Double Regenerating Codes for Hierarchical Data Centers

Yuchong Hu[1], Patrick P. C. Lee[2], and Xiaoyang Zhang[1]
[1]Huazhong University of Science and Technology, [2]The Chinese University of Hong Kong
Email: *yuchonghu@hust.edu.cn, pclee@cse.cuhk.edu.hk, zhangxiaoyang1993@gmail.com*

***Abstract*—Data centers increasingly adopt erasure coding to ensure fault-tolerant storage with low redundancy, yet the hierarchical nature of data centers incurs substantial oversubscribed cross-rack bandwidth in failure repair. We present Double Regenerating Codes (DRC), whose idea is to perform two-stage regeneration, so as to minimize the cross-rack repair bandwidth for a single-node repair with the minimum storage redundancy. We prove the existence of a DRC construction, and show via quantitative comparisons that DRC significantly reduces the cross-rack repair bandwidth of state-of-the-art minimum storage regenerating codes.**

I. Introduction

Enterprises deploy data centers for large-scale storage, yet a critical deployment challenge is to tolerate data loss against failures. Erasure coding enables high fault-tolerant storage with much less redundancy than traditional replication. Given the ever-increasing data growth, erasure coding is widely adopted for data center storage (e.g., [6], [8], [11], [16]). For example, Facebook reportedly reduces the storage redundancy from $3\times$ in triple replication to $1.4\times$ via erasure coding [11], [16], thereby saving petabytes of storage.

An erasure code is often constructed by two configurable parameters n and k (where $k < n$). Given the original data of size M, an (n, k) erasure code divides the original data into k fragments of size M/k each, and transforms them into n encoded fragments of the same size. Each encoded fragment is stored in a distinct node (or server). This paper focuses on the erasure code constructions that satisfy the *maximum distance separable (MDS) property*, meaning that any k out of n encoded fragments suffice to reconstruct the original data, while the storage redundancy is the minimum.

Erasure coding trades performance for storage efficiency. In particular, the repair of any lost fragment involves transfers of additional fragments. The *conventional* way of repairing a lost fragment is to retrieve k fragments from other non-failed nodes, so as to reconstruct the original data and hence the lost fragment. To reduce the *repair bandwidth* (i.e., the amount of transferred traffic for repair), regenerating codes [4] are erasure codes that realize the optimal trade-off between repair bandwidth and storage efficiency, by allowing non-failed nodes to encode and send their stored fragments during repair. One construction of regenerating codes is *minimum storage regenerating (MSR)* codes [4], which minimize the repair bandwidth for reconstructing a single lost fragment while preserving the MDS property.

However, deploying erasure coding in data centers remains challenging due to the hierarchical nature of data centers. A data center is typically organized in multiple *racks*, each comprising multiple nodes for storage. Nodes within each rack are inter-connected via a top-of-rack switch, and the top-of-rack switches of multiple racks are further inter-connected via a network core of switches. To tolerate both node and rack failures, a typical approach is to place fragments in distinct nodes, each of which is located in a distinct rack [6], [8], [11], [13], [14], [16]. Such a placement causes the repair of any lost fragment to inevitably transfer fragments from other non-failed nodes across racks. This incurs substantial cross-rack bandwidth, which is heavily *oversubscribed*, for example, by a factor of 5 to 20 [1], [3], [19] (i.e., the cross-rack capacity available per node in the worst case is only 1/5 to 1/20 of the inner-rack capacity). Thus, our goal is to minimize the *cross-rack repair bandwidth* (i.e., the amount of cross-rack data transferred during repair) in hierarchical data centers. Here, we focus on the repair of a single-node failure, which is the most common failure scenario in practice [8], [13], [14].

We observe that instead of placing one fragment per rack, we can place multiple fragments per rack (while we still keep one fragment per node), and exploit inner-rack regeneration to reduce the cross-rack repair bandwidth. Figure 1 provides a motivating example with $n = 6$ and $k = 3$. Suppose that node 1 fails. Figure 1(a) shows the conventional repair, in which the cross-rack repair bandwidth is the original data size M. Figure 1(b) shows the repair with MSR codes, in which the cross-rack repair bandwidth is $5M/9$ based on the optimality results in [4]. Figure 1(c) shows our new repair scheme, which reduces the cross-rack repair bandwidth to $M/3$, or equivalently, 40% lower than that of MSR codes. The core idea in Figure 1(c) is to perform regeneration *twice*: first within a rack and then across multiple racks. We call this approach *double regeneration*, whose repair design specifically targets the unbalanced nature of inner-rack and cross-rack capacities in hierarchical data centers.

Double regeneration makes two trade-offs. First, the code in Figure 1(c) can only tolerate a single-rack failure (as opposed to three-rack failures in Figures 1(a) and 1(b)). Nevertheless, rack failures are much rarer than node failures in practice [6]. Thus, instead of tolerating multiple node and rack failures, we can tolerate the same multiple-node failures but only a single-rack failure. Second, the sum of the inner-rack and cross-rack repair bandwidths in Figure 1(c) is M, which is higher than that in MSR codes (i.e., $5M/9$). Nevertheless, the inner-rack capacity is more abundant than the cross-rack capacity due to oversubscription. Thus, we can trade the inner-rack repair bandwidth for the cross-rack repair bandwidth.

This paper proposes Double Regenerating Codes (DRC) for

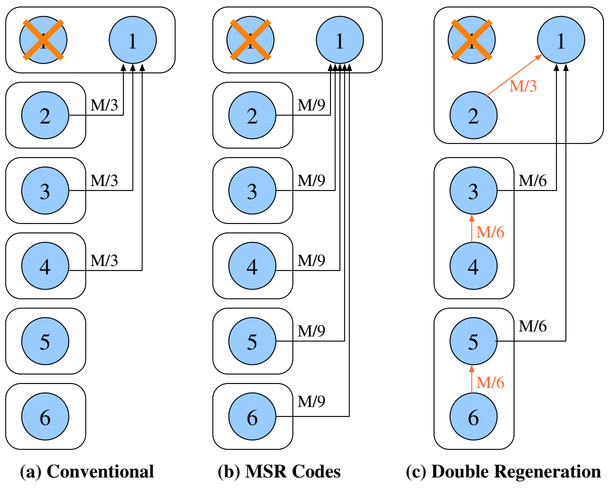


Fig. 1. Example with $n = 6$ and $k = 3$: (a) Conventional repair reconstructs original data and incurs the cross-rack repair bandwidth M; (b) MSR codes incur the cross-rack repair bandwidth $5M/9$ [4]; (c) Double regeneration performs inner-rack encoding and incurs the cross-rack repair bandwidth $M/3$.

hierarchical data centers. Our contributions are two-fold. First, we prove that there exists a DRC construction that minimizes the cross-rack repair bandwidth for a single-node repair, while preserving the MDS property (i.e., DRC maintains the minimum storage redundancy). Second, we show via quantitative comparisons that DRC reduces the cross-rack repair bandwidth of state-of-the-art MSR codes by up to 45.5%.

II. Double Regeneration

We provide a system model that formalizes double regeneration. We analyze the system model via an information flow graph, and derive the lower bound of the cross-rack repair bandwidth under double regeneration.

A. System Model

We encode the original data of size M into n fragments of size M/k each using an (n, k) MDS erasure code. We then distribute the encoded fragments across n nodes (i.e., one encoded fragment per node) that are evenly located in r racks with n/r nodes each. We pose three conditions on the parameters: (i) n is a multiple of r (i.e., $\frac{n}{r}$ is an integer); (ii) we require $\frac{n}{r} \leq k$, so that any single-node failure cannot be repaired locally within a rack and there must be cross-rack repair bandwidth incurred; and (iii) we require $\frac{n}{r} \leq n - k$, so that there is no data loss in a single-rack failure. Note that we can increase the tolerable number of rack failures by adding more redundancy (i.e., increasing $n - k$). Let rack R_h be the h^{th} rack (where $1 \leq h \leq r$), and node $X_{h,i}$ be the i^{th} node in rack R_h (where $1 \leq i \leq n/r$).

A single-node repair in double regeneration works as follows. Without loss of generality, our analysis assumes that node $X_{1,1}$ fails throughout the paper. We select a new node $\underline{X}_{1,1}$ in rack R_1 to restore the lost fragment in $X_{1,1}$. The repair has two stages. In the first repair stage, in rack R_h (where $2 \leq h \leq r$), we select a special node $X_{h,1}$ (without loss of generality), which collects encoded data from each of surviving nodes (including itself) in the same rack (i.e., $X_{h,1}, X_{h,2}, \cdots, X_{h,n/r}$). We call this special node a *relayer*, which re-encodes the collected data and sends the re-encoded data to $\underline{X}_{1,1}$ across racks. In the second repair stage, $\underline{X}_{1,1}$ first collects encoded data from each of other surviving nodes $X_{1,2}, X_{1,3}, \cdots, X_{1,n/r}$ in rack R_1. It then reconstructs the lost fragment using all collected data from within the same rack and from the relayers in other racks.

Our analysis assumes that the bandwidth between any pair of racks is homogeneous and denotes it by β. Accordingly, the cross-rack repair bandwidth for a single-node repair is $(r-1)\beta$. Our goal is to minimize β, while preserving the MDS property.

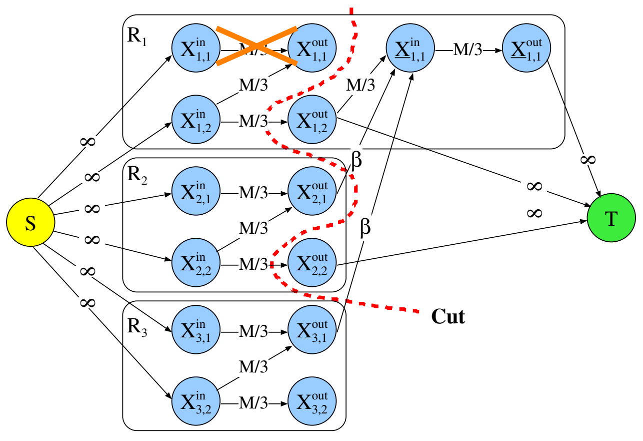


Fig. 2. Information flow graph for $n = 6$, $k = 3$, and $r = 3$.

B. Information Flow Graph

We construct an information flow graph $\mathcal{G}(n, k, r, \beta)$ to describe a data center network for a single-node repair. Figure 2 depicts $\mathcal{G}$ for $n = 6$, $k = 3$, and $r = 3$.

$\mathcal{G}$ has three types of nodes: (i) *virtual source* S and (ii) *data collector* T, which correspond to the source and destination nodes of information flow, respectively; and (iii) the input/output node pair ($X^{in}_{h,i}$, $X^{out}_{h,i}$), which corresponds to the storage node $X_{h,i}$. Similarly, the input/output node pair ($\underline{X}^{in}_{1,1}$, $\underline{X}^{out}_{1,1}$) represents the new node $\underline{X}_{1,1}$.

$\mathcal{G}$ has six types of directed edges: (i) an edge from S to every $X^{in}_{h,i}$ with infinite capacity; (ii) an edge from $X^{in}_{h,i}$ to $X^{out}_{h,i}$, and from $\underline{X}^{in}_{1,1}$ to $\underline{X}^{out}_{1,1}$, with capacity M/k (i.e., the amount of stored data); (iii) an edge from $X^{in}_{h,i}$ to $X^{out}_{h,1}$ ($h \neq 1$,$i \neq 1$) with capacity M/k (i.e., the maximum amount of inner-rack repair traffic between two nodes in R_h); (iv) an edge from $X^{out}_{1,i}$ ($i \neq 1$) to $\underline{X}^{in}_{1,1}$ with capacity M/k (i.e., the maximum amount of inner-rack repair traffic between two nodes in rack R_1); (v) an edge from $X^{out}_{h,1}$ ($h \neq 1$) to $\underline{X}^{in}_{1,1}$ with capacity β (i.e., the maximum amount of cross-rack repair traffic between two racks); and (vi) an edge from each of k selected output nodes to T with infinite capacity for data reconstruction.

C. Lower Bound

We derive the lower bound of β by considering the capacities of all possible min-cuts of $\mathcal{G}$. We define a *cut* as the set of directed edges such that any path from S to T must have at least one edge in the cut. A *min-cut* is the cut that has the minimum sum of capacities of all its edges. Note that there are $\binom{n}{k}$ possible data collectors due to the MDS property. Thus, there are $\binom{n}{k}$ variants of $\mathcal{G}$, and hence $\binom{n}{k}$ possible min-cuts.

Lemma 1 ([4])**.** *If all $\binom{n}{k}$ possible min-cuts of $\mathcal{G}$ separating S and T are no smaller than M, then random linear network codes suffice to rebuild the original data when any k out of n nodes are connected to T, with a probability that is driven arbitrarily to one by increasing the field size.*

Based on Lemma 1, we present the next lemma that specifies the *necessary condition* of the lower bound of β in $\mathcal{G}$ if there exist valid network codes.

Lemma 2. *If the capacities of all $\binom{n}{k}$ possible min-cuts of $\mathcal{G}$ are at least M, then $\beta \geq \frac{M}{k} \cdot \frac{1}{r-\lfloor kr/n \rfloor}$.*

Proof: To specify the min-cut corresponding to each data collector, suppose that the following k nodes are connected to T so that the original data can be rebuilt: (i) the new node $\underline{X}_{1,1}$; (ii) x nodes of rack R_1 (except the failed node $X_{1,1}$); (iii) y relayers; (iv) w_1 nodes whose inner-rack relayers are not connected to T; and (v) w_2 nodes whose inner-rack relayers are connected to T. By definition,

$$1 + x + y + w_1 + w_2 = k. \tag{1}$$

Figure 2 gives $x = 1$, $y = 0$, $w_1 = 1$, and $w_2 = 0$. Let $\Lambda(x, y, w_1, w_2)$ denote the capacity of a cut, which is a function of x, y, w_1, and w_2.

We derive Λ as follows. We do not consider the cut that has an edge directed either from S or to T, since this edge has infinite capacity. We observe that (e.g., see Figure 2): (i) all surviving nodes of R_1 (e.g., $X_{1,2}$) can contribute $(n/r - 1) \cdot M/k$ to Λ; (ii) all nodes whose inner-rack relayers are connected to T (e.g., $X_{2,1}$ and $X_{2,2}$) can contribute $y \cdot n/r \cdot M/k$ to Λ; (iii) all $(r - 1 - y)$ relayers that are not connect to T (e.g., $X_{3,1}$) can contribute $(r - 1 - y)\beta$ to Λ[1]; (iv) the w_1 nodes whose inner-rack relayers are not connected to T can contribute $w_1 \cdot M/k$ to Λ; (v) the w_2 nodes whose inner-rack relayers are connected to T cannot contribute anything to Λ since all its information has been contributed by their inner-rack relayers. Thus, the capacity of a cut is:

$$\begin{aligned} \Lambda &= (n/r - 1) \cdot M/k + y \cdot n/r \cdot M/k \\ &\quad + (r - 1 - y)\beta + w_1 \cdot M/k. \end{aligned} \tag{2}$$

Since Λ of all the $\binom{n}{k}$ min-cuts are at least M, Equation (2) implies that for all $\binom{n}{k}$ variants of $\mathcal{G}$,

$$\beta \geq M/k \cdot (n/r - \frac{w_1 + n - k - 1}{r - 1 - y}). \tag{3}$$

[1]$X_{3,1}$ can contribute $2M/k$ to Λ, but β is clearly no larger than $2M/k$ since the former has the encoded information from the latter.

Let $\beta'(y, w_1)$ be the right side of Equation (3). Then for all $\binom{n}{k}$ variants of $\mathcal{G}$, Equation (3) can be reduced to:

$$\beta \geq \max\{\beta'(y, w_1)\}. \tag{4}$$

We now derive $\max\{\beta'(y, w_1)\}$. At most $n/r - 1$ nodes in rack R_1 (with failed node $X_{1,1}$) can be connected to T. Thus,

$$x \leq n/r - 1. \tag{5}$$

Also, for each rack whose relayer is connected to T, at most $n/r - 1$ nodes can also be connected to T. Thus,

$$w_2 \leq (n/r - 1)y. \tag{6}$$

By Equation (1), we observe that larger values of x and w_2 will make a smaller value range for y and w_1. By Equation (3), we observe that $\beta'(y, w_1) \propto -y$ and $\beta'(y, w_1) \propto -w_1$. Thus, $\beta'(y, w_1)$ is maximized when x and w_2 attain their maximum values (i.e., $x = n/r - 1$ and $w_2 = (n/r - 1)y$). In this case, Equation (1) can be reduced to:

$$w_1 = k - n/r - y \cdot n/r. \tag{7}$$

Since $w_1 \geq 0$, Equation (7) implies that

$$y \leq \lfloor kr/n \rfloor - 1, \quad y \in \mathbb{Z}. \tag{8}$$

Therefore, if Equations (7) and (8) hold, $\max\{\beta'(y, w_1)\}$ can be derived. That is,

$$\max\{\beta'(y, w_1)\} = \frac{M}{k} \cdot \frac{1}{r - \lfloor kr/n \rfloor} \tag{9}$$

By Equations (4) and (9), Lemma 2 concludes. $\square$

III. Code Construction

Lemma 2 provides the necessary condition of the lower bound of β if we ensure the existence of random linear network codes. If we can construct linear codes that match the lower bound of β, the bound is tight and the code construction is bandwidth-optimal. We now propose such a linear code construction, called *Double Regenerating Codes (DRC)*, such that its optimal single-node repair satisfies $\beta = \frac{M}{k} \cdot \frac{1}{r-\lfloor kr/n \rfloor}$ and maintains the MDS property after a single-node repair. We prove the existence of DRC, by extending the proof of [20] for hierarchical data centers.

To explain our DRC construction, we extend our system model in Section II-A. We divide the original data of size M into qk (uncoded) *blocks*, where $q = r - \lfloor kr/n \rfloor$, and transform them into qn encoded blocks. Each node stores an encoded fragment consisting of q encoded blocks, each of which has size equal to the lower bound of β. For each node $X_{h,i}$ (where $1 \leq h \leq r$ and $1 \leq i \leq n/r$), its j^{th} (encoded) block (where $1 \leq j \leq q$) is a linear combination of the qk original blocks over a finite field $\mathbb{F}$. Let $\mathbf{p}_{h,i,j}$ be a column vector of size qk that specifies the coefficients for the above linear combination. Let $\mathbf{P}_{h,i}$ be a $qk \times q$ matrix comprising the column vectors $\{\mathbf{p}_{h,i,j}\}_{1 \leq j \leq q}$. Thus, we can now specify DRC by the collection $\{\mathbf{P}_{h,i}\}_{1 \leq h \leq r, 1 \leq i \leq n/r}$. In the following, we use $\mathbf{P}_{h,i}$ and $\mathbf{p}_{h,i,j}$ to refer to the fragment and the j^{th} block stored in $X_{h,i}$, respectively.

The single-node repair under DRC works as follows, based on the system model in Section II-A. Suppose that $X_{1,1}$ fails, and we reconstruct a new fragment $P'_{1,1}$ in a new node $\underline{X}_{1,1}$. In the first repair stage, each relayer $X_{h,1}$ (where $2 \leq h \leq r$) computes a new block, $\mathbf{p}'_h$, from all stored blocks in rack R_h:

$$\mathbf{p}'_h = [\mathbf{P}_{h,1}; \mathbf{P}_{h,2}; \cdots; \mathbf{P}_{h,n/r}] \cdot \mathbf{c}_h, \tag{10}$$

where $\mathbf{c}_h$ denotes a coefficient vector of size qn/r. In the second repair stage, $\underline{X}_{1,1}$ computes a new fragment, $\mathbf{P}'_{1,1}$, from all the surviving blocks in rack R_1 as well as $\mathbf{p}'_h$'s from the relayers in rack R_h's (where $2 \leq h \leq r$):

$$\mathbf{P}'_{1,1} = [\mathbf{P}_{1,2}; \cdots; \mathbf{P}_{1,n/r}; \mathbf{p}'_2; \cdots \mathbf{p}'_r] \cdot \mathbf{D}, \tag{11}$$

where $\mathbf{D}$ is a $(q(n/r-1)+r-1) \times q$ coefficient matrix.

To maintain the MDS property, we ensure that for any k out of n nodes, the span of the qk vectors of any k nodes has full rank. Let $\mathcal{U}$ be a set of any $k-1$ fragments out of all $\mathbf{P}_{h,i}$'s except $\mathbf{P}_{1,1}$ (i.e., $\mathcal{U}$ is a set of any $k-1$ surviving fragments). We first present the following lemma.

Lemma 3. *Consider the collection $\{\mathbf{P}_{h,i}\}_{1\leq h\leq r, 1\leq i\leq n/r}$ that satisfies the MDS property and let $\{\mathbf{P}_{1,2}, \cdots, \mathbf{P}_{1,n/r}\} \subseteq \mathcal{U}$. It is possible to select the following qk vectors whose span has full rank: (i) we select q vectors from each of $k-1$ fragments in $\mathcal{U}$; (ii) there must exist q racks excluding R_1 such that each of these q racks has at least one fragment that is not in $\mathcal{U}$, and we select one vector from each of these q fragments.*

Proof: The fragments in $\mathcal{U}$ can fully cover at most $1 + \lfloor \frac{(k-1)-(n/r-1)}{n/r} \rfloor = \lfloor kr/n \rfloor$ racks including R_1. Thus, we must be able to find $r - \lfloor kr/n \rfloor = q$ racks (without R_1), such that each of these racks has one fragment that does not belong to $\mathcal{U}$. We call these q fragments $\mathbf{P}_{h_1,i_1}, \cdots, \mathbf{P}_{h_q,i_q}$.

Our proof is similar to that of Lemma 4 of [20]. We define a set $\mathcal{V}$ that is initialized as $\emptyset$. We find one vector from one of the q fragments $\mathbf{P}_{h_1,i_1}, \cdots, \mathbf{P}_{h_q,i_q}$ that is linearly independent of the set of vectors currently in $\mathcal{V} \cup \mathcal{U}$; if so, we add the vector to $\mathcal{V}$. We repeat this process for the remaining fragments iteratively until we add q vectors to $\mathcal{V}$.

We argue that we can always add one vector to $\mathcal{V}$ in each iteration. We prove by contradiction. Suppose that we cannot find such a vector from the remaining fragments before the m^{th} iteration. For any remaining fragment, say $\mathbf{P}_{h_m,i_m}$, the span of vectors in $\mathcal{V} \cup \mathcal{U} \cup \{\mathbf{P}_{h_m,i_m}\}$ has rank less than $q(k-1)+q = qk$ (i.e., it does not have full rank). However, the span of $\mathcal{U} \cup \{P_{h_m,i_m}\}$ must have full rank because of the MDS property (since they represent k surviving fragments). This leads to a contradiction. Thus, the span of the qk vectors in $\mathcal{V} \cup \mathcal{U}$ must have full rank after q iterations, and the lemma holds. □

We now prove the existence of DRC by showing that it maintains the MDS property after a single-node repair.

Theorem 1. *There exists a linear coding construction for DRC defined in the finite field $\mathbb{F}$, such that the MDS property is still maintained after a single-node repair with a probability arbitrarily driven to 1 by increasing the field size of $\mathbb{F}$.*

Proof: Suppose that the collection $\{\mathbf{P}_{h,i}\}_{1\leq h\leq r, 1\leq i\leq n/r}$ satisfies the MDS property initially. We show that there exist $\mathbf{c}_h$ $(2 \leq h \leq r)$ and $\mathbf{D}$ (resp. Equations (10) and (11)), such that after a single-node repair, the new collection $\{\mathbf{P}'_{1,1}, \mathbf{P}_{1,2}, \cdots, \mathbf{P}_{1,n/r}, \cdots, \mathbf{P}_{r,1}, \cdots, \mathbf{P}_{r,n/r}\}$ maintains the MDS property.

Clearly, all surviving fragments in $\{\mathbf{P}_{h,i}\}_{2\leq h\leq r, 1\leq i\leq n/r}$ satisfy the MDS property. We only need to show that the span of the qk vectors in $\{P'_{1,1}\} \cup \mathcal{U}$ has full rank for any possible $\mathcal{U}$. We consider two cases of $\mathcal{U}$.

Case 1: $\{\mathbf{P}_{1,2}, \cdots, \mathbf{P}_{1,n/r}\} \subseteq \mathcal{U}$. Based on Equations (10) and (11), we can tune $\mathbf{c}_h$ and $\mathbf{D}$ such that $\mathbf{P}'_{1,1}$ is composed of the q vectors out of q different racks aside R_1 (i.e., the q vectors of $\mathcal{V}$ after q iterations), and the span of the q vectors plus the vectors in $\mathcal{U}$ have full rank, based on Lemma 3.

Case 2: $\{\mathbf{P}_{1,2}, \cdots, \mathbf{P}_{1,n/r}\} \subsetneq \mathcal{U}$. In other words, there exists one $P_{1,i'} \notin \mathcal{U}$ (where $2 \leq i' \leq n/r$). By Equation (11), we can tune $\mathbf{D}$ so that $P'_{1,1}$ is composed of the q vectors of $P_{1,i'}$, and the set of vectors in $\{P_{1,i'}\} \cup \mathcal{U}$ must have full rank due to the MDS property.

For both cases, we can show that $\det(\{\mathbf{P}'_{1,1}, \mathcal{U}\})$ is a nonzero number for some assignments of $\mathbf{c}_h$ and $\mathbf{D}$ since the span of vectors in $\{\mathbf{P}'_{1,1}, \mathcal{U}\}$ has full rank. This means that $\det(\{\mathbf{P}'_{1,1}, \mathcal{U}\})$ is a non-zero polynomial. Thus, $\det(\{\mathbf{P}'_{1,1}, \mathcal{U}\}) \neq 0$ holds with a probability arbitrarily driven to one by increasing the field size of $\mathbb{F}$, as a result of the Schwartz-Zippel Theorem [10]. The arguments can be found in [20] and we omit details here. Thus, Theorem 1 holds. □

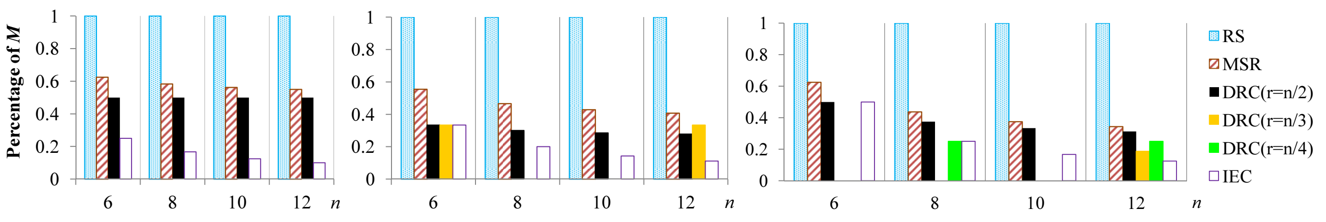
IV. Quantitative Comparisons

We obtain the cross-rack repair bandwidth for a single-node repair for several erasure codes:

- **RS:** Reed-Solomon (RS) codes [15] use the conventional repair (Section I), whose cross-rack repair bandwidth equals the original data size M.
- **MSR:** From [4], the repair bandwidth of MSR codes is $\frac{M}{k} \cdot \frac{d}{d-k+1}$ when the new node connects to d surviving nodes. Since MSR-coded data is spread across nodes in distinct racks [13], the minimum cross-rack repair bandwidth is the same as the minimum repair bandwidth. We choose the minimum cross-rack repair bandwidth at $d = n-1$, i.e., $\frac{M}{k} \cdot \frac{n-1}{n-k}$.
- **DRC:** The minimum cross-rack repair bandwidth (from $r-1$ surviving racks) is $(r-1)\min\{\beta\} = \frac{M}{k} \cdot \frac{r-1}{r-\lfloor kr/n \rfloor}$.
- **IEC:** For comparisons, we borrow the idea from [4] and assume the existence of ideal erasure codes (IEC), whose cross-rack repair bandwidth equals the lost data size of the failed node, i.e., M/k.

We set $n-k = 2, 3, 4$, which tolerates at most four failures as in practical data centers [8], [16]. For DRC, we set r under following conditions (Section II-A): n is a multiple of r, $r \geq \frac{n}{k}$, and $r \geq \frac{n}{n-k}$. Also, we make r an integer.

Figure 3 depicts the cross-rack repair bandwidths of different erasure codes versus the number of nodes n in terms of the percentage of the original data size M. The cross-rack repair bandwidth of DRC is always less than that of MSR codes, by up to 45.5% (e.g., $n = 12, k = 8, r = 4$). Also, DRC has the



(a) $n - k = 2$ (b) $n - k = 3$ (c) $n - k = 4$

Fig. 3. Cross-rack repair bandwidth (in percentage of M) versus n for repairing a single-node failure. We only plot DRC where r is an integer.

same cross-rack repair bandwidth as IEC in some cases (e.g., $n = 6, k = 3, r = 3$). In general, under the same n and k, when r is smaller, the cross-rack repair bandwidth of DRC becomes smaller as well; however, there exist some cases that a smaller r increases the cross-rack repair bandwidth (e.g., when $n = 12$, $k = 8$, and r changes from 4 to 3). The reason is that the "floor" function may keep $\lfloor \frac{kr}{n} \rfloor$ the same even if r becomes smaller.

V. Related Work

Some follow-up studies on regenerating codes consider the heterogeneity of network bandwidth in the repair problem. For example, tree-structured regeneration [9] allows nodes to relay repair traffic in a tree topology, thereby reducing the overall repair time. Other studies (e.g., [2], [5], [17]) consider heterogeneity of node or link resources, and derive the minimum repair bandwidth or the maximum information theoretic capacity. Our work focuses on minimizing the more critical cross-rack bandwidth in data centers.

The special topological structure of rack-based data centers motivates different repair studies. The studies [7], [12] focus on a two-rack topology. The work [18] considers a multi-rack topology, but focuses on locally repairable codes (which are non-MDS). In addition, the above approaches do not exploit node cooperation within a rack to minimize the cross-rack repair bandwidth. Our DRC is MDS and provably minimizes the cross-rack repair bandwidth.

VI. Conclusions

We present Double Regenerating Codes (DRC) for erasure-coded storage in hierarchical data centers. DRC minimizes the cross-rack repair bandwidth for a single-node repair, while keeping minimum storage redundancy. We prove its existence and demonstrate the repair effectiveness of DRC.

VII. Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61502191), CCF-Tencent Open Fund 2015, Wuhan National Laboratory for Optoelectronics School of Computer Science and Technology, University Grants Committee of Hong Kong (Grant No. AoE/E-02/08), Cisco University Research Program Fund (CG#593756) from Silicon Valley Community Foundation, and Research Committee of CUHK.

References

[1] F. Ahmad, S. T. Chakradhar, A. Raghunathan, and T. Vijaykumar. ShuffleWatcher: Shuffle-aware Scheduling in Multi-tenant MapReduce Clusters. In *Proc. of USENIX ATC*, 2014.

[2] S. Akhlaghi, A. Kiani, and M. R. Ghanavati. Cost-bandwidth tradeoff in distributed storage systems. *Computer Communications*, 33(17):2105–2115, 2010.

[3] T. Benson, A. Akella, and D. A. Maltz. Network Traffic Characteristics of Data Centers in the Wild. In *Proc. of ACM IMC*, 2010.

[4] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran. Network Coding for Distributed Storage Systems. *IEEE Trans. on Info. Theory*, 56(9):4539–4551, Sep 2010.

[5] T. Ernvall, S. El Rouayheb, C. Hollanti, and H. V. Poor. Capacity and Security of Heterogeneous Distributed Storage Systems. *IEEE JSAC*, 31(12):2701–2709, 2013.

[6] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan. Availability in Globally Distributed Storage Systems. In *Proc. of USENIX OSDI*, 2010.

[7] B. Gaston, J. Pujol, and M. Villanueva. A Realistic Distributed Storage System That Minimizes Data Storage and Repair Bandwidth. In *Proc. of Data Compression Conf.*, 2013.

[8] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin. Erasure Coding in Windows Azure Storage. In *Proc. of USENIX ATC*, 2012.

[9] J. Li, S. Yang, X. Wang, and B. Li. Tree-structured Data Regeneration in Distributed Storage Systems with Regenerating Codes. In *Proc. of IEEE INFOCOM*, 2010.

[10] R. Motwani and P. Raghavan. Randomized Algorithms. In *Cambridge University Press*, 1995.

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

[12] J. Pernas, C. Yuen, B. Gastón, and J. Pujol. Non-homogeneous Two-rack Model for Distributed Storage Systems. In *Proc. of IEEE ISIT*, 2013.

[13] K. V. Rashmi, P. Nakkiran, J. Wang, N. B. Shah, and K. Ramchandran. Having Your Cake and Eating It Too: Jointly Optimal Erasure Codes for I/O, Storage, and Network-bandwidth. In *Proc. of USENIX FAST*, 2015.

[14] K. V. Rashmi, N. B. Shah, D. Gu, H. Kuang, D. Borthakur, and K. Ramchandran. A Hitchhiker's Guide to Fast and Efficient Data Reconstruction in Erasure-coded Data Centers. In *Proc. of ACM SIGCOMM*, 2014.

[15] I. Reed and G. Solomon. Polynomial Codes over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[16] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur. Xoring Elephants: Novel Erasure Codes for Big Data. In *Proc. of VLDB Endowment*, 2013.

[17] N. B. Shah, K. V. Rashmi, and P. V. Kumar. A Flexible Class of Regenerating Codes for Distributed Storage. In *Proc. of IEEE ISIT*, 2010.

[18] M. A. Tebbi, T. H. Chan, and C. W. Sung. A Code Design Framework for Multi-rack Distributed Storage. In *IEEE ITW*, 2014.

[19] A. Vahdat, M. Al-Fares, N. Farrington, R. N. Mysore, G. Porter, and S. Radhakrishnan. Scale-out networking in the data center. *IEEE Micro*, 30(4):29–41, 2010.

[20] Y. Wu, A. G. Dimakis, and K. Ramchandran. Deterministic regenerating codes for distributed storage. In *Allerton Conference on Control, Computing and Communication*, 2007.