Double Regenerating Codes for Hierarchical Data Centers

Yuchong Hu, Patrick P. C. Lee, and Xiaoyang Zhang

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 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...
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, 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., $n = r \cdot h$ is an integer); (ii) we require $\frac{2}{3} \leq k$, 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 $\frac{2}{3} \leq n - k$, 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 $X_{1,1}$ 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., $X_{h,1}, X_{h,2}, \ldots, X_{h,n/r}$). We call this special node a relayer, which re-encodes the collected data and sends the re-encoded data to $X_{1,1}$ across racks. In the second repair stage, $X_{1,1}$ first collects encoded data from each of other surviving nodes $X_{1,2}, X_{1,3}, \ldots, X_{1,n/r}$ 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 $\beta$. Accordingly, the cross-rack repair bandwidth for a single-node repair is $(r-1)\beta$. Our goal is to minimize $\beta$, while preserving the MDS property.

B. Information Flow Graph

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

$G$ 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 $(X_{h,i}^{in}, X_{h,i}^{out})$, which corresponds to the storage node $X_{h,i}$. Similarly, the input/output node pair $(\Lambda_{1,1}^{in}, \Lambda_{1,1}^{out})$ represents the new node $\Lambda_{1,1}$.

$G$ has six types of directed edges: (i) an edge from $S$ to every $X_{h,1}^{in}$ with infinite capacity; (ii) an edge from $X_{h,1}^{in}$ to $X_{h,1}^{out}$, and from $X_{h,1}^{in}$ to $\Lambda_{1,1}^{out}$, with capacity $M/k$ (i.e., the amount of stored data); (iii) an edge from $X_{h,1}^{in}$ to $X_{h,1}^{out}$ (i.e., the amount of inner-rack repair traffic between two nodes in $R_h$); (iv) an edge from $X_{h,1}^{out}$ (i.e., the amount of inner-rack repair traffic between two nodes in $R_h$); (v) an edge from $X_{h,1}^{out}$ (i.e., the amount of inner-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 $\beta$ by considering the capacities of all possible min-cuts of $G$. 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 $G$, and hence $\binom{n}{k}$ possible min-cuts.

Lemma 1 ([14]). If all $\binom{n}{k}$ possible min-cuts of $G$ 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 $\beta$ in $G$ if there exist valid network codes.

Lemma 2. If the capacities of all $\binom{n}{k}$ possible min-cuts of $G$ are at least $M$, then $\beta \geq \frac{M}{k} \cdot \frac{1}{r - \left[\frac{kr}{n}\right]}$.

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 $X_1$; (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,

$$1 + x + y + w_1 + w_2 = k.$$  

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 $\Lambda$ 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 $\Lambda$; (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 $\Lambda$; (iii) all $(r - 1 - y)$ relayers that are not connect to $T$ (e.g., $X_{3,1}$) can contribute $(r - 1 - y)\beta$ to $\Lambda'$; (iv) the $w_1$ nodes whose inner-rack relayers are not connected to $T$ can contribute $w_1 \cdot M/k$ to $\Lambda$; (v) the $w_2$ nodes whose inner-rack relayers are connected to $T$ cannot contribute anything to $\Lambda$ since all its information has been contributed by their inner-rack relayers. Thus, the capacity of a cut is:

$$\Lambda = (n/r - 1) \cdot M/k + y \cdot n/r \cdot M/k + (r - 1 - y)\beta + w_1 \cdot M/k.$$  

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

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

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

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

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, $x \leq n/r - 1$.

Also, for each rack whose relayer is connected to $T$, at most $n/r - 1$ nodes can also be connected to $T$. Thus, $w_2 \leq (n/r - 1)y$.

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:

$$w_1 = k - n/r - y \cdot n/r.$$  

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

$$y \leq \left[\frac{kr}{n}\right] - 1, \quad y \in \mathbb{Z}.$$  

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

$$\max \{\beta'(y, w_1)\} = \frac{M}{k} \cdot \frac{1}{r - \left[\frac{kr}{n}\right]}.$$  

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

III. Code Construction

Lemma 2 provides the necessary condition of the lower bound of $\beta$ if we ensure the existence of random linear network codes. If we can construct linear codes that match the lower bound of $\beta$, 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 $\beta = \frac{M}{k} \cdot \frac{1}{r - \left[\frac{kr}{n}\right]}$ 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 $q = r - \left[\frac{kr}{n}\right]$, 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 $\beta$. 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 $F$. Let $p_{h,i,j}$ be a column vector of size $qk$ that specifies the coefficients for the above linear combination. Let $P_{h,i}$ be a $qk \times q$ matrix comprising the column vectors $\{p_{h,i,j}\}_{1 \leq j \leq q}$. Thus, we can now specify DRC by the collection $\{P_{h,i}\}_{1 \leq h \leq r, 1 \leq i \leq n/r}$. In the following, we use $P_{h,i}$ and $p_{h,i,j}$ 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 \(X_{1,1}'\). In the first repair stage, each relay \(X_{h,1}\) (where \(2 \leq h \leq r\)) computes a new block, \(p'_h\), from all stored blocks in rack \(R_h\):

\[
p'_h = [p_{h,1}; p_{h,2}; \cdots; p_{h,n/r}] \cdot c_h,
\]

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

\[
p'_{1,1} = [p_{1,2}; \cdots; p_{1,n/r}; p'_2; \cdots; p'_h] \cdot D,
\]

where \(D\) 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 \(U\) be a set of any \(k - 1\) fragments out of all \(P_{h,i}\)’s except \(P_{1,1}\) (i.e., \(U\) is a set of any \(k - 1\) surviving fragments). We first present the following lemma.

**Lemma 3.** Consider the collection \([P_{h,1}]_{1 \leq h \leq r, 1 \leq i \leq n/r}\) satisfying the MDS property and let \([P_{1,2}, \cdots, P_{1,n/r}]\) \(\subseteq U\). It is possible to select the following \(qk\) vectors whose span has full rank: (i) we select \(q\) vectors from each of \(k\) fragments in \(U\); (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 \(U\), and we select one vector from each of these \(q\) fragments.

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

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

We argue that we can always add one vector to \(V\) 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 \(P_{m,m}\), the span of vectors in \(V \cup U\) \(\cup \{P_{h,m}\}\) has rank less than \(q(k-1)+q=qk\) (i.e., it does not have full rank). However, the span of \(U \cup \{P_{h,m}\}\) 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 \(V \cup U\) 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 \([P_{h,1}]_{1 \leq h \leq r, 1 \leq i \leq n/r}\) satisfies the MDS property initially. We show that there exist \(c_h\) (\(2 \leq h \leq r\)) and \(D\) (resp. Equations (10) and (11)), such that after a single-node repair, the new collection \([P'_{1,1}, P'_{1,2}, \cdots, P'_{1,n/r}; P_{2,r}, \cdots, P_{r,n/r}]\) maintains the MDS property. Clearly, all surviving fragments in \([P_{h,1}]_{2 \leq h \leq r, 1 \leq i \leq n/r}\) satisfy the MDS property. We only need to show that the span of the \(qk\) vectors in \([P'_{1,1}] \cup U\) has full rank for any possible \(U\). We consider two cases of \(U\).

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

Case 2: \([P_{1,2}, \cdots, P_{1,n/r}] \not\subseteq U\). In other words, there exists one \(P_{h,i} \not\in U\) (where \(2 \leq i \leq n/r\)). By Equation (11), we can tune \(D\) so that \(P'_{1,1}\) is composed of the \(q\) vectors of \(P_{1,i}\), and the set of vectors in \([P_{1,i}] \cup U\) must have full rank due to the MDS property.

For both cases, we can show that \(\text{det}([P'_{1,1}, U])\) is a nonzero number for some assignments of \(c_h\) and \(D\) since the span of the \(qk\) vectors in \([P'_{1,1}] \cup U\) has full rank. This means that \(\text{det}([P'_{1,1}, U])\) is a non-zero polynomial. Thus, \(\text{det}([P'_{1,1}, U]) \neq 0\) 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 \(\frac{M}{r} \cdot \frac{q}{q-1}\) 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., \(\frac{M}{r} \cdot \frac{n-1}{n-1}\).
- **DRC:** The minimum cross-rack repair bandwidth (from \(r - 1\) surviving racks) is \((r - 1) \min(\beta) = \frac{M}{r} \cdot \frac{r - 1}{r - [kr/n]}\).
- **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 \(r, r \geq \frac{q}{q-1}\), and \(r \geq \frac{n}{n-r}\). 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
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 \(n = 12, k = 8, r\) changes from 4 to 3). The reason is that the “floor” function may keep \(\lfloor \frac{r}{n} \rfloor\) 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