construction of infrastructural facilities [1]. In a cloud storage system (CSS), a data file (*the message*) is encoded and stored in n distributed storage nodes. The storage capacity of each node is α. In the process of *data reconstruction*, a data collector (DC) can reconstruct the original data file provided he or she has access to k of the n distributed storage nodes.

1. Regenerating Code

Data reliability is an important issue in CSSs. A self-sustaining CSS must be able to regenerate (namely, repair) failed nodes. The most common method employed to repair failed nodes is that which seeks to replicate data in multiple storage nodes; it is used in many practical storage systems [2].

As a generalization of replication, erasure coding can offer better storage efficiency by using maximum distance separable codes [2]. As reported in [3]–[6], for the same redundancy factor, erasure coding can obtain significantly higher reliability in comparison with replication. However, erasure coding requires more overhead in terms of the total repair bandwidth, γ, (namely, the amount of data downloaded to repair a failed node) than replication [7]. A promising optional coding scheme for distributed storage systems (DSSs) named "regenerating coding" was introduced in [2].

Regenerating code is efficient in terms of both storage and repair bandwidth. Under the definition of regenerating code [2], data reconstruction can be achieved by connecting to any k of n

nodes. When a node fails, a regeneration process is conducted to repair the data stored in the failed node. Any d of the $(n-1)$ remaining nodes can be selected by the replacement node as helper nodes and β ($\leq \alpha$) symbols are downloaded from each. The total repair bandwidth, $\gamma = d\beta$, is much smaller than the message size, F. A regenerating code under the above setup is called strict-regenerating code (S-RC). The parameters involved must satisfy a bound given by [2]

$$F \leq \sum_{i=1}^{k} \min\{\alpha, (d-i+1)\beta\}. \quad (1)$$

An optimal tradeoff exists between storage and repair bandwidth. The two extremal points in this tradeoff are termed as the minimum storage regenerating (MSR) point and the minimum bandwidth regenerating (MBR) point. MSR and MBR correspond to the best storage efficiency and the minimum repair bandwidth, respectively. Explicit constructions of MSR and MBR codes can be found in [8]–[11]. Specifically, optimal regenerating codes for homogeneous DSSs were constructed by using a product-matrix (PM) framework in [8]. These PM-based regenerating codes possess the property of *striping of data*, which results in a low complexity from an implementation standpoint. In [11], new encoding schemes for error-correcting MSR and MBR codes that generalize earlier results on error-correcting regenerating codes were proposed. Constructions for exact-regenerating codes between MSR and MBR points were given in [12].

Data security is another important issue for CSSs. When a CSS consists of nodes widely spread across the Internet, some nodes may be not secure. In [13], the problem of securing DSSs against passive eavesdroppers that can observe a limited number of storage nodes was studied and a general upper bound on the secrecy capacity was derived. Information-theoretically secure regenerating codes, which achieve an information-theoretic secrecy capacity, can be found in [14]. A link eavesdropping problem for remote DSSs, where data is stored in two geographically different data centers to increase its reliability, was studied in [1].

2. Heterogeneous DSSs

Earlier researches focus on a homogeneous DSS model in which all nodes have the same node storage capacity, α, and repair bandwidth, β. Studies on heterogeneous systems that contain nodes from different sources and with different characteristics have recently emerged and been developed. Examples include peer-to-peer (p2p) or hybrid (p2p-assisted) CSSs [15], [16], Internet caching systems for video-on-demand applications [17], [18], and caching systems in heterogeneous wireless networks [19].

Motivated by the above heterogeneous applications, researches on heterogeneous DSSs are emerging. The capacity of heterogeneous DSSs with different storage sizes and repair bandwidths was studied in [20]. The work of [20] focuses on characterizing the upper and lower bounds of the capacity of a DSS. In [21], the fundamental tradeoff between system storage cost and system repair cost was investigated for heterogeneous DSSs with different storage and repair costs. Considering heterogeneous DSSs with dynamic repair bandwidth and dynamic storage capacity, the authors of [22] investigated the fundamental tradeoff between storage and repair cost with flexible reconstruction degree. Coding schemes for a DSS with one super-node (that is, the node that is the most reliable and having the largest storage capacity among all other nodes) were studied in [23]. For heterogeneous CSSs, Yu and others considered irregular fractional repetition codes to provide very low repair cost and less disk I/O access at the expense of higher storage overhead [24]. The authors of [25] and [26] studied the storage allocation problem of heterogeneous DSSs under a total storage budget constraint, where nodes may fail with different probabilities.

Researches on regenerating codes for heterogeneous DSSs are relatively sparse. In [27], a two-rack model was designed to investigate the tradeoff between storage and repair bandwidth. It was shown that all the points on the tradeoff curve, including the MBR point, become feasible if the nodes in the rack with higher regenerating bandwidth are allowed to store more information. In [28], relax-regenerating code (R-RC) construction was proposed for heterogeneous DSSs with different repair bandwidths under a more relaxed setup. Provided that the total data downloaded exceeds a certain threshold, a DC (or a replacement node) can succeed in data reconstruction (or regeneration) irrespective of the number of nodes to which it connects [28]. However, the repair bandwidth of R-RC construction meets the *cut-set bound* only when the maximum flexibility of regeneration is allowed. So far, it is still challenging to construct S-RC for applications of heterogeneous DSSs.

3. Related Terminology

A. Repair Bandwidth

The amount of data that a replacement node downloads from all d helper nodes during a repair process is defined as *total repair bandwidth*, γ. The amount of data that a replacement node downloads from helper node i during a repair process is called the repair bandwidth, β_i ($\leq \alpha$), of node i. The maximum and minimum values of $\beta_i, i \in [1, n]$, are denoted by $\beta_{\max}$ and $\beta_{\min}$, respectively. Note that in the heterogeneous system considered in this paper, different nodes have different

repair bandwidths, while the node storage capacities, α_i, of different nodes are the same; that is, $\alpha_i = \alpha$. In addition, we focus on the case of single-node failure because it is the dominant failure case in DSSs [29].

B. Striping of Data

The striping property [8] can be explained as follows. Given an optimal regenerating code with parameter set $\{\alpha, \beta_{\max}, F\}$, a second optimal regenerating code with parameter set $\{\sigma\alpha, \sigma\beta_{\max}, \sigma F\}$ for any positive integer σ can be obtained. In reality, a big file (message) of size σF symbols can be divided into σ groups with F symbols per group. For each group, the $\{\alpha, \beta_{\max}, F\}$ code can be applied independently. In general, a code constructed for a smaller $\beta_{\max}$ through concatenation of codes will be of lesser complexity from a practical standpoint [8]. Such a process is referred to as *striping of data*. For these reasons, we design regenerating code for $\beta_{\min} = 1$ in the present paper.

4. Results Presented in This Paper

In this paper, we consider a heterogeneous DSS model, where two geographically different datacenters are deployed and storage nodes are configured with different repair bandwidths but identical storage capacities. An extended product-matrix (EPM) framework is proposed to construct S-RC for heterogeneous DSS applications. Applying the EPM framework to the heterogeneous DSS considered, we provide an explicit construction of minimum remote-repair bandwidth regenerating (MRBR) codes. The parameters of the MRBR codes can meet a cut-set bound. Strict mathematical proofs are provided for the data reconstruction and regeneration properties of our MRBR codes. An example for the MRBR code is also presented for the purposes of demonstration.

II. System Model and MRBR Code

1. System Model and Analysis

The system model considered in this paper is based on the two-datacenter scenario introduced in [1]. The data are stored in two geographically different datacenters, named local storage datacenter (LSD) and remote storage datacenter (RSD). There are N_L and N_R storage nodes in the LSD and the RSD, respectively. The total number of storage nodes is given by

$$n = N_L + N_R . \tag{2}$$

In this model, a DC connects to any k nodes in the LSD through infinite-capacity links such that it can download all the data stored in the k nodes and reconstruct an original file. When a node in the LSD fails, inter-datacenter links (IDLs) between the LSD and the RSD are established for data repair. The replacement node for the failed node connects to N'_L helper nodes from the LSD and N'_R helper nodes from the RSD so as to regenerate the data stored in the failed node. The total number of helper nodes is given by

$$d = N'_L + N'_R . \tag{3}$$

The replacement node in the LSD downloads β_L symbols from each helper node in the LSD and β_R symbols from each helper node in the RSD. As in [1], we assume that

$$\beta_L = m\beta_R \qquad (m \geq 1), \tag{4}$$

where m is an integer. Clearly, the system model considered is a heterogeneous CSS when $m > 1$ holds. In addition, the remote repair bandwidth, γ_R, is defined as

$$\gamma_R = \beta_R N'_R . \tag{5}$$

The communication between the LSD and the RSD over IDLs can be susceptible to eavesdropping. As in [1], we assume that

$$N'_R = N_R \tag{6}$$

and that an eavesdropper can obtain all the information of γ_R sent from the RSD. Thus, γ_R is a dominant factor affecting the security level of the CSS. So, it is necessary to minimize γ_R such that the system security can be improved.

2. MRBR Code

For the considered system model, there is a fundamental tradeoff curve between α and γ_R [1]. On the tradeoff curve, there exists an extremal point that corresponds to the minimum remote-repair bandwidth ($\gamma_{R,\min}$). This point is named MRBR point. An S-RC code becomes known as an MRBR code if it attains this point. Note that $\beta_{\min} = \beta_R$ and $\beta_{\max} = \beta_L$. According to the striping of data, $\beta_R = 1$ is assumed in all MRBR codes hereafter.

We denote an MRBR code over a finite field $\mathbb{F}_q$ of size q as an $[n, k, d, m, N'_R]$ regenerating code with parameter set (α, β_R, F). We assume that the parameters k and d are the minimum values that can always guarantee the data reconstruction and regeneration properties of an MRBR code. This assumption means that the ranges of d and N'_R will be $k \leq d \leq n-1$ (see [8] for details) and $0 < N'_R \leq n-k$, respectively. There is an upper bound for the parameter N'_R because a DC needs to connect k nodes in the LSD for reconstruction. Thus, there are at most $(n-k)$ storage nodes in the RSD.

According to [1], the fundamental tradeoff curve between α and γ_R is subject to

$$F \le \sum_{i=0}^{k-1} \min\{\alpha, (N'_{\mathrm{L}} - i)\beta_{\mathrm{L}} + N'_{\mathrm{R}}\beta_{\mathrm{R}}\}. \quad (7)$$

That is, the parameters of a regenerating code must satisfy (7). Here, we define

$$F_{\mathrm{C}} \overset{\Delta}{=} \sum_{i=0}^{k-1} \min\{\alpha, (N'_{\mathrm{L}} - i)\beta_{\mathrm{L}} + N'_{\mathrm{R}}\beta_{\mathrm{R}}\} \quad (8)$$

as the capacity of $[n, k, d, m, N'_{\mathrm{R}}]$ regenerating codes. The minimum storage, α^*, was achieved in [1] as

$$\alpha^* = \begin{cases} F/k & \gamma_{\mathrm{R}} \in [f(0), \infty) \\ (F - g(i)\gamma_{\mathrm{R}})/(k-i) & \gamma_{\mathrm{R}} \in [f(i), f(i-1)), \end{cases} \quad (9)$$

where

$$f(i) = 2FN'_{\mathrm{R}} / \left[i(2mk - mi - m) + 2k(md - mk + m - N'_{\mathrm{R}}) \right], \quad (10)$$

$$g(i) = i\left[2md - 2mk - 2(m-1)N'_{\mathrm{R}} + mi + m\right] / 2N'_{\mathrm{R}}, \quad (11)$$

$$\gamma_{\mathrm{R,min}} = 2FN'_{\mathrm{R}} / k(2md - mk + m - 2N'_{\mathrm{R}}). \quad (12)$$

The MRBR point corresponding to $\gamma_{\mathrm{R,min}}$ can be achieved by

$$\alpha_{\mathrm{MRBR}} = \frac{F}{k} \times \frac{2\left[k(m-2) - (m-1)\right]N'_{\mathrm{R}} + 2md}{2md - mk + m - 2N'_{\mathrm{R}}},$$
$$\gamma_{\mathrm{R,\,MRBR}} = \frac{F}{k} \times \frac{2N'_{\mathrm{R}}}{2md - mk + m - 2N'_{\mathrm{R}}}, \quad (13)$$

where α_{MRBR} and $\gamma_{\mathrm{R,\,MRBR}}$ represent the storage capacity and the remote-repair bandwidth of each node, respectively. Note that $\gamma_{\mathrm{R,MRBR}} = \gamma_{\mathrm{R,min}} = \beta_{\mathrm{R}} N'_{\mathrm{R}}$. The complete derivation of (13) can be found in Section 1 of the Appendix.

Now, we show the optimality of all MRBR codes. One can find that an MRBR code satisfies the following two properties:

- An MRBR code satisfying (13) achieves the cut-set bound of the tradeoff between γ_{R} and α with equality.
- Decreasing α or β_{R} will result in a new parameter set that violates the cut-set bound. According to the definition of optimal regenerating code [8], the MRBR code is optimal.

III. EPM Framework

1. Overview of PM Framework

The PM framework was presented in [8] to construct regenerating codes. Under the PM framework, each codeword in a DSS can be described as follows: $\mathbf{C} = \mathbf{\Psi M}$, where $\mathbf{\Psi}$ is an encoding matrix of size $(n \times d)$, $\mathbf{M}$ is a message matrix of size $(d \times \alpha)$, and $\mathbf{C}$ is a code matrix of size $(n \times \alpha)$. Specifically, $\mathbf{M}$ contains the F message symbols in a possibly redundant fashion, and $\mathbf{\Psi}$ is designed in advance and independent of the message symbols. We denote the ith row of $\mathbf{\Psi}$ by $\boldsymbol{\psi}_i$ and the ith row of $\mathbf{C}$ by $\mathbf{c}_i$; then, it is clear that $\mathbf{c}_i = \boldsymbol{\psi}_i \mathbf{M}$, $1 \le i \le n$. In fact, $\boldsymbol{\psi}_i$ is the encoding vector of node i, and $\mathbf{c}_i$ is the coded message vector stored in node i. All message symbols and coded symbols belong to a finite field $\mathbb{F}_q$ of size q.

The operations of data-reconstruction and regeneration proceed in the following way.

Data-reconstruction. We denote a set of k nodes to which a DC connects as $\{1, \ldots, k\}$. The ith node in this set passes on the message vector $\boldsymbol{\psi}_i \mathbf{M}$ to the DC. Thus, from the k nodes, the DC obtains the product matrix $\mathbf{\Psi}_{\mathrm{DC}} \mathbf{M}$, where $\mathbf{\Psi}_{\mathrm{DC}}$ is a submatrix of $\mathbf{\Psi}$ consisting of rows corresponding to the k nodes. Based on the properties of $\mathbf{\Psi}$ and $\mathbf{M}$, the original message can be recovered from $\mathbf{\Psi}_{\mathrm{DC}} \mathbf{M}$. The accurate procedure of recovering $\mathbf{M}$ is dependent on the particular construction.

Regeneration. Assume that a node, f, fails and that a replacement node tries to regenerate the data stored in node f; that is, $\mathbf{c}_f = \boldsymbol{\psi}_f \mathbf{M}$. We denote an arbitrary subset $\{1, \ldots, d\}$ of d helper nodes to which the replacement node connects. The jth helper node in the subset computes the inner product $\mathbf{c}_j \mathbf{u}_f^{\mathrm{T}}$ (namely, one symbol; here, T denotes transpose) and sends it to the replacement node. Note that $\mathbf{u}_f$ is a row vector consisting of α components of $\boldsymbol{\psi}_f$. Thus, from the d helper nodes, the replacement node obtains d symbols, which can be expressed as $\mathbf{\Psi}_{\mathrm{repair}} \mathbf{M} \mathbf{u}_f^{\mathrm{T}}$, where $\mathbf{\Psi}_{\mathrm{repair}}$ is a submatrix of $\mathbf{\Psi}$ consisting of rows corresponding to the d helper nodes. From $\mathbf{\Psi}_{\mathrm{repair}} \mathbf{M} \mathbf{u}_f^{\mathrm{T}}$, the placement node can exactly recover the data stored previously in the failed node f. Once again, the precise procedure depends on the specific constructions of $\mathbf{\Psi}$ and $\mathbf{M}$.

So far, it is clear that the PM framework is suitable for homogeneous DSSs in which all nodes have the same storage capacity α and the same repair bandwidth β.

In the following subsection, we propose an EPM framework for a heterogeneous system, where the repair bandwidth β_i may be different for different helper nodes while the node storage capacities α_i of all nodes are the same; namely, $\alpha_i = \alpha$.

2. EPM Framework

To highlight the difference between EPM and PM, a superscript letter "e" is adopted in notations. We denote $w = \max\{\beta_i, i \in [1, n]\}$ and $z = \alpha/w$. We set y to be an arbitrary integer. Then, under the EPM framework, each codeword in a DSS is given by

$$\mathbf{C}^{\mathrm{e}} = \mathbf{\Psi}^{\mathrm{e}} \mathbf{M}^{\mathrm{e}}, \quad (14)$$

where the encoding matrix $\mathbf{\Psi}^{\mathrm{e}}$ is of size $(n \times yw)$, the message

matrix $\mathbf{M}^{\mathrm{e}}$ is of size ($yw \times (zw = \alpha)$), and the code matrix $\mathbf{C}^{\mathrm{e}}$ is of size ($n \times \alpha$). As in the PM framework, $\boldsymbol{\psi}_i^{\mathrm{e}}$ stands for the ith row of $\boldsymbol{\Psi}^{\mathrm{e}}$ and represents the encoding vector of node i, and $\mathbf{c}_i^{\mathrm{e}}$ stands for the ith row of $\mathbf{C}^{\mathrm{e}}$ and represents the coded message vector stored in node i. The encoding matrix $\boldsymbol{\Psi}^{\mathrm{e}}$ is of the form

$$\boldsymbol{\Psi}^{\mathrm{e}} = [\boldsymbol{\Psi}_1^{\mathrm{e}} \ \boldsymbol{\Psi}_2^{\mathrm{e}} \ \cdots \ \boldsymbol{\Psi}_w^{\mathrm{e}}], \tag{15}$$

where each submatrix $\boldsymbol{\Psi}_i^{\mathrm{e}}, i \in [1, w]$ is of size ($n \times y$). As will be shown in Section IV, specific requirements can be imposed on these w submatrices for particular code designs.

Equivalently, the encoding matrix $\boldsymbol{\Psi}^{\mathrm{e}}$ can also be expressed as

$$\boldsymbol{\Psi}^{\mathrm{e}} = \sum_{j=1}^{w} \boldsymbol{\Phi}_j, \tag{16}$$

where

$$\boldsymbol{\Phi}_j = \left[\underbrace{\mathbf{O}^{n\times y} \ \ldots \ \mathbf{O}^{n\times y}}_{j-1} \quad \boldsymbol{\Psi}_j^{\mathrm{e}} \quad \underbrace{\mathbf{O}^{n\times y} \ \ldots \ \mathbf{O}^{n\times y}}_{w-j} \right], \tag{17}$$

where $\mathbf{O}^{n\times y}$ is a null matrix of size ($n \times y$).

The message matrix $\mathbf{M}^{\mathrm{e}}$ is a block diagonal matrix given by

$$\mathbf{M}^{\mathrm{e}} = \begin{bmatrix} \mathbf{M}_1^{\mathrm{e}} & & & & \\ & \mathbf{M}_2^{\mathrm{e}} & & 0 & \\ & & \mathbf{M}_3^{\mathrm{e}} & & \\ & 0 & & \ddots & \\ & & & & \mathbf{M}_w^{\mathrm{e}} \end{bmatrix}, \tag{18}$$

where each submatrix $\mathbf{M}_j^{\mathrm{e}}, j \in [1, w]$ is of size ($y \times z$). These w submatrices contain F message symbols in a possibly redundant fashion.

According to (15) and (18), $\mathbf{C}^{\mathrm{e}}$ can be expressed as

$$\mathbf{C}^{\mathrm{e}} = \boldsymbol{\Psi}^{\mathrm{e}}\mathbf{M}^{\mathrm{e}} = \left[\boldsymbol{\Psi}_1^{\mathrm{e}}\mathbf{M}_1^{\mathrm{e}} \ \boldsymbol{\Psi}_2^{\mathrm{e}}\mathbf{M}_2^{\mathrm{e}} \ \cdots \ \boldsymbol{\Psi}_w^{\mathrm{e}}\mathbf{M}_w^{\mathrm{e}} \right]. \tag{19}$$

From (19), the coded message vector stored by the ith node is given by

$$\mathbf{c}_i^{\mathrm{e}} = \left[\boldsymbol{\psi}_{1,i}^{\mathrm{e}}\mathbf{M}_1^{\mathrm{e}} \ \boldsymbol{\psi}_{2,i}^{\mathrm{e}}\mathbf{M}_2^{\mathrm{e}} \ \cdots \ \boldsymbol{\psi}_{w,i}^{\mathrm{e}}\mathbf{M}_w^{\mathrm{e}} \right], \tag{20}$$

where $\boldsymbol{\psi}_{j,i}^{\mathrm{e}}$ represents the ith row of submatrix $\boldsymbol{\Psi}_j^{\mathrm{e}}$, $j \in [1, w]$. Furthermore, (20) can be expressed as

$$\mathbf{c}_i^{\mathrm{e}} = \sum_{j=1}^{w} \mathbf{c}_{j,i}^{\mathrm{e}} = \sum_{j=1}^{w} \boldsymbol{\varphi}_{j,i}\mathbf{M}^{\mathrm{e}}, \tag{21}$$

where $\boldsymbol{\varphi}_{j,i}$ represents the ith row of $\boldsymbol{\Phi}_j, j \in [1, w]$ and

$$\mathbf{c}_{j,i}^{\mathrm{e}} = \boldsymbol{\varphi}_{j,i}\mathbf{M}^{\mathrm{e}} = \left[\underbrace{\mathbf{o}^{1\times z} \ \ldots \ \mathbf{o}^{1\times z}}_{j-1} \quad \boldsymbol{\psi}_{j,i}^{\mathrm{e}}\mathbf{M}_j^{\mathrm{e}} \quad \underbrace{\mathbf{o}^{1\times z} \ \ldots \ \mathbf{o}^{1\times z}}_{w-j} \right], \tag{22}$$

is the jth message vector component. Here, $\mathbf{o}^{1\times z}$ is a null vector of size ($1 \times z$). Thus, from the message vector $\mathbf{c}_i^{\mathrm{e}}$, the ith node can easily achieve any message vector component $\mathbf{c}_{j,i}^{\mathrm{e}}$ $j \in [1, w]$. This is termed as a decomposition property of our EPM framework, which is important for the following processes of data-reconstruction and regeneration.

Data-reconstruction. We denote an arbitrary subset {1, ... , k} of k nodes to which a DC connects. The ith node in this subset passes on the message vector $\mathbf{c}_i^{\mathrm{e}}$ to the DC. The decomposition property of our EPM framework makes it easy to obtain the w message vector components $\mathbf{c}_{1,i}^{\mathrm{e}}, \mathbf{c}_{2,i}^{\mathrm{e}}, \ \ldots \ , \mathbf{c}_{w,i}^{\mathrm{e}}$ from $\mathbf{c}_i^{\mathrm{e}}$. Thus, from the k nodes, the DC obtains the product matrix $\boldsymbol{\Psi}_{\mathrm{DC}}^{\mathrm{e}}\mathbf{M}^{\mathrm{e}}$, where $\boldsymbol{\Psi}_{\mathrm{DC}}^{\mathrm{e}}$ is a ($kw \times yw$) matrix consisting of the kw rows $\{\boldsymbol{\varphi}_{1,1}, \ldots, \boldsymbol{\varphi}_{w,1}, \ldots, \boldsymbol{\varphi}_{1,k}, \ldots, \boldsymbol{\varphi}_{w,k}\}$. Based on the properties of $\boldsymbol{\Psi}^{\mathrm{e}}$ and $\mathbf{M}^{\mathrm{e}}$, the original message can be recovered from $\boldsymbol{\Psi}_{\mathrm{DC}}^{\mathrm{e}}\mathbf{M}^{\mathrm{e}}$.

Regeneration. Assume that the node f fails. Hence, a replacement node tries to regenerate the data stored in node f; that is, $\mathbf{c}_f^{\mathrm{e}} = \boldsymbol{\psi}_f^{\mathrm{e}}\mathbf{M}^{\mathrm{e}}$. We denote an arbitrary subset {1 , ... , d} of d helper nodes to which the replacement node connects. According to the decomposition property of our EPM framework, the helper node j in this subset can obtain β_j ($\leq w$) message vector components $\mathbf{c}_{1,j}^{\mathrm{e}}, \ldots, \mathbf{c}_{\beta_j,j}^{\mathrm{e}}$ from $\mathbf{c}_j^{\mathrm{e}}$. The helper node j calculates β_j inner products $\mathbf{c}_{1,j}^{\mathrm{e}}(\mathbf{u}_f^{\mathrm{e}})^{\mathrm{T}}, \ldots, \mathbf{c}_{\beta_j,j}^{\mathrm{e}}(\mathbf{u}_f^{\mathrm{e}})^{\mathrm{T}}$ (namely, β_j symbols), where $\mathbf{u}_f^{\mathrm{e}}$ is a row vector consisting of α components of $\boldsymbol{\psi}_f^{\mathrm{e}}$, and passes on these inner products to the replacement node. Thus, from the d helper nodes, the replacement node obtains

$$d' = \sum_{j=1}^{d} \beta_j \tag{23}$$

symbols, which can be expressed as $\boldsymbol{\Psi}_{\mathrm{repair}}^{\mathrm{e}}\mathbf{M}^{\mathrm{e}}(\mathbf{u}_f^{\mathrm{e}})^{\mathrm{T}}$, where $\boldsymbol{\Psi}_{\mathrm{repair}}^{\mathrm{e}}$ is a $(d' \times yw)$ matrix consisting of the d' row $\{\boldsymbol{\varphi}_{1,1}, \ldots, \boldsymbol{\varphi}_{\beta_1,1}, \ldots, \boldsymbol{\varphi}_{1,d}, \ldots, \boldsymbol{\varphi}_{\beta_d,d}\}$. From $\boldsymbol{\Psi}_{\mathrm{repair}}^{\mathrm{e}}\mathbf{M}^{\mathrm{e}}(\mathbf{u}_f^{\mathrm{e}})^{\mathrm{T}}$, based on the properties of $\boldsymbol{\Psi}^{\mathrm{e}}$ and $\mathbf{M}^{\mathrm{e}}$, the placement node can exactly recover the data stored previously in the failed node f.

EPM Framework versus PM Framework. Generally, the EPM framework is suitable for heterogeneous DSSs, where the repair bandwidths of storage nodes are different. When the repair bandwidths of all storage nodes are the same (namely, $\beta_i = \beta, \forall\ i$), the DSS model considered turns out to be a homogeneous one. In this special case, we have $w = 1$, and the encoding, data-reconstruction, and regeneration under the EPM framework are identical to those under the PM framework. This means that the EPM framework can be applied to homogeneous systems also.

IV. EPM-MRBR Code Construction

In this section, we apply the EPM framework to construct a

class of MRBR codes with $\beta_R = 1$, named "EPM-MRBR code," for our considered two-datacenter model. This construction is under the following conditions:

$$\begin{cases} m = 2, \\ k/n = 1/2, \\ N'_L / N_L = 1/2. \end{cases} \tag{24}$$

An MRBR code must possess data-reconstruction and regeneration properties. For an MRBR code, the parameter α corresponds to α_{MRBR} in (13). Thus, we can obtain the following:

$$\begin{cases} F = (\alpha - k + 1)k, \\ \alpha = 2d - N'_R, \end{cases} \tag{25}$$

$$\begin{cases} F = (k+1)k, \\ \alpha = n. \end{cases} \tag{26}$$

Derivations of (25) and (26) are provided in Section 2 of the Appendix.

Based on the second and third conditions in (24), and (26), it can be observed that the values of parameters n, α, N_L, F are even numbers. According to (4) and the first condition in (24), we can determine that $\beta_L = 2\beta_R$. Thus, we have $w = 2$ in the present code with $\beta_R = 1$.

Based on (23), (3), (4), and the first condition in (24), it can be determined further that

$$d' = \sum_{j=1}^{d} \beta_j = \beta_L N'_L + \beta_R N'_R = mN'_L + N'_R = 2N'_L + N'_R. \tag{27}$$

For the case $w = 2$ of the EPM framework, we let

$$y = d'/w = d'/2 \quad \text{and} \quad z = \alpha/w = \alpha/2. \tag{28}$$

From (27), the third condition in (24), (6), and (2), it can be determined that

$$d' = n. \tag{29}$$

It can be determined from (2), (29), and (6) that d', N_R, and N'_R are also even numbers. Thus, the parameter set of the MRBR code that will be constructed here is

$$\left(\alpha = n = 2k = d', \beta_R = 1, F = k(k+1)\right). \tag{30}$$

Substituting $w = 2$ into (15), the encoding matrix $\mathbf{\Psi}^e$ of the MRBR code that will be constructed here can be obtained by

$$\mathbf{\Psi}^e = [\mathbf{\Psi}_1^e, \mathbf{\Psi}_2^e]. \tag{31}$$

Assume that the first N_L of n nodes are located in the LSD and the last N_R nodes are located in the RSD. The matrix $\mathbf{\Psi}^e = [\mathbf{\Psi}_1^e, \mathbf{\Psi}_2^e]$ is chosen such that the two ($n \times k$) matrices $\mathbf{\Psi}_1^e$ and $\mathbf{\Psi}_2^e$ satisfy the following properties:

- The last N'_R rows are linearly independent with any N'_L rows of the first $N_L (> N'_L)$ rows (note that $N'_R = N_R$ as mentioned above).
- Any k rows of the first $N_L (\geq k)$ rows are linearly independent.

The above requirements can be met by choosing $\mathbf{\Psi}^e$ to be a Vandermonde matrix with elements chosen carefully, where the finite field $\mathbb{F}_q$ is of size n or higher. Please refer to the choice of the encoding matrix $\mathbf{\Psi}$ of the PM framework in [8]. For example, we can choose $\mathbf{\Psi}^e$ to be a Vandermonde matrix, as follows:

$$\mathbf{\Psi}^e = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2^2 & \cdots & 2^{n-1} \\ 1 & 3 & 3^2 & \cdots & 3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & n & n^2 & \cdots & n^{n-1} \end{bmatrix}, \tag{32}$$

whereby this matrix is of full rank. Thus, the matrix $\mathbf{\Psi}^e$ implies that the two matrices $\mathbf{\Psi}_1^e$ and $\mathbf{\Psi}_2^e$ satisfy the listed requirements above. Note that all the elements of $\mathbf{\Psi}^e$ are over a finite field $\mathbb{F}_q$ of size q.

Substituting (28) into (18), the $(d' = \alpha) \times \alpha$ message matrix $\mathbf{M}^e$ of an MRBR code can be obtained. The matrix satisfies the property of symmetry and contains $F = k(k + 1)$ message symbols from the message set $\{u_i\}_{i=1}^{F}$.

The reconstruction and exact-regeneration properties of an MRBR code will be given by the following two theorems.

Theorem 1 (EPM-MRBR Data-Reconstruction). The decomposition property of the EPM framework allows node i to achieve $w = 2$ message vector components $\mathbf{c}_{1,i}^e$ and $\mathbf{c}_{2,i}^e$ from $\mathbf{c}_i^e$. Thus, the message matrix $\mathbf{M}^e$ can be recovered by connecting to any k nodes in the LSD.

Proof. In the case of $w = 2$ under the EPM framework, the DC achieves the product matrix $\mathbf{\Psi}_{DC}^e \mathbf{M}^e$ from the k nodes in the LSD, where $\mathbf{\Psi}_{DC}^e$ is a ($2k \times d'$) matrix consisting of the $2k$ rows $\{\boldsymbol{\varphi}_{1,1}, \boldsymbol{\varphi}_{2,1}, \ldots, \boldsymbol{\varphi}_{1,k}, \boldsymbol{\varphi}_{2,k}\}$.

According to the specific construction and properties of matrix $\mathbf{\Psi}^e$, the $((2k = d') \times d')$ matrix $\mathbf{\Psi}_{DC}^e$ is constructed to be of full rank; thus, it is invertible. Therefore, the DC can recover $\mathbf{M}^e$ by multiplying the matrix $\mathbf{\Psi}_{DC}^e \mathbf{M}^e$ on the left by $(\mathbf{\Psi}_{DC}^e)^{-1}$. ■

Theorem 2 (EPM-MRBR Exact-Regeneration). The exact-regeneration of any failed node in the LSD can be achieved by downloading $\beta_L = 2$ symbols from each of the helper nodes in the LSD and $\beta_R = 1$ symbol from each of the helper nodes in the RSD.

Proof. The replacement node of the failed node f connects to an arbitrary subset $\{1, \ldots, N'_L + N'_R\}$ of $N'_L + N'_R = d$ helper nodes. It is assumed that the helper node i, $i \in [1, N'_L]$

is a node in the LSD and that the helper node j, $j \in [N'_{\mathrm{L}}+1, N'_{\mathrm{L}}+N'_{\mathrm{R}}]$ is a node in the RSD. According to the decomposition property of the EPM framework, the helper node j in this subset can achieve the two message vector components $\mathbf{c}^{\mathrm{e}}_{1,j}$ and $\mathbf{c}^{\mathrm{e}}_{2,j}$. In the present construction, we choose the vector $\mathbf{u}^{\mathrm{e}}_f$ equal to $\boldsymbol{\psi}^{\mathrm{e}}_f$. Thus, the helper node i from the LSD calculates two inner products; that is, $\mathbf{c}^{\mathrm{e}}_{1,i}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}}$ and $\mathbf{c}^{\mathrm{e}}_{2,i}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}}$, $i \in [1, N'_{\mathrm{L}}]$. The helper node j from the RSD randomly calculates one of two inner products; that is, either $\mathbf{c}^{\mathrm{e}}_{1,j}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}}$ or $\mathbf{c}^{\mathrm{e}}_{2,j}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}}$, $j \in [N'_{\mathrm{L}}+1, N'_{\mathrm{L}}+N'_{\mathrm{R}}]$, It is worth noting that, in RSD, there are $N'_{\mathrm{R}}/2$ helper nodes calculating $\mathbf{c}^{\mathrm{e}}_{1,j}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}}$ and the other $N'_{\mathrm{R}}/2$ helper nodes calculating $\mathbf{c}^{\mathrm{e}}_{2,j}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}}$. Then, the d helper nodes pass on the $2N'_{\mathrm{L}}+N'_{\mathrm{R}}=d'$ inner product values to the replacement node; thus, the replacement node obtains d' symbols, which can be expressed as $\boldsymbol{\Psi}^{\mathrm{e}}_{\mathrm{repair}}\mathbf{M}^{\mathrm{e}}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}}$, where $\boldsymbol{\Psi}^{\mathrm{e}}_{\mathrm{repair}}$ is a $(d' \times d')$ matrix consisting of the following d' rows: $\{\boldsymbol{\varphi}_{1,1}, \boldsymbol{\varphi}_{2,1}, \ldots, \boldsymbol{\varphi}_{1,N'_{\mathrm{L}}}, \boldsymbol{\varphi}_{2,N'_{\mathrm{L}}}, \boldsymbol{\varphi}_{1,N'_{\mathrm{L}}+1}, \boldsymbol{\varphi}_{2,N'_{\mathrm{L}}+2}, \ldots, \boldsymbol{\varphi}_{1,N'_{\mathrm{L}}+N'_{\mathrm{R}}-1}, \boldsymbol{\varphi}_{2,N'_{\mathrm{L}}+N'_{\mathrm{R}}}\}$.

According to the specific construction and properties of matrix $\boldsymbol{\Psi}^{\mathrm{e}}$, the matrix $\boldsymbol{\Psi}^{\mathrm{e}}_{\mathrm{repair}}$ is constructed to be of full rank; thus, it is invertible. This allows the replacement node to recover the $\mathbf{M}^{\mathrm{e}}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}}$ by multiplying the matrix $\boldsymbol{\Psi}^{\mathrm{e}}_{\mathrm{repair}}\mathbf{M}^{\mathrm{e}}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}}$ on the left by $(\boldsymbol{\Psi}^{\mathrm{e}}_{\mathrm{repair}})^{-1}$. Since $\mathbf{M}^{\mathrm{e}}$ is symmetric, we have $(\mathbf{M}^{\mathrm{e}}(\boldsymbol{\psi}^{\mathrm{e}}_f)^{\mathrm{T}})^{\mathrm{T}} = \boldsymbol{\psi}^{\mathrm{e}}_f\mathbf{M}^{\mathrm{e}}$; this is the data stored previously in the failed node f. ■

Example Implementation of EPM-MRBR Code. We construct an EPM-MRBR code whose parameters are configured as follows: $n = 6$, $k = 3$, $d = 4$, $m = 2$, $N'_{\mathrm{R}} = 2$.

It is clear that $\alpha = d' = 6$, $F = 12$, and $N'_{\mathrm{L}} = 2$. Let us choose $q = 7$; that is, we are operating over $\mathbb{F}_7$. The matrices $\mathbf{M}^{\mathrm{e}}_1$ and $\mathbf{M}^{\mathrm{e}}_2$ are filled up by the twelve message symbols from the message set $\{u_i\}_1^{12}$ given by

$$\mathbf{M}^{\mathrm{e}}_1 = \begin{bmatrix} u_1 & u_2 & u_3 \\ u_2 & u_4 & u_5 \\ u_3 & u_5 & u_6 \end{bmatrix}, \quad \mathbf{M}^{\mathrm{e}}_2 = \begin{bmatrix} u_7 & u_8 & u_9 \\ u_8 & u_{10} & u_{11} \\ u_9 & u_{11} & u_{12} \end{bmatrix}. \quad (33)$$

Choose $\boldsymbol{\Psi}^{\mathrm{e}}$ to be the (6 × 6) Vandermonde matrix over $\mathbb{F}_7$ given by

$$\boldsymbol{\Psi}^{\mathrm{e}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 & 2 & 4 \\ 1 & 3 & 2 & 6 & 4 & 5 \\ 1 & 4 & 2 & 1 & 4 & 2 \\ 1 & 5 & 4 & 6 & 2 & 3 \\ 1 & 6 & 1 & 6 & 1 & 6 \end{bmatrix}. \quad (34)$$

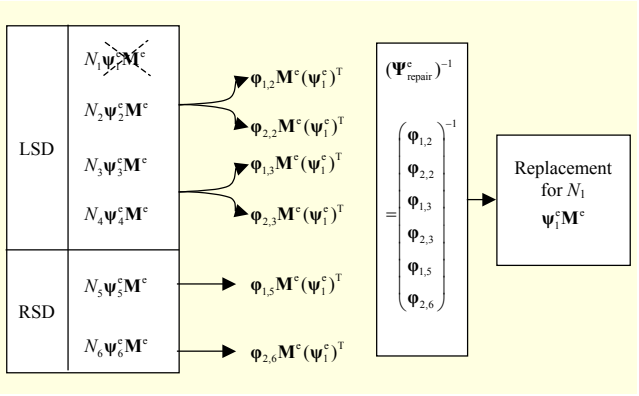

Fig. 1. Example for EMP-MRBR code construction: notions $N_1, \ldots, N_6$ represent nodes $1, \ldots, 6$. On failure of node 1, replacement node downloads $\beta_{\mathrm{L}} = 2$ symbols from each of nodes 2, 3 (in LSD) and downloads $\beta_{\mathrm{R}} = 1$ symbol from each of nodes 5, 6 (in RSD), under which node 1 is exactly regenerated.

In Fig. 1, the exact-regeneration of failed node "1" is demonstrated, where $\boldsymbol{\psi}^{\mathrm{e}}_1 = [1\ 1\ 1\ 1\ 1\ 1]$ and the $\alpha = 6$ symbols stored in node 1 are $\mathbf{c}^{\mathrm{e}}_1 = \boldsymbol{\psi}^{\mathrm{e}}_1\mathbf{M}^{\mathrm{e}}$. Helper nodes 2 and 3 pass on inner products $\boldsymbol{\varphi}_{1,i}\mathbf{M}^{\mathrm{e}}[1\,1\,1\,1\,1\,1]^{\mathrm{T}}$ and $\boldsymbol{\varphi}_{2,i}\mathbf{M}^{\mathrm{e}}[1\,1\,1\,1\,1\,1]^{\mathrm{T}}$ (for i = 2, 3), and helper nodes 5 and 6 pass on inner products $\boldsymbol{\varphi}_{1,5}\mathbf{M}^{\mathrm{e}}[1\,1\,1\,1\,1\,1]^{\mathrm{T}}$ and $\boldsymbol{\varphi}_{2,6}\mathbf{M}^{\mathrm{e}}[1\,1\,1\,1\,1\,1]^{\mathrm{T}}$, to the replacement node to generate node 1. Then, the replacement node will multiply the six symbols it receives with $(\boldsymbol{\Psi}^{\mathrm{e}}_{\mathrm{repair}})^{-1}$, where $\boldsymbol{\Psi}^{\mathrm{e}}_{\mathrm{repair}}$ is a (6 × 6) matrix consisting of the six rows $\{\boldsymbol{\varphi}_{1,2}, \boldsymbol{\varphi}_{2,2}, \boldsymbol{\varphi}_{1,3}, \boldsymbol{\varphi}_{2,3}, \boldsymbol{\varphi}_{1,5}, \boldsymbol{\varphi}_{2,6}\}$, as explained in Theorem 2.

V. Analysis of EPM Framework and MRBR Code

1. Implementation Complexity of EPM-MRBR Code

Under the EPM framework, the MRBR code possesses several desirable properties, such as linearity, small field size, striping, and low complexity, as discussed below.

A. Linearity and Field Size

The MRBR code is linear on a finite field $\mathbb{F}_q$; that is, the stored symbol is a linear combination of the source symbols from the finite field $\mathbb{F}_q$. As mentioned previously, any field of size n or higher suffices in our MRBR code. By shrewdly choosing the matrices that meet the required properties, it may be possible to reduce the field size further.

B. Striping

In the presence of striping, the whole message is divided into

stripes of small sizes corresponding to $\beta_R = 1$. Since each stripe is of minimum size, the complexities of encoding, reconstruction, and regeneration operations are lowered considerably; further, the buffer sizes required at DCs and replacement nodes are also quite small. In practice, the stripes can be processed in parallel and efficiently by using GPU/FPGA/multi-core processors.

2. EPM Framework for Multiple Datacenters

The EPM framework can be applied to the coding of the two-datacenter scenario, as shown in Section IV. Applying the EPM framework to multi-datacenter scenarios depends on specific system constructions. For the purpose of clarity, a three-datacenter scenario is provided as example in Section 3 of the Appendix.

VI. Conclusion

In this paper, an EPM framework — a generalization of a previous PM framework — is proposed for general heterogeneous DSSs with different repair bandwidths but identical storage capacities. A feature of the EPM framework is that different amounts of repair data can be downloaded from different nodes. This feature leads to several desirable properties, such as the ability to take full advantage of different bandwidth resources.

We apply the EPM framework to a specific heterogeneous DSS, where data are distributed and stored on two datacenters. Then, an explicit construction of MRBR codes is provided for the case of $m = 2$, $k/n = 1/2$ and $N'_L/N_L = 1/2$. Strict mathematical proofs have been provided to show the reconstruction and regeneration properties of our MRBR code. An example implementation of the MRBR code is presented also. This MRBR code is the first optimal S-RC for heterogeneous DSSs with different repair bandwidths, where the storage nodes are located in two geographically different datacenters. As a kind of S-RC, our MRBR code possesses several desirable advantages, such as linearity, small field size, and striping. Thus, it can be implemented with low complexity in practice. Our results also prove that the MRBR point on the storage and remote-repair bandwidth tradeoff is achievable under the additional constraint of exact-regeneration and specific conditions of system parameters. In future work, we plan to investigate deterministic designs of MRBR code for arbitrary parameter values.

Appendix

1. Derivation of (13)

According to (12), it can be obtained that

$$\gamma_{R,MRBR} = \gamma_{R,\min} = 2FN'_R / k\left(2md - mk + m - 2N'_R\right). \quad (1.1)$$

According to (10), we have

$$f(0) = 2FN'_R / k\left(2md - 2mk + 2m - 2N'_R\right) > \gamma_{R,MRBR}. \quad (1.2)$$

At the MRBR point, we have $\alpha^* = \alpha_{MRBR}$. According to (1.2), (9) can be rewritten as

$$\alpha_{MRBR} = \alpha^*|_{\gamma_R = \gamma_{R,MRBR}} = \left(F - g(i)\gamma_{R,MRBR}\right)/(k-i), \quad (1.3)$$

where $\gamma_{R,MRBR} \in [f(i), f(i-1))$. When $i = k - 1$, it can be obtained from (10) that

$$f(k-1) = 2FN'_R / k\left(2md - mk + m - 2N'_R\right) = \gamma_{R,MRBR}. \quad (1.4)$$

Consequently, (1.3) can be rewritten as

$$\alpha_{MRBR} = \left(F - g(k-1)\gamma_{R,MRBR}\right)/\left(k-(k-1)\right). \quad (1.5)$$

Substituting (11) and (1.1) into (1.5), we can obtain

$$\alpha_{MRBR} = \frac{F}{k} \times \frac{2\left[k(m-2) - (m-1)\right]N'_R + 2md}{2md - mk + m - 2N'_R}. \quad (1.6)$$

Thus, (13) follows from (1.1) and (1.6).

2. Deriving Processes of (25) and (26)

According to the first condition in (24), we can rewrite (13) as

$$\alpha = \left(F\left(-N'_R + 2d\right)\right)/k\left(2d - k + 1 - N'_R\right), \quad (2.1)$$

$$\gamma_{R,MRBR} = FN'_R / k\left(2d - k + 1 - N'_R\right). \quad (2.2)$$

Combining (5) and (2.2) with $\beta_R = 1$, we can obtain

$$FN'_R / k\left(2d - k + 1 - N'_R\right) = \gamma_R = N'_R. \quad (2.3)$$

Then, we can obtain the following:

$$F = k\left(2d - k + 1 - N'_R\right). \quad (2.4)$$

Substituting (2.4) into (2.1), we have

$$\alpha = 2d - N'_R. \quad (2.5)$$

Let message symbols F satisfy the capacity as shown in (8). Then, replacing F_C with F in (8), we have

$$\begin{aligned} F &= \sum_{i=0}^{k-1} \min\left\{\alpha, \left(N'_L - i\right)\beta_L + N'_R\beta_R\right\} \\ &= \sum_{i=0}^{k-1} \min\left\{\alpha, 2\left(d - N'_R - i\right) + N'_R\right\} \\ &= \sum_{i=0}^{k-1} \left(2d - N'_R - 2i\right) = k\left(\alpha - k + 1\right), \end{aligned} \quad (2.6)$$

where the second equality in (2.6) follows from (3), (4), and the first condition in (24); moreover, the third and fourth equalities

in (2.6) follow from (2.5). Then, (25) follows from (2.5) and (2.6).

From (3), (2.5) can be rewritten as

$$\alpha = 2\left(N'_{\mathrm{L}} + N'_{\mathrm{R}}\right) - N'_{\mathrm{R}} = N_{\mathrm{L}} + N_{\mathrm{R}} = n, \quad (2.7)$$

where the second equality in (2.7) follows from (6) and the third condition in (24); moreover, the third equality in (2.7) follows from (2). Substituting (2.7) into (2.6) under the second condition in (24), we have

$$F = k\left(k+1\right). \quad (2.8)$$

Then, (26) follows from (2.7) and (2.8).

3. EPM Framework for Three-Datacenter Scenario

Consider a three-datacenter scenario where data are distributed and stored on three geographically different datacenters. There are N_{L} storage nodes in a local storage datacenter (LSD), N_{R1} nodes in a remote storage datacenter (RSD-1), and N_{R2} nodes in a different remote storage datacenter (RSD-2). The total number of storage nodes is given by

$$n = N_{\mathrm{L}} + N_{\mathrm{R1}} + N_{\mathrm{R2}}. \quad (3.1)$$

The original file F is encoded and stored in all of LSD, RSD-1, and RSD-2. A DC connects to any k nodes in the LSD; thus, it can download all the data stored in the k nodes and reconstruct the original file F. Inter-datacenter links (IDLs) between the LSD and RSD-1/RSD-2 are established for data repair when a node in the LSD fails. To regenerate the data stored previously in the failed node, a replacement node in the LSD connects to d helper nodes with N'_{L} helper nodes from the LSD, N'_{R1} helper nodes from RSD-1, and N'_{R2} helper nodes from RSD-2; that is,

$$d = N'_{\mathrm{L}} + N'_{\mathrm{R1}} + N'_{\mathrm{R2}}. \quad (3.2)$$

The replacement node downloads β_{L}, β_{R1}, and β_{R2} symbols from each of the helper nodes in the LSD, RSD-1, and RSD-2, respectively. Further, it is assumed that

$$\beta_{\mathrm{L}} = m_1\beta_{\mathrm{R1}} \quad \text{and} \quad \beta_{\mathrm{R2}} = m_2\beta_{\mathrm{R1}},\ m_1 \geq 1 \quad \text{and} \quad m_2 \geq 1, \quad (3.3)$$

where, m_1 and m_2 are integers. In addition, the total remote repair bandwidth γ_{R} is defined as

$$\gamma_{\mathrm{R}} = \beta_{\mathrm{R1}}N'_{\mathrm{R1}} + \beta_{\mathrm{R2}}N'_{\mathrm{R2}}, \quad (3.4)$$

which represents the total amount of data that a replacement node in the LSD downloads from all remote helper nodes in RSD-1 and RSD-2. Since the data of γ_{R} are transmitted over the Internet, the communication between the LSD and RSD-1/RSD-2 can be susceptible to eavesdropping. Without loss of generality, it is assumed that

$$m_1 = 3 \quad \text{and} \quad m_2 = 2. \quad (3.5)$$

According to the striping of data, we construct codes with $\beta_{\mathrm{R1}} = 1$. Then, we have $\beta_{\mathrm{R2}} = 2$ and $\beta_{\mathrm{L}} = 3$. According to the definition of w (that is, $w = \max\left\{\beta_i, i \in [1,n]\right\}$), we have the case $w = 3$ here.

The EPM framework can be applied to the above three-datacenter scenario. According to the decomposition property of our EPM framework, the ith node can obtain $w = 3$ message vector components $\mathbf{c}^{\mathrm{e}}_{1,i}, \mathbf{c}^{\mathrm{e}}_{2,i}, \mathbf{c}^{\mathrm{e}}_{3,i}$ from the message vector $\mathbf{c}^{\mathrm{e}}_i$.

The reconstruction and regeneration of the three-datacenter scenario are obviously the case of w = 3 under the EPM framework.

References

[1] Y.J. Chen, C.H. Liao, and L.C. Wang, "An Eavesdropping Prevention Problem when Repairing Network Coded Data from Remote Distributed Storage," *Global Commun. Conf.*, Atlanta, GA, USA, Dec. 9–13, 2013, pp. 2711–2716.

[2] A.G. Dimakis et al., "Network Coding for Distributed Storage Systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, Sept. 2010, pp. 4539–4551.

[3] R. Bhagwan et al., "Total Recall: System Support for Automated Availability Management," *Symp. Networked Syst. Des. Implementation*, San Francisco, CA, USA, Mar. 29–31, 2004, pp. 337–350.

[4] F. Dabek et al., "Designing a DHT for Low Latency and High Throughput," *Symp. Networked Syst. Des. Implementation*, San Francisco, CA, USA, Mar. 29–31, 2004, pp. 85–98.

[5] S. Rhea et al., "Pond: The OceanStore Prototype," *USENIX Conf. File Storage Technol.*, San Francisco, CA, USA, Mar. 31–Apr. 2, 2003, pp. 1–14.

[6] H. Weatherspoon and J.D. Kubiatowicz, "Erasure Coding vs. Replication: A Quantitative Comparison," in *Peer-to-Peer Syst.: 1st Int. Workshop, IPTPS 2002 Cambridge, MA, USA, Mar. 7–8, 2002, Revised Papers*, Heidelberg, Germany: Springer, 2002, pp. 328–337.

[7] R. Rodrigues and B. Liskov, "High Availability in DHTs: Erasure Coding vs. Replication," in *Peer-to-Peer Syst. IV: Int. Workshop, Ithaca, NY, USA, Feb. 24–25, 2005, Revised Sel. Papers*, Heidelberg, Germany: Springer, 2005, pp. 226–239.

[8] K.V. Rashmi, N.B. Shah, and P.V. Kumar, "Optimal Exact-Regenerating Codes for the MSR and MBR Points via a Product-Matrix Construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, Aug. 2011, pp. 5227–5239.

[9] K.V. Rashmi et al., "Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage," *Ann. Allerton Conf. Commun., Contr., Comput.*, Monticello, IL, USA, Sept. 30–Oct.

2, 2009, pp. 1243–1249.

[10] O. Olmez and A. Ramamoorthy, "Repairable Replication-Based Storage Systems Using Resolvable Designs," *Ann. Allerton Conf. Commun., Contr., Comput.*, Monticello, IL, USA, Oct. 1–5, 2012, pp. 1174–1181.

[11] Y.S. Han et al., "Update-Efficient Error-Correcting Product-Matrix Codes," *IEEE Trans. Commun.*, vol. 63, no. 6, June 2015, pp. 1925–1938.

[12] T. Ernvall, "Codes between MBR and MSR Points with Exact Repair Property," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, Nov. 2014, pp. 6993–7005.

[13] S. Pawar, S. El Rouayheb, and K. Ramchandran, "On Secure Distributed Data Storage under Repair Dynamics," *IEEE Int. Symp. Inf. Theory*, Austin, TX, USA, July 13, 2010, pp. 2543–2547.

[14] N.B. Shah, K. Rashmi, and P.V. Kumar, "Information-Theoretically Secure Regenerating Codes for Distributed Storage," *IEEE Global Telecommun. Conf.*, Houston, TX, USA, Dec. 5–9, 2011, pp. 1–5.

[15] J. Kubiatowicz et al., "OceanStore: An Architecture for Global-Scale Persistent Storage," *Int. Conf. Architectural Support Programming Languages Operaing Syst.*, Cambridge, MA, USA, Nov. 12–15, 2000, pp. 190–201.

[16] A. Ha, *P2P Startup Space Monkey Raises 2.25 m Led by Google Ventures and Venture 51*, Aol TechCrunch, July 11, 2012, Accessed Feb. 25, 2015. http://techcrunch.com/2012/07/11/space-monkey-seed-round

[17] H. Zhang et al., "A Distributed Multichannel Demand-Adaptive P2P VoD System with Optimized Caching and Neighbor-Selection," *Proc. SPIE*, San Diego, CA, USA, Aug. 22–24, 2011, pp. 81350X-1–81350X-19.

[18] S. Pawar et al., "Codes for a Distributed Caching Based Video-on-Demand System," *Conf. Record Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 6–9, 2011, pp. 1783–1787.

[19] N. Golrezaei, A.G. Dimakis, and A.F. Molisch, "Wireless Device-to-Device Communications with Distributed Caching," *IEEE Int. Symp. Inf. Theory Proc.*, Cambridge, MA, USA, July 1–6, 2012, pp. 2781–2785.

[20] T. Ernvall et al., "Capacity and Security of Heterogeneous Distributed Storage Systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, Dec. 2013, pp. 2701–2709.

[21] Q. Yu, K.W. Shum, and C.W. Sung, "Tradeoff between Storage Cost and Repair Cost in Heterogeneous Distributed Storage Systems," *Trans. Emerg. Telecommun. Technol.*, vol. 26, no. 10, Oct. 2015, pp. 1201–1211.

[22] K.G. Benerjee and M.K. Gupta, "Tradeoff for Heterogeneous Distributed Storage Systems between Storage and Repair Cost." Preprint, submitted Mar. 8, 2015. http://arxiv.org/abs/1503.02276v1

[23] V.T. Van, C. Yuen, and J. Li, "Non-homogeneous Distributed Storage Systems," *Ann. Allerton Conf. Commun., Contr., Comput.*, Monticello, IL, USA, Oct. 1–5, 2012, pp, 1133–1140.

[24] Q. Yu, C.W. Sung, and T.H. Chan, "Irregular Fractional Repetition Code Optimization for Heterogeneous Cloud Storage," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, May 2014, pp. 1048–1060.

[25] D. Leong, A.G. Dimakis, and T. Ho, "Distributed Storage Allocations," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, July 2012, pp. 4733–4752.

[26] V. Ntranos, G. Caire, and A.G. Dimakis, "Allocations for Heterogenous Distributed Storage," *IEEE Int. Symp. Inf. Theory Proc.*, Cambridge, MA, USA, July 1–6, 2012, pp. 2761–2765.

[27] J. Pernas et al., "Non-homogeneous Two-Rack Model for Distributed Storage Systems," *IEEE Int. Symp. Inf. Theory Proc.*, Istanbul, Turkey, July 7–2, 2013, pp. 1237–1241.

[28] N.B. Shah, K.V. Rashmi, and P.V. Kumar, "A Flexible Class of Regenerating Codes for Distributed Storage," *IEEE Int. Symp. Inf. Theory Proc.*, Austin, TX, USA, July 13–18, 2010, pp. 1943–1947.

[29] C. Huang et al., "Erasure Coding in Windows Azure Storage." *USENIX Ann. Tech. Conf.*, Boston, MA, USA, June 13–15, 2012, pp. 82–96.

Jian Xu received her BS degree in electronic information engineering from Shandong University (SDU), Jinan, China, in 2011. She is currently pursuing her PhD in communication and information systems at SDU. Her research interests include distributed storage/regenerating codes and cloud storage system security.

Yewen Cao received his BS degree in communications, MS degree in electronic engineering, and PhD degree in communication and electronic systems from the Chengdu Institute of Information Technology China, School of Electrical Science and Technology, Peking University, Beijing, China, in 1986, 1989, and 1995, respectively. Since 1999, he has been a professor of communications at Shandong University, Jinan, China. He was a research fellow at the National University of Singapore (Sept. 2000 to Aug. 2002), and a post-doctoral research fellow at both the University of Bradford, UK (Sept. 2002 to Sept. 2003), and the University of Glamorgan, Pontypridd, UK (Oct. 2003 to Sept. 2005). His research interests include communication theory (modulations and coding), techniques in 3G, 4G, and 5G mobile communication systems, and mobile IP network computing. He has been either an author or a co-author of over 70 papers in academic journals (international or Chinese) and high-profile international conferences held by the IEEE organization. He is the owner of over 15 patents.

Deqiang Wang received his BS degree in radio technology and his MS degree in signal processing from Shandong University (SDU), Jinan, China, in 1990 and 1995, respectively. He then went on to receive his PhD degree in communication and information systems from Beijing University of Posts and Telecommunications, China, in 2005. Since 1995, he has been with the faculty of the School of Information Science and Engineering, SDU, where he is currently a full professor. His research interests include ultra-wideband communications, multicarrier communications, and adaptive signal processing for wireless communications.

Changlei Wu received his BS degree in electronic information engineering from Southeast University, Nanjing, China, in 1998 and his MS degree in communication and information systems from the Beijing Institute of Technology, China, in 2005. He is currently a PhD student with the School of Information Science and Engineering, Shandong University, Jinan, China. From 1998 to 2001, he worked for Shandong CVIC software engineering Co. Ltd., Jinan, China. Since 2005, he has been with the School of Electrical Engineering and Automation, Qilu University of Technology, Jinan, China. His research interests include wireless communications, network information theory, and interference networks.

Guang Yang received her MS degree in signal and information processing from Shandong University (SDU) of Science and Technology, Jinan, China, in 2013. She is currently pursuing her PhD degree in communication and information systems at SDU, Jinan, China. Her current research interests include non-cooperative and cooperative game theory–based resource allocation interference management in small-cell networks.