Generalized Rack-aware Regenerating Codes for Jointly Optimal Node and Rack Repairs

Hanxu Hou[†‡] and Patrick P. C. Lee[‡]
[†] School of Electrical Engineering & Intelligentization, Dongguan University of Technology
[‡] Department of Computer Science and Engineering, The Chinese University of Hong Kong

***Abstract*— In data centers, storage nodes are organized in racks and the cross-rack communication bandwidth is often much lower than the intra-rack communication bandwidth. Two common failures in data centers are single-node failures and single-rack failures. In this paper, we study the problem of minimizing the cross-rack repair bandwidth in both repairing single-node failures and repairing single-rack failures. We characterize, given that the minimum cross-rack repair bandwidth for single-node failures is achieved, the optimal trade-off between storage and cross-rack repair bandwidth for single-rack failures. We further propose a general family of storage codes, *Generalized Rack-aware Regenerating Codes (GRRC)*, that achieve the optimal trade-off. We obtain two extreme points of GRRC, namely the *minimum storage generalized rack-aware regeneration* (MSGRR) point and the *minimum bandwidth generalized rack-aware regeneration* (MBGRR) point. We show that MSGRR codes have strictly less cross-rack repair bandwidth for single-rack failures than the related minimum storage multi-node repair codes for most parameters. We also show that MBGRR codes have less cross-rack repair bandwidth for single-rack failures than the minimum bandwidth multi-node repair codes for all our evaluated parameters.**

***Index Terms*—Regenerating codes, cross-rack repair bandwidth, single-node failures, single-rack failures.**

I. Introduction

Maximum distance separable (MDS) codes are a class of erasure codes that are widely adopted in distributed storage systems to achieve data reliability with the minimum storage redundancy. An (n, k) MDS code encodes a data file of $k\alpha$ *data symbols* (i.e., the units for erasure coding operations) to obtain $n\alpha$ *coded symbols* over a finite field, where $k < n$ and $\alpha \geq 1$ (α is called the *sub-packetization* level). The $n\alpha$ coded symbols are distributed across n different nodes, each of which stores α symbols. MDS codes satisfy the *reconstruction property*, such that any k out of n nodes can retrieve the $k\alpha$ data symbols.

Modern distributed storage systems often organize nodes in racks, and the cross-rack communication bandwidth is much lower than the intra-rack communication bandwidth [1]. When a node fails, it is critical to repair the failed symbols with the *cross-rack repair bandwidth* (i.e., the total amount of symbols transferred across racks during repair) as small as possible. Several erasure code constructions have been proposed to minimize the cross-rack repair bandwidth [2]–[10]. For example, *Rack-aware regenerating codes* (RRC) [4] minimize the cross-rack repair bandwidth for any single-node failure and provide an optimal trade-off analysis between storage and cross-rack repair bandwidth. In RRC, n nodes are evenly placed in r racks with n/r nodes each, where n is a multiple of r. A data file is encoded into $n\alpha$ symbols that are stored in n nodes with α symbols each, such that the reconstruction property is satisfied. Suppose that a node fails and its α symbols are lost. A new node is identified in the same rack as the failed node for replacement. The new node retrieves all other symbols within the same rack and the encoded symbols from other racks (called *helper racks*) for reconstructing the α lost symbols.

Existing erasure code constructions for optimal cross-rack repair currently focus on optimizing node-level repair. In rack-based data centers, *rack failures*, albeit less prevalent than node failures, can also happen and need to be tolerated [11]. It is critical to design erasure codes that optimize both node-level and rack-level repairs, such that the cross-rack repair bandwidth in both cases can be minimized.

Contributions. In this paper, we propose a general family of erasure codes, called *generalized rack-aware regenerating codes (GRRC)*, that minimize the cross-rack repair bandwidth for both single-node failures and single-rack failures for rack-based data centers. We make the following contributions.

- We derive the optimal trade-off between storage and cross-rack repair bandwidth for any single-rack failure for GRRC, subject to the minimum cross-rack repair bandwidth for any single-node failure. Based on the trade-off between storage and cross-rack repair bandwidth for any single-rack failure of GRRC, we characterize two extreme points, namely the *minimum storage generalized rack-aware regeneration (MSGRR)* point and the *minimum bandwidth generalized rack-aware regeneration (MBGRR)* point.
- We show that MSGRR codes have the same cross-rack repair bandwidth for single-rack failures as the minimum storage multi-node repair codes [12] when kr is a multiple of n, and have strictly less cross-rack repair bandwidth than the minimum storage multi-node repair codes when kr is not a multiple of n. We also show that MBGRR codes have less cross-rack repair bandwidth for single-rack failures than the minimum bandwidth multi-node repair codes [12] for all the evaluated parameters.

The main difference between GRRC and the repair of multi-node failures in [12] is as follows. In the repair of n/r-node

This work was partially supported by the National Key R&D Program of China (No. 2020YFA0712300), the National Natural Science Foundation of China (No. 62071121), Research Grants Council of HKSAR (AoE/P-404/18) and Innovation and Technology Fund (ITS/315/18FX).

failures in [12], a centralized server retrieves encoded symbols directly from some selected surviving nodes via cross-rack transmissions and dispatches the reconstructed symbols to n/r new nodes; there is no further encoding among the encoded symbols. In contrast, in the repair of a single-rack failure (with n/r-node failures) in GRRC, the encoded symbols from the surviving nodes within each helper rack are *re-encoded*, and the re-encoded symbols are transmitted to the new replacement rack via cross-rack transmissions. Our results show that such re-encoding is critical for minimizing the cross-rack repair bandwidth for a single-rack failure.

Related work. Many studies focus on minimizing the number of symbols transferred in repair. There are mainly three directions.

1) Optimal repair for single-node failures. Regenerating codes (RC) [13] are the seminal work on minimizing the *repair bandwidth* (i.e., the total amount of symbols transferred during repair) for a single-node failure. RC operates on the optimal trade-off between storage and repair bandwidth, with two extreme points: the *minimum storage regeneration (MSR)* point and the *minimum bandwidth regeneration (MBR)* point. Exact-repair constructions of RC are investigated in [14]–[23], most of which focus on MSR codes or MBR codes.

2) Optimal repair for multi-node failures. Cooperative regenerating codes [24] repair multi-node failures in a distributed manner. Each of the replacement nodes first downloads encoded symbols from multiple surviving nodes, and recovers the failed symbols of a failed node by downloading some encoded symbols from the other replacement nodes. Rack-aware cooperative regenerating codes [25] repair multi-node failures over the rack-based storage, where the node failures are uniformly distributed among a certain number of racks.

Another direction of repairing multi-node failures is centralized repair [26], in which all the failed symbols are first repaired at a central server before and the regenerated symbols are dispatched to different replacement nodes. It is shown in [27], [28] that ZigZag codes [17], which are MSR codes with minimum repair bandwidth for any single-node failure, also have minimum repair bandwidth for multi-node failures in centralized repair. Ye and Barg [29] present an explicit construction of MDS array codes that provide minimum repair bandwidth for any e-node failures for all $e \leq n-k$. Zorgui and Wang [12] present the optimal trade-off between storage and repair bandwidth for multi-node failures in centralized repair, and also present exact-repair constructions of MDS codes to achieve minimum repair bandwidth for multi-node failures in centralized repair.

3) Optimal cross-rack repair. Several exact-repair constructions for rack-based data centers have been proposed to minimize the cross-rack repair bandwidth for single-node failures [2]–[10], [30]. In addition to RRC [4], previous studies [9], [10] also study the optimal trade-off between storage and cross-repair bandwidth, but focus on the analysis where kr is a multiple of n. Note that existing studies mainly focus on optimizing single-node repair, but do not consider the joint optimal single-node and single-rack repairs as in this work.

II. Optimal Trade-off Between Storage and Cross-Rack Repair Bandwidth

In this section, we describe the system model. We analyze the optimal trade-off between storage and cross-rack repair bandwidth for single-rack failures, subject to the condition that the cross-rack repair bandwidth for single-node failures is minimum. We further propose *Generalized Rack-aware Regenerating Codes* (GRRC) that operate on the optimal trade-off for single-rack failures.

A. System Model

We consider a rack-based data center, in which there are n nodes that are evenly partitioned into r racks with n/r nodes each, where n is assumed to be a multiple of r. A data file of B data symbols is encoded into $n\alpha$ symbols that are stored in n nodes with α symbols each. We label the r racks from 1 to r, and label the n/r nodes in each rack from 1 to n/r. We assume that the intra-rack communication bandwidth is abundant, and all $\alpha n/r$ symbols stored in each rack can be used to repair a single-node failure or a single-rack failure. We select a node in each rack to be a *relayer node* that can retrieve encoded symbols from the surviving nodes in the same rack during repair. Without loss of generality, we choose node 1 in each rack as the relayer node. In this work, we optimize both single-node repair and single-rack repair. We elaborate their repair properties and define the notations as follows.

Repair property for single-node failures. We follow the repair model of RRC [4] for single-node failures. When a node fails, we select a new node in the same rack as the failed node for replacement. The new node arbitrarily selects $d_{\rm node}$ *helper racks*, where $\frac{kr}{n} \leq d_{\rm node} \leq r-1$, and downloads $\beta_{\rm node}$ encoded symbols from the relayer node in each of the $d_{\rm node}$ helper racks, in which the relayer node computes the $\beta_{\rm node}$ encoded symbols based on the $\alpha n/r$ symbols within the same rack. The new node can recover the α lost symbols with the $\alpha(n/r-1)$ symbols from other $n/r-1$ nodes in the same rack, as well as the $d_{\rm node}\beta_{\rm node}$ downloaded symbols from the helper racks. Thus, the cross-rack repair bandwidth for single-node failures is $d_{\rm node}\beta_{\rm node}$. The optimal trade-off [4] between storage and cross-rack repair bandwidth for single-node failures is:

$$k\alpha + \sum_{\ell=1}^{m} \min\{(d_{\rm node}-\ell+1)\beta_{\rm node}-\alpha, 0\} \geq B, \qquad (1)$$

where $m = \lfloor \frac{kr}{n} \rfloor$. There are two extreme points in the optimal trade-off in Eq. (1), namely the *minimum storage rack-aware regeneration (MSRR)* point and the *minimum bandwidth rack-aware regeneration (MBRR)* point, corresponding to the minimum storage and the minimum cross-rack repair bandwidth, respectively. The MSRR point corresponds to

$$(\alpha, \beta_{\rm node}) = (\frac{B}{k}, \frac{B}{k(d_{\rm node}-m+1)}), \qquad (2)$$

and the MBRR point corresponds to

$$(\alpha, \beta_{\rm node}) = (\frac{2Bd_{\rm node}}{2kd_{\rm node}-m(m-1)}, \frac{2B}{2kd_{\rm node}-m(m-1)}). \qquad (3)$$

Repair property for single-rack failures. When a rack fails, we create a new rack that contains n/r new nodes, including one relayer node and $n/r-1$ non-relayer nodes. The relayer node in the new rack arbitrarily selects $d_{\rm rack}$ helper racks, where $\frac{kr}{n} \leq d_{\rm rack} \leq r-1$, and downloads $\beta_{\rm rack}$ encoded symbols from each of $d_{\rm rack}$ racks. The relayer node in the new rack can repair the $\alpha n/r$ lost symbols with the $d_{\rm rack}\beta_{\rm rack}$ downloaded symbols and sends $\alpha(n/r-1)$ recovered symbols to the other $n/r-1$ new nodes in the new rack. The *cross-rack repair bandwidth* $\gamma_{\rm rack}$ for a single-rack failure is $\gamma_{\rm rack} = d_{\rm rack}\beta_{\rm rack}$.

Goal. Our goal is to design a new erasure code construction that minimizes the cross-rack repair bandwidth for both single-node failures and single-rack failures, subject to the repair properties for both single-node failures and single-rack failures as well as the *reconstruction property* (i.e., any k out of n nodes can reconstruct the data file).

Since node failures are much more prevalent than rack failures [11], we require that the cross-rack repair bandwidth for single-node failures be minimum as specified in Eq. (1). Then our analysis goal is to characterize the optimal trade-off between storage and cross-rack repair bandwidth for single-rack failures, subject to the minimum cross-rack repair bandwidth for single-node failures.

B. Information Flow Graphs

We draw the information flow graphs for repairing a single-rack failure, so as to derive the trade-off between storage and cross-rack repair bandwidth for single-rack failures.

Given the system parameters n, k, r, α, $d_{\rm rack}$, and $\beta_{\rm rack}$, we draw a directed acyclic graph, denoted by $G(n,k,r,\alpha,d_{\rm rack},\beta_{\rm rack})$ (or G in short). Fig. 1 shows an example of G with $(n,k,r,d_{\rm rack}) = (12,6,4,3)$. G contains a source node S that corresponds to the data file of size B, and a data collector T. Each node i in rack ℓ is represented by a pair of an input node $\mathrm{In}_{\ell,i}$ and an output node $\mathrm{Out}_{\ell,i}$, where $\ell = 1,2,\ldots,r$ and $i = 1,2,\ldots,n/r$. We also draw directed edges as follows:

- For $\ell = 1,2,\ldots,r$ and $i = 1,2,\ldots,n/r$, we draw a directed edge from each input node $\mathrm{In}_{\ell,i}$ to its corresponding output node $\mathrm{Out}_{\ell,i}$ with capacity α, representing the storage size of α units in node i in rack ℓ.
- For $\ell = 1,2,\ldots,r$ and $i = 1,2,\ldots,n/r$, we draw a directed edge from S to each input node $\mathrm{In}_{\ell,i}$ with infinite capacity.
- For $\ell = 1,2,\ldots,r$ and $i = 2,3,\ldots,n/r$, we draw a directed edge from each output node $\mathrm{Out}_{\ell,i}$ to the output node of the relayer node, $\mathrm{Out}_{\ell,1}$, in rack ℓ with infinite capacity, representing that the relayer node in rack ℓ can access all the symbols in rack ℓ.

Suppose that rack f fails, where $f \in \{1,2,\ldots,r\}$. We create a new rack to replace the failed rack f, and put n/r pairs of input node $\mathrm{In}'_{f,i}$ and output node $\mathrm{Out}'_{f,i}$ in G to replace n/r nodes in rack f. To model the local encoding process in the new rack f, we add a virtual node, denoted by Virt_f. We draw the directed edges as follows:

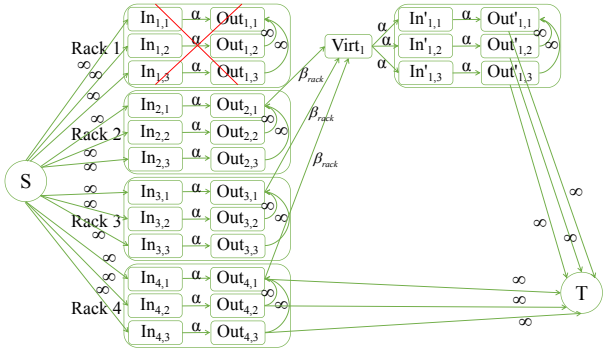

Fig. 1: Information flow graph of G with $(n,k,r,d_{\rm rack}) = (12,6,4,3)$.

- For $i = 1,2,\ldots,n/r$, we draw a directed edge from Virt_f to $\mathrm{In}'_{f,i}$ with capacity α.
- For $i = 1,2,\ldots,n/r$, we draw a directed edge from $\mathrm{In}'_{f,i}$ to $\mathrm{Out}'_{f,i}$ with capacity α.
- For $i = 2,3,\ldots,n/r$, we draw a directed edge from $\mathrm{Out}'_{f,i}$ to $\mathrm{Out}'_{f,1}$ with infinite capacity.
- Suppose that we select $d_{\rm rack}$ helper racks $h_1,h_2,\ldots,h_{d_{\rm rack}} \in \{1,2,\ldots,r\} \setminus \{f\}$. For $\ell = 1,2,\ldots,d_{\rm rack}$, we draw a directed edge from $\mathrm{Out}_{h_\ell,1}$ to Virt_f with capacity $\beta_{\rm rack}$.
- To represent the reconstruction property, we draw k edges from any k output nodes to T with infinite capacity.

Note that there are $\binom{n}{k}$ combinations of connecting k output nodes to T, and we can represent the set of all possible graphs of G as $\mathcal{G}(n,k,r,d_{\rm rack},\alpha,\beta_{\rm rack})$ (or $\mathcal{G}$ in short). Given G and T, we define an (S,T)-cut as a subset of the edges in G, such that S and T are disconnected if the edges in the subset are removed from G. We define the capacity of (S,T)-cut as the sum of the capacities of the edges in the cut. Let $\mathrm{mincut}(G)$ be an (S,T)-cut of a given G with the smallest capacity, and $\min_{\mathcal{G}} \mathrm{mincut}(G)$ be the minimum capacity. We characterize the minimum cut in the following theorem.

Theorem 1. *Given n, k, r, $d_{\rm rack}$, α, $\beta_{\rm rack}$, and B, if an erasure code satisfies the repair property for single-rack failures and the reconstruction property, the following inequality holds:*

$$k\alpha + \sum_{\ell=1}^{m} \min\{(d_{\rm rack}-\ell+1)\beta_{\rm rack} - \alpha n/r, 0\} \geq B. \quad (4)$$

Proof. We present our proof in three steps.

Step 1. In calculating the minimum cut, we make the following claim. If a relayer node in a rack is connected to the data collector T and not all the other $n/r-1$ nodes in the rack are connected to T, then the capacity of (S,T)-cut is not the smallest.

To prove the claim, suppose that a relayer node is connected to T. Then the relayer node can contribute $\min\{(n/r-1)\alpha + d_{\rm rack}\beta_{\rm rack}, \alpha n/r\}$ symbols to the cut if the relayer node is a new node and $\alpha n/r$ symbols to the cut if the relayer node is not a new node. The other $n/r-1$ nodes in the same rack of the relayer node have no contribution to the cut whether they are connected to T or not. Therefore, if the relayer node is

connected to T, all the other $n/r-1$ nodes in the same rack should be connected to T, in order to minimize the capacity of the cut.

Step 2. We next show that there exists an information flow graph G, such that $\mathrm{mincut}(G)$ is equal to the left-hand side of Eq. (4).

Suppose that in G, racks $1,2,\ldots,m$ fail in this order. For $\ell=1,2,\ldots,m$, we draw a directed edge from each of the output nodes $\mathrm{Out}'_{1,1}$, $\mathrm{Out}'_{2,1}$, ..., $\mathrm{Out}'_{\ell-1,1}$, $\mathrm{Out}_{\ell+1,1}$, $\mathrm{Out}_{\ell+2,1}$, ..., $\mathrm{Out}_{d_{\mathrm{rack}}+1,1}$ of the relayer nodes in d_{rack} helper racks $1,2,\ldots,\ell-1,\ell+1,\ell+2,\ldots,d_{\mathrm{rack}}+1$, respectively, to the virtual node Virt_ℓ with capacity β_{rack}. For $i=1,2,\ldots,n/r$, we also draw a directed edge from Virt_ℓ to each of n/r input nodes $\mathrm{In}'_{\ell,i}$ with capacity α. Furthermore, we draw a directed edge from the input node $\mathrm{In}'_{\ell,i}$ to the output node $\mathrm{Out}'_{\ell,i}$ with capacity α.

Consider that the data collector T connects to all mn/r nodes in m racks $1,2,\ldots,m$ and $k-mn/r$ nodes (except the relayer node) in rack $m+1$. For $\ell=1,2,\ldots,m$, the n/r nodes in rack ℓ contribute $\min\{(n/r-1)\alpha+(d_{\mathrm{rack}}-\ell+1)\beta_{\mathrm{rack}},\alpha n/r\}$ symbols to the cut. Thus, the min-cut of the graph G is the left-hand side of Eq. (4).

Step 3. Finally, we show that the min-cut of any G in the set $\mathcal{G}$ is no less than the left-hand side of Eq. (4).

Given G, let the k output nodes connected to T be $\{\mathrm{Out}_{\ell,i}:(\ell,i)\in\mathbb{I}\}$, where the cardinality of $\mathbb{I}$ is k. Consider the first subset with n/r output nodes of $\mathbb{I}$. If the first subset only contains one relayer node, then the subset with n/r output nodes can contribute $\min\{(n/r-1)\alpha+d_{\mathrm{rack}}\beta_{\mathrm{rack}},\alpha n/r\}$ symbols to the cut by the proof of Step 1. Otherwise, if the first subset contains more than one relayer node or contains no relayer node, then the contributed symbols to the cut by the first subset are no less than $\min\{(n/r-1)\alpha+d_{\mathrm{rack}}\beta_{\mathrm{rack}},\alpha n/r\}$. Therefore, the first subset contributes at least $\min\{(n/r-1)\alpha+d_{\mathrm{rack}}\beta_{\mathrm{rack}},\alpha n/r\}$ symbols to the cut. Similarly, we can show that the ℓ-th subset with n/r output nodes can contribute at least $\min\{(n/r-1)\alpha+(d_{\mathrm{rack}}-\ell+1)\beta_{\mathrm{rack}},\alpha n/r\}$ symbols to the cut, where $\ell=2,3,\ldots,m$. We have that the min-cut of any G is the left-hand side of Eq. (4). □

The erasure codes that achieve the minimum cross-rack repair bandwidth for single-node failures while satisfying the equality in Eq. (4) are called the *Generalized Rack-aware Regenerating Codes (GRRC)*. Note that when $r=n$ (i.e., each rack contains one node), the trade-off curve in Eq. (4) reduces to the trade-off curve of regenerating codes [13].

We can obtain two extreme points in the optimal trade-off in Eq. (4), namely the *minimum storage generalized rack-aware regeneration (MSGRR)* point and the *minimum bandwidth generalized rack-aware regeneration (MBGRR)* point, which correspond to the minimum storage and the minimum cross-rack repair bandwidth for single-rack failures, respectively, given that the MSRR point in Eq. (2) and the MBRR point Eq. (3) are achieved, respectively. From Eq. (1) and Eq. (4), the MSGRR point is achieved when

$$\alpha=\frac{B}{k},\beta_{\mathrm{rack}}=\frac{B}{k(d_{\mathrm{rack}}-m+1)}\cdot\frac{n}{r}. \tag{5}$$

If $d_{\mathrm{node}}=d_{\mathrm{rack}}$, then $\beta_{\mathrm{rack}}=\beta_{\mathrm{node}}\cdot\frac{n}{r}$ by Eq. (2) and Eq. (5). In the following, we consider the case of $d_{\mathrm{node}}=d_{\mathrm{rack}}$. Note that the minimum storage points of the optimal trade-off in Eq. (4) and in Eq. (1) are achieved at the same value of α. However, the minimum repair bandwidth points of the optimal trade-off in Eq. (4) and in Eq. (1) are achieved at different values of α. Recall that the MBGRR code is obtained by first minimizing β_{node} and α in Eq. (1) and then minimizing β_{rack} in Eq. (4). By minimizing β_{node} and α in Eq. (1), we have α and β_{node} in Eq. (3). By substituting α in Eq. (3) into Eq. (4), we can obtain the MBGRR point in the next theorem.

Theorem 2. *The MBGRR point is achieved with* $\alpha=\frac{2Bd_{\mathrm{node}}}{2kd_{\mathrm{node}}-m(m-1)}$ *and*

$$\beta_{\mathrm{rack}}=\frac{\frac{-\alpha m(m-1)}{d_{\mathrm{rack}}}+2\tau\alpha n/r}{\tau(2d_{\mathrm{rack}}-\tau+1)}, \tag{6}$$

where τ *is a positive integer with* $\tau=1,2,\ldots,m-1$ *that satisfies*

$$\frac{n/r}{d_{\mathrm{rack}}-\tau}>\frac{\frac{-m(m-1)}{d_{\mathrm{rack}}}+2\tau n/r}{\tau(2d_{\mathrm{rack}}-\tau+1)}\geq\frac{n/r}{d_{\mathrm{rack}}-\tau+1}. \tag{7}$$

Proof. In MBGRR point, we have $\alpha=\frac{2Bd_{\mathrm{node}}}{2kd_{\mathrm{node}}-m(m-1)}$ according to Eq. (3). By letting Eq. (4) with equality, we obtain

$$k\alpha+\sum_{\ell=1}^{m}\min\{(d_{\mathrm{rack}}-\ell+1)\beta_{\mathrm{rack}}-\alpha n/r,0\}=B. \tag{8}$$

Recall that the m terms of the summation in the left-hand side of Eq. (8) are $\sum_{\ell=1}^{m}\min\{(d_{\mathrm{rack}}-\ell+1)\beta_{\mathrm{rack}}-\alpha n/r,0\}$. There exists an integer τ that ranges from 1 to $m-1$ such that

$$(d-\tau+1)\beta_{\mathrm{rack}}\geq\alpha n/r>(d-\tau)\beta_{\mathrm{rack}}.$$

Then we have

$$\sum_{\ell=1}^{m}\min\{(d_{\mathrm{rack}}-\ell+1)\beta_{\mathrm{rack}}-\alpha\frac{n}{r},0\}=\frac{\tau(2d-\tau+1)\beta_{\mathrm{rack}}}{2},$$

and from Eq. (8), we further obtain Eq. (6) and Eq. (7). □

Remarks. We remark that our MSGRR point is contained in the MSRR point, since the MSGRR point is derived by first achieving the MSRR point and then achieving the minimum cross-rack repair bandwidth for single-rack failures. Similarly, our MBGRR point is contained in the MBRR point.

A single-rack failure can be viewed as n/r-node failures in our model. The trade-off between storage and the repair bandwidth for multi-node failures for functional repair has been given by cooperative regenerating codes [24] with distributed repair as well as by [12] with centralized repair. Since the centralized repair [12] incurs less repair bandwidth than cooperative regenerating codes [24], we choose the centralized repair [12] as the baseline for our comparison.

Specifically, the codes for the two extreme points in the optimal tradeoff in [12] are called minimum storage multi-node repair (MSMR) codes and minimum bandwidth multi-node repair (MBMR) codes. If we directly employ the MSMR codes in a rack-based data center, we can obtain the storage and cross-rack repair bandwidth for a single-rack failure (n/r-node failures) for MSMR codes as follows:

$$(\alpha_{\mathrm{MSMR}}, \gamma_{\mathrm{MSMR}}) = (\frac{B}{k}, \frac{B}{k} \cdot \frac{\frac{n}{r}d}{d-k+\frac{n}{r}}), \tag{9}$$

where $d = d_{\mathrm{rack}}\frac{n}{r}$. Similarly, the storage and cross-rack repair bandwidth for a single-rack failure (n/r-node failures) for MBMR codes are:

$$(\alpha_{\mathrm{MBMR}}, \gamma_{\mathrm{MBMR}}) = (\frac{\gamma_{\mathrm{MBMR}}}{n/r}, \frac{Bd}{dm - \frac{n}{r} \cdot \frac{m(m-1)}{2}}), \tag{10}$$

when $\frac{kr}{n}$ is an integer,

$$\begin{aligned}&(\alpha_{\mathrm{MBMR}}, \gamma_{\mathrm{MBMR}})\\ =&(\gamma_{\mathrm{MBMR}}\frac{d+m(n-k-\frac{n}{r})}{d(k-m\frac{n}{r})}, \frac{Bd}{d(m+1)-\frac{nm(m+1)}{2r}}),\end{aligned} \tag{11}$$

when $\frac{kr}{n}$ is not an integer, where $d = d_{\mathrm{rack}}\frac{n}{r}$.

III. Comparison

In this section, we evaluate the cross-rack repair bandwidth for MSGRR codes, MBGRR codes and the related centralized regenerating codes [12] at two extreme points. For MSGRR codes and MBGRR codes, let $d_{\mathrm{node}} = d_{\mathrm{rack}}$.

We first consider the comparison for MSGRR codes and MSMR codes, which is summarized in the following theorem.

Theorem 3. *If $\frac{kr}{n}$ is an integer, then the cross-rack repair bandwidth for a single-rack failure of MSGRR codes is the same as that of MSMR codes. If $\frac{kr}{n}$ is not an integer, then MSGRR codes have strictly less cross-rack repair bandwidth for a single-rack failure than MSMR codes.*

Proof. Recall that the cross-rack repair bandwidth for a single-rack failure of MSGRR codes is given in Eq. (5) and the cross-rack repair bandwidth for a single-rack failure of MSMR codes [12] is in Eq. (9). The difference of the cross-rack repair bandwidth for single-rack failures for MSGRR codes and MSMR codes is

$$\begin{aligned}&\gamma_{\mathrm{rack}} - \gamma_{\mathrm{MSMR}}\\ =&\frac{B}{k(d_{\mathrm{rack}}-m+1)} \cdot \frac{n}{r} \cdot d_{\mathrm{rack}} - \frac{B}{k} \cdot \frac{\frac{n}{r}d_{\mathrm{rack}}\frac{n}{r}}{d_{\mathrm{rack}}\frac{n}{r}-k+\frac{n}{r}}\\ =&\frac{Bnd_{\mathrm{rack}}}{kr} \cdot (\frac{1}{d_{\mathrm{rack}}-m+1} - \frac{1}{d_{\mathrm{rack}}-\frac{kr}{n}+1}).\end{aligned}$$

When $\frac{kr}{n}$ is an integer, we have $m = \frac{kr}{n}$ and MSGRR codes have the same cross-rack repair bandwidth for single-rack failures as MSMR codes. When $\frac{kr}{n}$ is not an integer, we have $m < \frac{kr}{n}$ and MSGRR codes have strictly less cross-rack repair bandwidth for single-rack failures than MSMR codes. □

Table I shows the comparison of MSGRR codes and MSMR codes in terms of cross-rack repair bandwidth. The results in

TABLE I: Cross-rack repair bandwidth for single-rack failures (n/r-node failures) of MSGRR codes and MSMR codes.

$(n,k,r,d_{\mathrm{rack}})$	MSGRR	MSMR	Improvement
$(20,9,5,4)$	$\frac{16}{27}$	$\frac{64}{99}$	8.3%
$(20,9,5,3)$	$\frac{2}{3}$	$\frac{16}{21}$	12.5%
$(20,10,5,4)$	$\frac{8}{15}$	$\frac{16}{25}$	16.7%
$(20,10,5,3)$	$\frac{3}{5}$	$\frac{4}{5}$	25.0%
$(20,11,5,4)$	$\frac{16}{33}$	$\frac{64}{99}$	25.0%
$(20,11,5,3)$	$\frac{6}{11}$	$\frac{48}{55}$	37.5%
$(20,13,5,4)$	$\frac{8}{13}$	$\frac{64}{91}$	12.5%
$(20,14,5,4)$	$\frac{4}{7}$	$\frac{16}{21}$	25.0%
$(20,15,5,4)$	$\frac{8}{15}$	$\frac{64}{75}$	37.5%
$(18,10,6,5)$	$\frac{1}{2}$	$\frac{9}{16}$	11.1%
$(18,10,6,4)$	$\frac{3}{5}$	$\frac{18}{25}$	16.7%
$(18,11,6,5)$	$\frac{5}{11}$	$\frac{45}{77}$	22.2%
$(18,11,6,4)$	$\frac{6}{11}$	$\frac{9}{11}$	33.3%
$(18,13,6,5)$	$\frac{15}{26}$	$\frac{9}{13}$	16.7%
$(18,14,6,5)$	$\frac{15}{28}$	$\frac{45}{56}$	33.3%

TABLE II: Cross-rack repair bandwidth for single-rack failures (n/r-node failures) of MBGRR codes and MBMR codes.

$(n,k,r,d_{\mathrm{rack}})$	MBGRR	MBMR	Improvement
$(20,9,5,4)$	$\frac{3}{7}$	$\frac{4}{9}$	3.6%
$(20,10,5,4)$	$\frac{5}{13}$	$\frac{4}{9}$	13.5%
$(20,11,5,4)$	$\frac{15}{43}$	$\frac{4}{9}$	21.5%
$(20,12,5,4)$	$\frac{116}{315}$	$\frac{4}{9}$	17.1%
$(20,13,5,4)$	$\frac{116}{343}$	$\frac{2}{5}$	15.5%
$(20,14,5,4)$	$\frac{116}{371}$	$\frac{2}{5}$	21.8%
$(20,15,5,4)$	$\frac{116}{399}$	$\frac{2}{5}$	27.3%

Table I show that the cross-rack repair bandwidth for single-rack failures of MSGRR codes is less than that of MSMR codes, when $\frac{kr}{n}$ is not an integer.

The cross-rack repair bandwidth for a single-rack failure of MBGRR codes is given in Theorem 2 and the cross-rack repair bandwidth for a single-rack failure of MBMR codes is given in Eq. (10) and Eq. (11). Table II shows the comparison for MBGRR codes and MBMR codes in terms of cross-rack repair bandwidth for a single-rack failure for some parameters, which demonstrates that the cross-rack repair bandwidth for a single-rack failure of MBGRR codes is less than that of MBMR codes for all our evaluated parameters.

IV. Conclusion

In this paper, we propose GRRC that can achieve the minimum cross-rack repair bandwidth for single-rack failures under the condition that the cross-rack repair bandwidth for single-node failures is minimum. We derive two extreme optimal points of GRRC, namely MSGRR and MBGRR points. We show that MSGRR codes have strictly less cross-rack repair bandwidth for single-rack failures than that of MSMR codes for most of the parameters. We also show that the cross-rack repair bandwidth for single-rack failures of MBGRR codes is less than that of MBMR codes for all the evaluated parameters. Exact-repair construction of the two extreme optimal points is one of our future work.

References

[1] M. Isard, V. Prabhakaran, J. Currey, U. Wieder, K. Talwar, and A. Goldberg, "Quincy: Fair scheduling for distributed computing clusters," in *Proc. of ACM SOSP*, 2009.

[2] Y. Hu, P. P. C. Lee, and X. Zhang, "Double Regenerating Codes for Hierarchical Data Centers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2016, pp. 245–249.

[3] Y. Hu, X. Li, M. Zhang, P. P. C. Lee, X. Zhang, P. Zhou, and D. Feng, "Optimal Repair Layering for Erasure-Coded Data Centers: From Theory to Practice," *ACM Transactions on Storage*, vol. 13, no. 4, pp. 33–56, 2017.

[4] H. Hou, P. P. Lee, K. W. Shum, and Y. Hu, "Rack-Aware Regenerating Codes for Data Centers," *IEEE Trans. Information Theory*, vol. 65, no. 8, pp. 4730—-4745, Aug. 2019.

[5] Z. Chen and A. Barg, "Explicit Constructions of MSR Codes for Clustered Distributed Storage: The Rack-Aware Storage Model," *IEEE Trans. Information Theory*, vol. 66, no. 2, pp. 886–899, Feb. 2020.

[6] L. Jin, G. Luo, and C. Xing, "Optimal Repairing Schemes for Reed-Solomon Codes with Alphabet Sizes Linear in Lengths under the Rack-Aware Model," *arXiv preprint arXiv:1911.08016*, 2019.

[7] H. Hou, P. P. C. Lee, and Y. S. Han, "Minimum Storage Rack-Aware Regenerating Codes with Exact Repair and Small Sub-Packetization," in *Proc. IEEE Int. Symp. Inf. Theory*, 2020.

[8] J. Pernas, C. Yuen, B. Gastón, and J. Pujol, "Non-Homogeneous Two-Rack Model for Distributed Storage Systems," in *Proc. IEEE Int. Symp. Inf. Theory*, 2013, pp. 1237–1241.

[9] J.-y. Sohn, B. Choi, S. W. Yoon, and J. Moon, "Capacity of Clustered Distributed Storage," *IEEE Trans. Information Theory*, vol. 65, no. 1, pp. 81–107, 2019.

[10] N. Prakash, V. Abdrashitov, and M. Médard, "The Storage versus Repair-Bandwidth Trade-off for Clustered Storage Systems," *IEEE Trans. Information Theory*, vol. 64, no. 8, pp. 5783–5805, August 2018.

[11] S. Muralidhar, W. Lloyd, S. Roy, C. Hill, E. Lin, W. Liu, S. Pan, S. Shankar, V. Sivakumar, L. Tang *et al.*, "f4: Facebook's Warm Blob Storage System," in *Proc. of USENIX OSDI*, 2014.

[12] M. Zorgui and Z. Wang, "Centralized Multi-Node Repair Regenerating Codes," *IEEE Trans. Information Theory*, vol. 65, no. 7, pp. 4180–4206, 2019.

[13] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *IEEE Trans. Information Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

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

[15] N. B. Shah, K. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference Alignment in Regenerating Codes for Distributed Storage: Necessity and Code Constructions," *IEEE Trans. Information Theory*, vol. 58, no. 4, pp. 2134–2158, 2012.

[16] C. Tian, "Characterizing the Rate Region of the (4,3,3) Exact-Repair Regenerating Codes," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 967–975, 2014.

[17] I. Tamo, Z. Wang, and J. Bruck, "Zigzag Codes: MDS Array Codes with Optimal Rebuilding," *IEEE Trans. Information Theory*, vol. 59, no. 3, pp. 1597–1616, May 2013.

[18] H. Hou, K. W. Shum, M. Chen, and H. Li, "BASIC Codes: Low-Complexity Regenerating Codes for Distributed Storage Systems," *IEEE Trans. Information Theory*, vol. 62, no. 6, pp. 3053–3069, 2016.

[19] M. Ye and A. Barg, "Explicit Constructions of Optimal-Access MDS Codes with Nearly Optimal Sub-Packetization," *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6307–6317, 2017.

[20] J. Li, X. Tang, and C. Tian, "A Generic Transformation to Enable Optimal Repair in MDS codes for Distributed Storage Systems," *IEEE Trans. Information Theory*, vol. 64, no. 9, pp. 6257–6267, 2018.

[21] M. Vajha, V. Ramkumar, B. P. Puranik, G. Kini, E. Lobo, B. Sasidharan, P. V. Kumar, A. M. Barg, M. Ye, and S. Narayanamurthy, "Clay Codes: Moulding MDS Codes to Yield an MSR Code," in *Proc. of USENIX FAST*, pp. 139–154.

[22] H. Hou and P. P. Lee, "Binary MDS Array Codes with Optimal Repair," *IEEE Trans. Information Theory*, vol. 66, no. 3, pp. 1405—-1422, Mar. 2020.

[23] H. Hou, P. P. Lee, and Y. S. Han, "Multi-Layer Transformed MDS Codes with Optimal Repair Access and Low Sub-Packetization," *arXiv preprint arXiv:1907.08938*, 2019.

[24] K. W. Shum and Y. Hu, "Cooperative Regenerating Codes," *IEEE Trans. Information Theory*, vol. 59, no. 11, pp. 7229–7258, 2013.

[25] S. Gupta and V. Lalitha, "Rack-Aware Cooperative Regenerating Codes," in *International Symposium on Information Theory and Its Applications (ISITA) 2020*, 2020, pp. 264–268.

[26] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic Interference Alignment for Optimal Repair of MDS Codes in Distributed Storage," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 2974–2987, 2013.

[27] Z. Wang, I. Tamo, and J. Bruck, "Optimal rebuilding of multiple erasures in mds codes," *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 1084–1101, 2017.

[28] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Centralized Repair of Multiple Node Failures With Applications to Communication Efficient Secret Sharing," *IEEE Transactions on Information Theory*, vol. 64, no. 12, pp. 7529–7550, 2018.

[29] M. Ye and A. Barg, "Explicit Constructions of High-Rate MDS Array Codes with Optimal Repair Bandwidth," *IEEE Trans. Information Theory*, vol. 63, no. 4, pp. 2001–2014, 2017.

[30] J. Wang, Y. Luo, and K. W. Shum, "Storage and Repair Bandwidth Tradeoff for Heterogeneous Cluster Distributed Storage Systems," *Science China Information Sciences*, vol. 63, no. 2, pp. 1–5, 2020.