SCIENCE CHINA
Information Sciences

• LETTER •

July 2022, Vol. 65 179301:1–179301:2
https://doi.org/10.1007/s11432-021-3304-6

Explicit construction of minimum bandwidth rack-aware regenerating codes

Liyang ZHOU[1,2] & Zhifang ZHANG[1,2*]

[1]*Key Laboratory of Mathematics Mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing* 100049*, China*

Received 18 April 2021/Revised 13 June 2021/Accepted 20 July 2021/Published online 31 March 2022

Citation Zhou L Y, Zhang Z F. Explicit construction of minimum bandwidth rack-aware regenerating codes. Sci China Inf Sci, 2022, 65(7): 179301, https://doi.org/10.1007/s11432-021-3304-6

Dear editor,

Erasure codes are increasingly adopted in modern storage systems (such as Windows Azure Storage [1] and Facebook storage [2]) to ensure fault-tolerant storage with low redundancy. Meanwhile, efficient repair of node failures becomes a central issue in coding for distributed storage. The repair efficiency is usually measured by the repair bandwidth which is the amount of data transmitted from the helper nodes during the repair process. The celebrated work [3] initiated the study of regenerating codes that can minimize the repair bandwidth for given storage redundancy. By now fruitful results have been achieved in regenerating codes which are partly included in the survey [4].

The initial model for regenerating codes treats the bandwidth cost equally between all storage nodes. For simplicity, we call this model as the homogeneous model throughout. However, large data centers usually possess heterogeneous structure where the storage nodes are organized in racks, so it permits to differentiate between the intra-rack communication and cross-rack communication. Specifically, suppose there are $\bar{n}$ racks each containing u out of the n nodes. To store a file of B symbols, each node stores α symbols such that any k nodes can together recover the file. When a node failure happens, its storage contents can be regenerated in a replacement node by downloading data from all the surviving nodes in the same rack and also from $\bar{d}$ helper racks. Because the cross-rack communication cost is much more expensive than the intra-rack communication cost, the latter is usually neglected in calculating the repair bandwidth. Thus the repair bandwidth $\gamma = \bar{d}\beta$ where β is the number of symbols transmitted from each helper rack to the replacement node.

Particularly when $u=1$, it degenerates into the homogenous model where regenerating codes have been well studied. In this study we focus on the case $u>1$. Similarly to [3], Hou et al. [5] derived the cut-set bound for the rack-aware storage model which along with some boundary conditions characterizes an α-β tradeoff curve. The two extreme points on the tradeoff curve indicate the rack-aware regenerating codes with the minimum bandwidth (i.e., MBRR codes) and those with the minimum storage (i.e., MSRR codes), respectively. Explicit constructions of MSRR codes were developed in [6] while no explicit constructions of MBRR codes have been found so far.

According to [5], the MBRR code has parameters

$$\alpha = \bar{d}\beta = B\bar{d}/\left(k\bar{d} - \frac{\bar{k}(\bar{k}-1)}{2}\right), \tag{1}$$

where $\bar{k} = \lfloor \frac{k}{u} \rfloor$. The authors of [5] provided an existential construction of MBRR codes over a field of size larger than $B\sum_{i=1}^{\min\{k,\bar{n}\}}\binom{n-\bar{n}}{k-i}\binom{\bar{n}}{i}$. In their construction, a product-matrix structure of the MBR codes in [7] was used to ensure the optimal repair bandwidth, while the data reconstruction from any k nodes was guaranteed by the invertibility of some related matrices. Unfortunately, they failed to give explicit constructions of the corresponding matrices. We conquered this problem through a nice merge of the multiplicative subgroup design into the product-matrix MBR codes. The multiplicative subgroup design was first adopted in [8] for ensuring repair locality in linear codes with the optimal distance, and then in [6] for constructing MSRR codes from the parity-check matrix. By applying the design in the product-matrix framework, we obtain the first explicit construction of MBRR codes for all admissible parameters considered in [5].

First introduce some notations. For integers $n > m \geqslant 0$, let $[m,n]=\{m,m+1,\ldots,n\}$ and $[n]=\{1,\ldots,n\}$. We label each of the n nodes by a pair (e,g) where $e \in [0,\bar{n}-1]$ indicates which rack the node lies in and $g \in [0,u-1]$ is the node index within the rack. Our MBRR codes are built over a finite field F satisfying $u|(|F|-1)$ and $|F|>n$. The codes apply to the scalar case $\beta=1$. Accordingly, the file is composed of $B = (k-\bar{k})\bar{d} + \frac{\bar{k}(\bar{k}+1)}{2} + \bar{k}(\bar{d}-\bar{k})$ symbols from F and each node stores $\bar{d}$ symbols. Next we describe the construction in three steps.

Step 1. Define two sets $J_1 = \{tu+u-1 : t \in [0,\bar{d}-1]\}$

* Corresponding author (email: zfz@amss.ac.cn)

© Science China Press and Springer-Verlag GmbH Germany, part of Springer Nature 2022

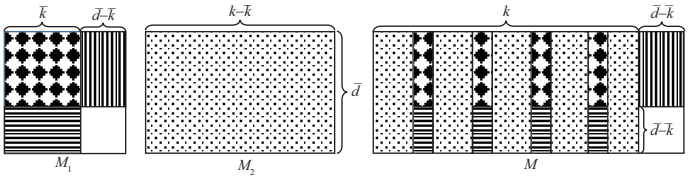

Figure 1 Construction of the matrix M.

and $J_2 = [0, k-1] - J_1$. It can be seen that $|J_1| = \bar{d}$ and $|J_2| = k - \bar{k}$. Moreover, let $J = J_1 \cup J_2$ which can be rewritten as

$$J = [0, k-1] \cup \{tu + u - 1 : t \in [\bar{k}, \bar{d} - 1]\}. \quad (2)$$

Then the B symbols of the file are arranged in a $\bar{d} \times (k - \bar{k} + \bar{d})$ matrix $M = (m_{i,j})_{i \in [0, \bar{d}-1], j \in J}$. Note the columns of M are indexed by the set J. When restricted to the columns indexed by J_1 (respectively, J_2), the resulting sub-matrix of M is denoted as M_1 (respectively, M_2). Moreover, M_1 has the following form:

$$M_1 = \begin{pmatrix} S & T \\ T^\tau & 0 \end{pmatrix}, \quad (3)$$

where S is a symmetric matrix of order $\bar{k}$ whose upper-triangular half contains exactly $\frac{\bar{k}(\bar{k}+1)}{2}$ symbols of the file, T is a $\bar{k} \times (\bar{d} - \bar{k})$ matrix composed of $\bar{k}(\bar{d} - \bar{k})$ symbols of the file, and T^τ denotes the transpose of T. As a result, M_1 is a $\bar{d} \times \bar{d}$ symmetric matrix containing $\frac{\bar{k}(\bar{k}+1)}{2} + \bar{k}(\bar{d} - \bar{k})$ symbols altogether. The matrix M_2 is a $\bar{d} \times (k - \bar{k})$ matrix containing the remaining $(k - \bar{k})\bar{d}$ symbols of the file. The construction of the matrix M is displayed in Figure 1.

Step 2. For $i \in [0, \bar{d} - 1]$ define polynomials:

$$f_i(x) = \sum_{j \in J} m_{i,j} x^j = \sum_{j=0}^{k-1} m_{i,j} x^j + \sum_{t=\bar{k}}^{\bar{d}-1} m_{i,tu+u-1} x^{tu+u-1}, \quad (4)$$

where the second equality comes from the form of J in (2). In short, each row of the matrix M defines a polynomial.

Step 3. Choose a primitive element of F, denoted by ξ. Let $\eta = \xi^{\frac{|F|-1}{u}}$. Obviously, $\eta^u = 1$. Note ξ and η are fixed and publicly known. Then let $\lambda_{(e,g)} = \xi^e \eta^g$ for $e \in [0, \bar{n} - 1], g \in [0, u - 1]$. For $(e, g) \neq (e', g')$, it is easy to see $\lambda_{(e,g)} \neq \lambda_{(e',g')}$. That is, we select n distinct elements $\lambda_{(e,g)}$'s in F, where $(e, g) \in [0, \bar{n} - 1] \times [0, u - 1]$. Finally we construct a code C by letting the node (e, g) store the $\bar{d}$ symbols $f_0(\lambda_{(e,g)}), f_1(\lambda_{(e,g)}), \ldots, f_{\bar{d}-1}(\lambda_{(e,g)})$.

In short, our code C is built as

$$C = M\Lambda, \quad (5)$$

where M is the matrix constructed from the data file as in Step 1, $\Lambda = (\lambda_{(e,g)}^j)$ has $k - \bar{k} + \bar{d}$ rows indexed by $j \in J$ and n columns indexed by $(e, g) \in [0, \bar{n} - 1] \times [0, u - 1]$. Then C is a $\bar{d} \times n$ code matrix each column of which contains exactly the $\bar{d}$ symbols stored in a node. We prove C is an MBRR code by showing it satisfies the data reconstruction property (Theorem 1) and optimal repair property (Theorem 2). Proofs of the two theorems are written in Appendixes A and B, respectively.

Theorem 1. The matrix M (i.e., the data file) can be recovered from any k columns of C.

Remark 1. The key observation is that the bottom $\bar{d} - \bar{k}$ polynomials (see Figure 1) have degree less than k. Thus any k nodes are sufficient to recover these polynomials. Then due to the symmetry of M_1, the above $\bar{k}$ polynomials also degenerate to polynomials of degree less than k based on recovery of the bottom polynomials.

Theorem 2. Any single node failure (i.e., any column of C) can be recovered by downloading $\beta = 1$ symbol from each of $\bar{d}$ helper racks in addition to the transmission within the rack containing the failed node.

Remark 2. The key idea is that due to the multiplicative subgroup structure in Λ (i.e., η has multiplicative order u), the punctured code C_e for each rack $e \in [0, \bar{n} - 1]$ matches evaluations of $\bar{d}$ polynomials of degree less than u. Moreover, the leading coefficients of these polynomials form a product-matrix MBR code in [7].

Additionally, we present in Appendix C a transformation to convert the code C in (5) into a systematic MBRR code where the B file symbols are stored in an uncoded form in k systematic nodes. This makes our code more desirable in practice because the file symbols can be accessed directly from the systematic nodes. In summary, our construction is explicit, systematic, built over small field and with small sub-packetization (i.e., $\alpha = \bar{d}$), so our study provides a practical solution to the MBRR codes.

Acknowledgements This work was supported in part by National Key R&D Program of China (Grant No. 2020YFA0712300) and National Natural Science Foundation of China (Grant No. 61872353).

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Huang C, Simitci H, Xu Y, et al. Erasure coding in windows azure storage. In: Proceedings of the 2012 USENIX Annual Technical Conference, Boston, 2012. 15–26
2 Sathiamoorthy M, Asteris M, Papailiopoulos D, et al. XORing elephants: novel erasure codes for big data. Proc VLDB Endow, 2013, 6: 325–336
3 Dimakis A G, Godfrey P B, Wu Y, et al. Network coding for distributed storage systems. IEEE Trans Inform Theor, 2010, 56: 4539–4551
4 Balaji S B, Krishnan M N, Vajha M, et al. Erasure coding for distributed storage: an overview. Sci China Inf Sci, 2018, 61: 100301
5 Hou H, Lee P P C, Shum K W, et al. Rack-aware regenerating codes for data centers. IEEE Trans Inform Theor, 2019, 65: 4730–4745
6 Chen Z, Barg A. Explicit constructions of MSR codes for clustered distributed storage: the rack-aware storage model. IEEE Trans Inform Theor, 2020, 66: 886–899
7 Rashmi K V, Shah N B, Kumar P V. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. IEEE Trans Inform Theor, 2011, 57: 5227–5239
8 Tamo I, Barg A. A family of optimal locally recoverable codes. IEEE Trans Inform Theor, 2014, 60: 4661–4676