

# 异构部分重复码的构造<sup>①</sup>



孙 伟, 沈克勤, 张鑫楠, 何亚锦

(长安大学 信息工程学院, 西安 710064)

通讯作者: 孙 伟, E-mail: 1277874948@qq.com

**摘 要:** 针对分布式存储系统部分重复 (Fractional Repetition, FR) 码大都是同构的问题, 提出了基于 Hadamard 矩阵和基于 [7, 3, 4] 简单图形构造异构的 FR 码的两种新型构造设计算法, 构造方法更加简洁. 其中基于 Hadamard 矩阵构造存储容量异构的 FR 码可实现由同构经过简单变换为异构的编码方式; 基于 [7, 3, 4] 简单图形构造可扩展异构 FR 码可实现扩展延伸. 经过与 RS 码理论分析对比发现, 设计的两种异构 FR 码的修复局部性、修复带宽开销进一步降低, 且可以实现故障节点精确无编码修复, 修复复杂度较低, 修复效率较高, 减少了修复故障节点的时间.

**关键词:** 分布式存储; 部分重复码; 节点修复; Hadamard 矩阵; 异构

引用格式: 孙伟, 沈克勤, 张鑫楠, 何亚锦. 异构部分重复码的构造. 计算机系统应用, 2021, 30(2): 226-230. <http://www.c-s-a.org.cn/1003-3254/7793.html>

## Construction of Heterogeneous Fractional Repetition Codes

SUN Wei, SHEN Ke-Qin, ZHANG Xin-Nan, HE Ya-Jin

(School of Information Engineering, Chang'an University, Xi'an 710064, China)

**Abstract:** Aiming at the problem that Fractional Repetition (FR) codes in distributed storage system are mostly homogeneous, two new simpler algorithms are proposed to construct and design heterogeneous FR codes with different storage capacity based on Hadamard matrix and based on simple graphics. Heterogeneous FR codes with different storage capacity can be converted from homogeneous to heterogeneous based on Hadamard matrix. Extensible heterogeneous FR code based on simple graphics can be extended. It can be found through the comparison with theoretical analysis of RS codes that the designed heterogeneous FR codes have lower repair locality, repair bandwidth overhead, and repair complexity. And this method can realize accurate and non-coding repair of fault nodes at high repair efficiency, reducing the repair time of fault nodes.

**Key words:** distributed storage; Fractional Repetition (FR) codes; node repair; Hadamard matrix; heterogeneous

近些年, 随着信息技术和大数据产业的发展, 数据量激增, 传统集中存储设备已无法满足海量数据存储要求. 分布式存储系统 (DSS) 应运而生, DDS 是由许多廉价磁盘组成, 具有成本低、可用性強和可扩展性高等突出优势. 它作为可存储大量数据的存储系统已经被许多互联网企业应用, 例如 Microsoft 和 Facebook<sup>[1,2]</sup>. 然而分布式存储系统的节点易发生故障而造成数据丢失, 所以故障节点的修复成为研究的重点.

主要通过复制和编码来保证数据的可靠性. 传统的 DSS 多采用复制 (replication) 策略<sup>[3]</sup>, 其中三副本最为常见. 将文件复制 2 次即得到 3 个副本, 分别将其存储到系统的不同节点. 当有节点发生故障导致数据丢失, 可以通过其他正常节点的副本进行修复. 但是其占有的存储系统开销过于庞大. 于是 Dimakis 提出了纠删码 (erasure codes) 策略<sup>[4]</sup>. 与复制策略相比, 经典的纠删码具有更大的数据可靠性和较小的存储开销. 其中

① 基金项目: 陕西省自然科学基金 (2019JM-386)

Foundation item: Natural Science Foundation of Shaanxi Province (2019JM-386)

收稿时间: 2020-06-30; 修改时间: 2020-07-27; 采用时间: 2020-07-31; csa 在线出版时间: 2021-01-27

最大距离可分 (Maximum Distance Separable, MDS) 码获得最优的存储开销性能。但是纠错码修复故障节点时需要的修复带宽开销过大。由于上述缺陷, Dimakis 等<sup>[4]</sup> 开创性的提出并介绍了再生码, 研究发现在故障节点修复时其的修复带宽开销明显降低。但是再生码修复故障节点时需要大量有限域的运算, 计算复杂度大。

2010 年 Rouayheb 提出了部分重复 (Fractional Repetition, FR) 码, FR 码是一种精确最小带宽再生码, 修复故障节点时的修复带宽减小, 同时不需要大量有限域的运算, 计算复杂度较小<sup>[5,6]</sup>。所以越来越多研究人员开始研究 FR 码, Ramamoorthy 设计的 FR 码引入了组合设计的思想<sup>[7]</sup>。相继利用二部图<sup>[8]</sup>, Steiner 系<sup>[9]</sup> 和序列<sup>[10]</sup> 构造 FR 码。随后研究人员又研究了 FR 码的修复消耗最小化<sup>[11]</sup>。

部分重复码的现有编码方式节点的存储容量基本相同, 同时重复度也基本相同, 都属于同构编码方式, 但是分布式存储系统由于设备故障, 硬件差别等原因, 导致存储成本不同, 各个节点存储容量不同, 因此需要异构编码方式。本文提出基于 Hadamard 矩阵和 [7, 3, 4] 图形分别构造异构的部分重复码的一般算法。与现有的 FR 码相比, 采用 Hadamard 矩阵和 [7, 3, 4] 图形构造 FR 码更加简洁明了, 其中基于 Hadamard 矩阵可实现由同构经过简单变换为异构编码方式; 基于 [7, 3, 4] 图形构造 FR 码可实现扩展延伸, 若当前结构无法满足需求, 可对其进行扩展, 直至满足需求, 而且无需改变现有结构。经过与 RS 码理论分析对比发现, 设计的两种异构 FR 码的修复局部性、修复带宽开销进一步降低, 且可以实现故障节点精确无编码修复, 修复复杂度较低, 修复效率较高, 减少了修复故障节点的时间。

## 1 基础知识

### 1.1 Hadamard 矩阵

定义 1<sup>[12]</sup>。满足:  $H_n H_n^T = nI_n$ ;  $H_n$  是一个  $n$  阶方阵其由 1 或 -1 构成,  $I_n$  是一个  $n$  阶单位矩阵, 称  $H_n$  为  $n$  阶 Hadamard 矩阵。

Hadamard 矩阵具有如下性质:

(1) 将 Hadamard 矩阵的任意两行 (列) 交换, 矩阵的任意一行 (列) 的所有元素乘 -1, 得到的矩阵仍然是 Hadamard 矩阵。

(2) 若  $H_n$  是  $n$  阶 Hadamard 矩阵, 需要满足  $n$  是 4 的倍数 ( $n > 2$ )。

本文所需的 Hadamard 矩阵均为标准型, 即  $H_n$  的第 1 行和第 1 列全是 1。

### 1.2 部分重复码

定义 2<sup>[13]</sup>。FR 码。给定一个部分重复码  $C = (\partial, M)$ , 其中参数为  $(n, k, d)$ , 将重复度设为  $\rho$  (即编码数据块复制  $\rho$  倍), 特定地,  $M = \{M_1, \dots, M_n\}$  为  $n$  个子集的集合,  $M_i$  ( $1 \leq i \leq n$ ) 中的每一个元素都属于集合  $\partial = \{1, \dots, \theta\}$ 。

另外还需满足以下两个条件:

- 1) 每个子集的大小为  $d$ ;
- 2)  $\partial$  中每一个元素分别存在于  $M$  的  $\rho$  个子集中。

如果  $d$  和  $\rho$  分别都为固定值则为同构 FR 码, 如果  $d$  不是固定的值则为存储容量异构的 FR 码; 如果  $\rho$  不是固定的值则为重复度异构的 FR 码。

FR 码的本质为将经过 MDS 编码后的数据块复制  $\rho$  倍, 随后将其分别放置在  $n$  个不同节点中, 其中同一个的编码数据块不能出现在一个节点中。

## 2 基于 Hadamard 矩阵构造异构部分重复码

本节基于 Hadamard 矩阵构造存储容量不同的异构部分重复码。首先选一个  $4s$  ( $s = 1, 2, 3, \dots$ ) 阶的 Hadamard 矩阵, 再将此 Hadamard 矩阵进行简单变换即可得到所需矩阵。将编码数据块与矩阵进行类比联系, 分布式存储系统中的节点对应矩阵的行, 而不同的编码块对应于矩阵的列。根据 Hadamard 矩阵, 引出存储容量不同的异构 FR 码一般性构造算法, 其具体步骤如下:

步骤 1。首先采用  $(n, k)$  MDS 编码 ( $n \geq k$ ) 对原始文件进行编码, 得到  $n$  个编码数据块  $c_1, \dots, c_{k-1}, c_k, c_{k+1}, \dots, c_n$ , 其中包括  $k$  个原始数据块和  $n - k$  个校验数据块<sup>[6]</sup>;

步骤 2。取一个  $n + 1$  阶标准型 Hadamard 矩阵  $H_{n+1}$  ( $n + 1$  为 4 的倍数), 由式 (1) 对  $H_{n+1}$  进行简单变换:

$$K'_{n+1} = \frac{J_{n+1} + H_{n+1}}{2} \quad (1)$$

矩阵  $J_{n+1}$  中元素全为 1,  $H_{n+1}$  为标准 Hadamard 矩阵, 得到的  $K'_{n+1}$  ( $n \geq k$ ) 是一个 0-1 矩阵, 将  $K'_{n+1}$  的第一行和第一列同时删去得到所需矩阵  $K_n$ ;

步骤 3。将矩阵  $K_n$  中第  $j$  列出现的第  $((j + 1) \bmod \left(\frac{n-1}{2}\right))$  个 1 改为 0 得到新的矩阵  $K_{n1}$ , 然后计算式 (1):

$$N_i = \{j : a_{ij} = 1\} \quad (2)$$

其中,  $j = 1, 2, \dots, n$ ,  $i$  表示第  $i$  个 FR 节点,  $a_{ij}$  表示矩阵第  $i$  行第  $j$  列的值。其中表示异构 FR 码的存储节点, 将

矩阵 $K_{n1}$ 中第*i*行中全部的1所分别对应的列数提取出来,即可得到一个节点 $N_i$ 存储的数据块,得到节点存储容量不同的异构FR码。

具体的,选取一个12阶的Hadamard矩阵如图1所示,对其按步骤2操作得到11阶矩阵 $K_{11}$ 如图2所示,每一列的第 $(j+1) \bmod \left(\frac{n-1}{2}\right)$ 个1改为0得到新的矩阵 $K_{111}$ 如图3所示。随后节点对应矩阵的行,而不同的编码块对应于矩阵的列。所以我们得到了存储容量不同的异构的FR码,每个节点的存储结构如图4所示,其节点的存储容量 $d = 3, 4, 5$ ,编码块的重复度 $\rho = 4$ 。

$$H_{12} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & -1 \end{pmatrix}$$

图1 12阶Hadamard矩阵

$$K_{11} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

图2  $K_{11}$ 矩阵

$$K_{111} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

图3  $K_{111}$ 矩阵

单个节点发生故障时,可以根据存活的其他节点直接下载所需的数据,即可恢复故障节点。例如在图4中,若第二个节点 $N_2$ 发生故障,通过下载节点 $N_7$ 恢复数据块5和11,下载 $N_9$ 恢复数据块3和7,即可完成节点 $N_2$ 的恢复;同时也可以根据节点 $N_3, N_8, N_{11}$ 分别下载数据块3,5,7,11,也可完成故障节点 $N_2$ 的恢复。单节点

发生故障可采用多种方式来恢复,同时也实现故障节点的精确无编码修复。当两个节点发生故障时,方法同一个故障节点修复。

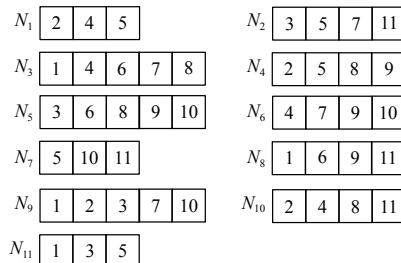


图4 存储容量异构的FR码节点存储结构图

### 3 可扩展的异构部分重复码

本小节提出一种基于[7, 3, 4]简单图形构造可扩展异构部分重复码的算法,此算法可以简单对部分重复码进行扩展,如现有结构不满足已有需求需要进行扩容,则不需破坏已有的结构,只需在图形尾部直接旋转拼接图形扩展,使得FR码可扩展,具体构造步骤如下所示:

步骤1. 首先采用 $(n, k)$ MDS编码( $n \geq k$ )对原始文件进行编码,得到*n*个编码数据块 $c_1, \dots, c_{k-1}, c_k, c_{k+1}, \dots, c_n$ ,其中包括*k*个原始数据块和*n-k*个校验数据块<sup>[6]</sup>;

步骤2. 取一个[7, 3, 4]简单图形,如图5所示。

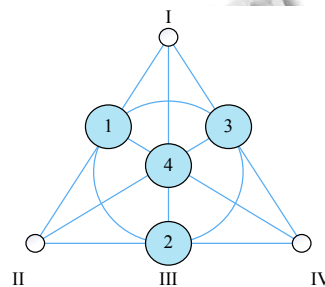


图5 [7, 3, 4]简单代码图形

步骤3. 通过[7, 3, 4]简单图形构造FR码:将外围3个顶点当作存储节点,将内部的4个顶点当作数据块,数据块按照顺时针存放,临近存储节点的3个数据块存放在该存储节点。

将[7, 3, 4]简单图形的外围3个顶点当作存储节点,将[7, 3, 4]简单图形内部的4个顶点当作数据块,数据块按照顺时针存放,临近存储节点的3个数据块存放在该存储节点。

步骤4. 通过将[7, 3, 4]简单图形旋转拼接构造可

扩展 FR 码:

1) 将扩展的[7, 3, 4]简单图形的外围罗马数字代表一个节点, 外围的第*i*个点表示分布式存储系统中的第*i*个存储节点 $N_i$ , 共有*n*个存储节点 ( $i = I, II, \dots, n$ ); 将与存储节点直接相连的数据块当作该存储节点存放的数据块, 即可得到存储容量和重复度异构的 FR 码。

2) 当数据块 $n = 3t + 4$ 时, 需要 $t + 1$ 个[7, 3, 4]简单图形旋转拼接, 构造出具有 $t + 3$ 个存储节点, *n*个数据块的 FR 码。当数据块 $n = 4 + 3t + s$  ( $s = 1, 2$ )时, 需要 $t + 2$ 个[7, 3, 4]简单图形旋转拼接, 构造出具有 $t + 3$ 个存储节点, *n*个数据块的 FR 码。

若构造一个具有 6 个存储节点, 13 个数据块的 FR 码。可将基本图形翻转拼接 3 次, 得到如图 6 所示图形, 按上述方法可得到 6 个存储节点所存放的数据块, 如图 7 所示, 若当前结构无法满足需求, 可对其进行扩展, 直至满足需求, 而且无需改变现有结构, 如图 8 所示。

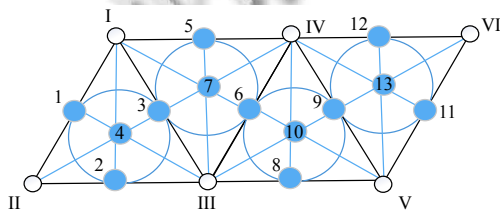


图 6 图形结构

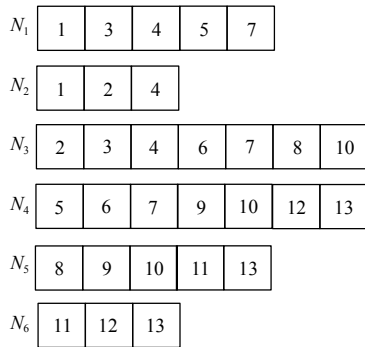


图 7 FR 码节点存储结构图

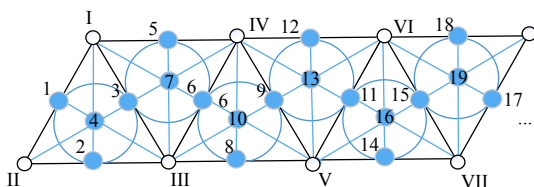


图 8 扩展图形结构

可以看出共有 6 个存储节点, 存储节点  $N_1$  和  $N_5$  的存储容量是 5, 存储节点  $N_3$  和  $N_4$  的存储容量是 7, 并

且重复度  $\rho$  为 2 或 3 的一个异构的 FR 码。

当单个节点发生故障时, 如图 7 中, 当第二个节点  $N_2$  发生故障时, 通过下载节点  $N_1$  恢复数据块 1 和 4, 下载  $N_3$  恢复数据块 2, 来完成节点  $N_2$  的恢复; 当节点  $N_3$  发生故障时, 通过下载节点  $N_1$  恢复数据块 3, 4 和 7, 下载  $N_2$  恢复数据块 2, 下载  $N_4$  恢复数据块 6 和 10, 下载  $N_5$  恢复数据块 8, 即可完成节点  $N_3$  的恢复。具体修复方式可选择性较大, 同时实现故障节点的精确无编码修复。

### 4 性能分析

本小节对基于 Hadamard 矩阵构造存储容量异构的部分重复码和基于[7, 3, 4]简单图形构造可扩展的异构部分重复码进行性能分析, 主要考虑修复带宽开销和修复局部性方面的性能, 并与常见的 RS 编码进行性能比较。为了便于比较, 基于 Hadamard 矩阵构造存储容量异构的部分重复码算法选取(11, 9)FR 码, 基于[7, 3, 4]简单图形构造可扩展异构部分重复码的算法选取(13, 9)FR 码, 取  $M = 1000$  Mbit。

#### 4.1 修复带宽开销

修复带宽开销指的是恢复失效节点时下载的数据量大小。例如采用(11, 9)RS 编码, 由于 RS 编码恢复故障节点需要下载全部原文件, 所以修复带宽开销为  $\eta = M$ ; 采用基于 Hadamard 矩阵构造存储容量不同的异构部分重复码构造(11, 9)FR 码时, 发生节点故障时的修复带宽开销为  $\eta = 3M/k, 5M/k$  或者  $7M/k$ 。为便于比较, 本文选取中间值作为代表值。在采用基于[7, 3, 4]简单图形构造可扩展异构部分重复码的算法构造(13, 9)FR 码时, 发生节点故障的修复带宽开销为  $\eta = 3M/k, 5M/k$  或者  $7M/k$ , 而采用(13, 9)RS 编码时, 发生节点故障的修复带宽开销为  $\eta = M$ 。由于图形特殊构造, 随着节点增多修复带宽开销基本都是  $7M/k$ , 所以选取其为代表值。以上 2 项数据与 RS 码仿真对比如图 9 所示。

经过对比不难发现本文提出的两种异构部分重复码构造的算法, 节点发生故障时修复带宽开销比 RS 编码明显降低。当节点数少于 16 时, 基于 Hadamard 矩阵构造的异构 FR 码修复带宽开销小于基于[7, 3, 4]简单图形构造异构 FR 码。但是当节点数逐渐增大, 基于[7, 3, 4]简单图形构造异构 FR 码的修复带宽开销小于基于 Hadamard 矩阵构造的异构 FR 码。

#### 4.2 修复局部性

发生节点故障时需要连接的存活节点数目称为修复局部性。当发生单节点故障时, (11, 9) RS 编码需要

连接9个节点恢复原文件用以修复故障节点,修复局部性为9;而基于Hadamard矩阵构造存储容量异构的FR码构造的(11,9)FR码,单节点故障时需要连接2个或者3个节点来修复,故修复局部性为2或者3。

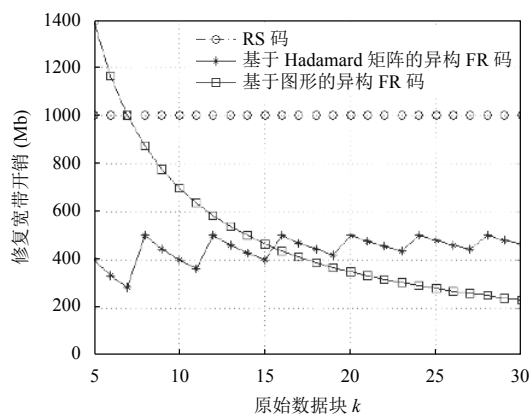


图9 修复带宽开销对比

采用基于[7, 3, 4]简单图形构造可扩展异构部分重复码的算法构造(13, 9)FR码时,发生节点故障时需要连接2, 3或者4个节点来恢复故障节点,所以修复局部性为2, 3或4;而(13, 9)RS码发生单节点故障时,RS码修复局部性为9。

分析发现修复单个故障节点时,基于Hadamard矩阵构造存储容量异构的FR码和基于[7, 3, 4]简单图形构造可扩展异构FR码的修复局部性,优于RS编码的修复局部性。但是[7, 3, 4]简单图形构造的异构FR码无法修复多节点故障,是下一步研究的方向。

## 5 结论

本文提出基于Hadamard矩阵和[7, 3, 4]图形分别构造异构的部分重复码的一般算法。与现有的FR码相比,采用Hadamard矩阵和[7, 3, 4]图形构造FR码更加简洁明了,其中基于Hadamard矩阵可实现由同构经过简单变换为异构编码方式;基于[7, 3, 4]图形构造FR码可实现扩展延升,若当前结构无法满足需求,可对其进行扩展,直至满足需求,且无需改变现有结构。经过与RS码理论分析对比发现,设计的两种异构FR码的修复局部性、修复带宽开销进一步降低,性能进一步提高。

### 参考文献

1 Huang C, Simitci H, Xu YK, *et al.* Erasure coding in windows azure storage. Proceedings of the 2012 USENIX Conference on Annual Technical Conference. Berkeley, CA, USA. 2012. 2.

2 Rashmi KV, Shah NB, Gu DK, *et al.* A “hitchhiker’s” guide to fast and efficient data reconstruction in erasure-coded data centers. Proceedings of the 2014 ACM Conference on SIGCOMM. Chicago, IL, USA. 2014. 331–342.

3 Liu Y, Vlassov V. Replication in distributed storage systems: State of the art, possible directions, and open issues. 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. Beijing, China. 2013. 225–232.

4 Dimakis AG, Godfrey PB, Wu YN. Network coding for distributed storage systems. IEEE Transactions on Information Theory, 2010, 56(9): 4539–4551. [doi: 10.1109/TIT.2010.2054295]

5 Rouayheb SE, Ramchandran K. Fractional repetition codes for repair in distributed storage systems. Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing. Monticello, IL, USA. 2010. 1510–1517.

6 王甜甜. 分布式存储系统中部分重复码构造研究 [硕士学位论文]. 西安: 长安大学, 2019.

7 Olmez O, Ramamoorthy A. Repairable replication-based storage systems using resolvable designs. Proceedings of 2012 50th Annual Allerton Conference on Communication, Control, and Computing. Monticello, IL, USA. 2012. 1174–1181.

8 Koo JC, Gill JT. Scalable constructions of fractional repetition codes in distributed storage systems. 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Monticello, IL, USA. 2011. 1366–1373.

9 Olmez O, Ramamoorthy A. Fractional repetition codes with flexible repair from combinatorial designs. IEEE Transactions on Information Theory, 2016, 62(4): 1565–1591. [doi: 10.1109/TIT.2016.2531720]

10 Benerjee KG, Gupta MK. On dress codes with flowers. 2015 Seventh International Workshop on Signal Design and its Applications in Communications (IWSDA). Bengaluru, India. 2015. 108–112.

11 Itani M, Sharafeddine S, Elkabbani I. Practical single node failure recovery using fractional repetition codes in data centers. 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA). Crans-Montana, Switzerland. 2016. 762–768.

12 Banica T, Nechita I. Almost hadamard matrices: The case of arbitrary exponents. Discrete Applied Mathematics, 2013, 161(16–17): 2367–2379.

13 Su YS. Pliable fractional repetition codes for distributed storage systems: Design and analysis. IEEE Transactions on Communications, 2018, 66(6): 2359–2375. [doi: 10.1109/TCOMM.2018.2799213]