DOI: 10.13382/j.jemi.B2104316

基于混合型正交表构造部分重复码*

王 静[1] 王相隆[1] 雷 珂[1] 田松涛[1] 刘向阳[2]
(1. 长安大学信息工程学院 西安 710064; 2. 国防科技大学信息通信学院 西安 710106)

摘 要:考虑到分布式存储系统中数据的存储和节点修复,提出一种基于混合型正交表的异构部分重复(fractional repetition, FR)码构造算法,并证明了该异构 FR 码是一般好的一般部分重复(generalized fractional repetition, GFR)码。利用混合型正交表中的水平对构造关联矩阵,根据关联矩阵对数据块在存储节点进行存放。另外,利用分组的方法在混合型正交表的基础上构造分组部分重复码,可以在局部组内实现单故障节点的精确无编码修复,修复局部性为 2 或 3,且能够对多个故障节点进行快速有效的修复。性能分析和实验仿真可知,所构造的分组 FR 码与 RS 码和简单再生码相比,在修复故障节点时具有较小的修复带宽开销和修复局部性,修复效率得到了提升。

关键词: 分布式存储;混合型正交表;部分重复码;局部修复
中图分类号: TP391; TN911.2 **文献标识码**: A **国家标准学科分类代码**: 520.5030

Construction of fractional repetition codes based on mixed orthogonal array

Wang Jing[1] Wang Xianglong[1] Lei Ke[1] Tian Songtao[1] Liu Xiangyang[2]
(1. School of Information Engineering, Chang'an University, Xi'an 710064, China; 2. College of Information and Communication, National University of Defense Technology, Xi'an 710106, China)

Abstract: For data storage and node repair in distributed storage systems, heterogeneous fractional repetition (FR) codes are constructed based on mixed orthogonal array. It is proved that the constructed heterogeneous FR codes are universally good generalized fractional repetition (GFR) codes. Concretely, the incidence matrix of FR codes is obtained by using the horizontal pairs in the mixed orthogonal array, and the data blocks are stored in the nodes of distributed storage systems. In addition, the grouping method is used to construct the grouping FR codes on the basis of the mixed orthogonal array, realizing the precise non-coding repair of a single fault node within the local repair group, and the repair locality is 2 or 3. Moreover, the grouping FR codes can repair multiple fault nodes quickly and efficiently. Performance analyses and experimental simulations show that, compared with RS codes and simple regeneration codes, the constructed grouping FR codes have lower repair bandwidth overhead and repair locality, and the repair efficiency is also improved.

Keywords: distributed storage; mixed orthogonal array; fractional repetition codes; local repair

0 引 言

随着信息技术的快速发展和智能设备的不断普及,互联网中数据量快速增长,分布式存储系统已经被广泛地使用和部署[1]。在实际的分布式存储系统中,节点故障时有发生,因此,在保证数据可以被恢复的条件下如何快速有效地修复故障节点是现今广大学者研究的热点。

当前分布式存储系统中故障节点主要通过复制和纠删码进行修复。复制策略是将原始文件复制多个副本,只要有一个数据副本保持完整,原始文件就可以被恢复。复制策略最常采用三副本复制,如谷歌文件系统(Google file systems, GFS)[2],但需要存储大量的副本才能保证数据的完整性。与复制策略相比,纠删码策略则通过编

收稿日期: 2021-05-18 Received Date: 2021-05-18
*基金项目:国家自然科学基金(62001059)、陕西省自然科学基金(2019JM-386)、陕西省重点研发计划项目(2021GY-019)资助

码增加冗余来提高存储的可靠性,实现故障节点中的数据修复。采用纠删码策略,如 RS(Reed-Solomon)码,修复故障节点需要重建整个文件,导致修复带宽开销过大[3],使得大量带宽和网络资源被占用。

文献[4]提出了再生码的概念,再生码将网络编码技术应用到分布式存储系统中,最大限度地减少了故障节点修复过程中的带宽开销。Rashmi 等[5]对再生码展开进一步的研究,得到节点存储开销和修复带宽开销之间的最优折中曲线,而达到曲线上两个极值点的再生码分别被称为最小带宽再生(minimum bandwidth regeneration, MBR)码和最小存储再生(minimum storage regeneration, MSR)码。虽然再生码能减少修复故障节点过程中的修复带宽开销,但会产生大量的磁盘 I/O 开销,且需要进行大量的有限域运算。针对上述问题,文献[6]提出了局部修复码(locally repairable codes, LRC)的概念。局部修复码连接较少的存活节点就可以修复故障节点,降低了故障节点的修复局部性。Kamath 等[7]将再生码和局部修复码结合,构造了一类局部再生码,分别称为局部最小存储再生码和局部最小带宽再生码。文献[8]基于局部最小存储再生码构造了一种局部协作再生码,可以通过相邻局部修复组来进行故障节点的修复。

文献[9]提出了一种更简单的 MBR 码的构造,无需编解码操作。该构造是由外部极大距离可分(MDS)码和内部重复码组成,并基于完全图设计,修复故障节点时需从 $d=\alpha$ 个存活节点中分别下载 $\beta=1$ 个数据块。进一步地,文献[10]提出了基于表的精确无编码修复策略,称为部分重复(fractional repetition, FR)码,可以降低修复故障节点过程中的修复带宽开销和计算复杂度。

FR 码采用双层编码结构,由外部的 MDS 码和内部的重复码组成。原文件首先经过确定的 MDS 码编码生成带有冗余的编码数据块,然后将编码数据块复制,并按照特定的分布规律排列在存储节点上,下载任意 k 个节点中的数据块可以重构原文件。近几年来,广大学者对 FR 码进行了深入的研究。Rouayheb 等[10]利用正则图和 Steiner 系构造了节点存储容量相同的 FR 码。文献[11]基于 FR 码构造了一种可局部修复的局部修复部分重复码,降低了故障节点的修复局部性。zhu 等[12-13]基于组合设计理论构造了适用于异构分布式存储系统的 FR 码,文献[14-17]分别讨论了 FR 码是一般好码的条件、FR 码的最优重构度和最优最小距离。Aydinian 等[18]基于半序集构造了一类具有一般好码性质的 FR 码,并将其拓展到异构分布式存储系统中。随后,Su 等[19-21]利用循环置换矩阵(CPMs)和仿射置换矩阵(APMs)构造了一类可以调节节点存储容量和重复度的 FR 码。上述 FR 码在多故障节点修复时的修复带宽开销和修复局部性较高,为此本文提出了基于混合型正交表构造异构 FR 码,所构造的异构 FR 码为一般好的一般部分重复(generalized fractional repetition, GFR)码,并在此基础上引入了分组的思想,提出了一种分组 FR 码的构造算法。基于混合型正交表构造的分组 FR 码可以对单个或多个故障节点进行修复,具体根据故障节点中数据块对应的水平对寻找存活节点并从中下载数据块来实现。分组的方法可实现故障节点在组内进行修复,可同时减小修复局部性和修复带宽开销。与 RS 码和简单再生码相比,基于混合型正交表的分组 FR 码在修复故障节点过程中的修复带宽开销和修复局部性较低,修复效率高。

1　混合型正交表和部分重复码

1.1　混合型正交表

定义 1[22] 设 $\boldsymbol{A}$ 是一个 $s\times l$ 矩阵,其任意一列元素都是由数(也称为水平)1,2,3,…所构成。如果 $\boldsymbol{A}$ 的任意两列中,同行元素所构成的水平对是一个完全对,且每对出现的次数相同时,则称 $\boldsymbol{A}$ 为一个正交表,记为 $L_s(t_1\times t_2\times\cdots\times t_l)$,其中 s 表示此表的行数,而 $t_1\times t_2\times\cdots\times t_l$ 表示此表有 l 列,并且第 1 列由 t_1 个水平组成,第 2 列由 t_2 个水平组成,…,第 l 列由 t_l 个水平组成。

正交表的性质如下。

1)正交表 $L_s(t_1\times t_2\times\cdots\times t_l)$ 的任意不同两列 p 和 q 所构成的水平对中,每个水平对都重复出现 s/t_pt_q 次。

2)正交表 $L_s(t_1\times t_2\times\cdots\times t_l)$ 中每列的各个水平出现的次数相同,如第 p 列中每个水平都重复出现 s/t_p 次。

在正交表 $L_s(t_1\times t_2\times\cdots\times t_l)$ 中,如果 $t_1=t_2=\cdots=t_l=t$,则可简记为 $L_s(t^l)$,称为 t 水平正交表,如果有任意两列或两列以上的水平数不相等时,则称为混合型正交表。

图 1(a)所示为 $L_{12}(6\times2^2)$ 混合型正交表,图 1(b)所示为 $L_8(4\times2^4)$ 混合型正交表。若取图 1(b)的混合型正交表 $L_8(4\times2^4)$ 中的第 1 列和第 2 列,所组成的水平对集{(1,1),(1,2),(2,1),(2,2),(3,1),(3,2),(4,1),(4,2)}是一个完全对,即第 1 列的水平数都会与第 2 列的水平数相匹配,称为搭配均匀。

1.2　FR 码

FR 码是由外部的 MDS 码和内部的重复码所构成。将一个文件分为大小相同的 m 个数据块,经(θ,m)MDS 码编码得到 $\boldsymbol{x}=\{x_1,\cdots,x_\theta\}$ 共 θ 个编码块,并复制 ρ 次得到 $\rho\theta$ 个数据块,ρ 称为重复度。把所有数据块按特定规律排列在 n 个节点 $N_i(i=1,2,\cdots,n)$ 上,每个节点存储 α 个数据块,则可得 $n\alpha=\rho\theta$。

定义 2[14] 一个 FR 码 $C:(n,\theta,\alpha,\rho)$ 是由一个字符集 $[\theta]:=\{1,2,\cdots,\theta\}$ 上的 n 个子集 $\{N_1,N_2,\cdots,N_n\}$ 组成的集合簇,并满足 $|N_i|=\alpha$,其中 $i=1,2,\cdots,n$;$[\theta]$ 中的每个

$$\begin{pmatrix} 2&1&1\\ 5&1&2\\ 5&2&1\\ 2&2&2\\ 4&1&1\\ 1&1&2\\ 1&2&1\\ 4&2&2\\ 3&1&1\\ 6&1&2\\ 6&2&1\\ 3&2&2 \end{pmatrix} \qquad \begin{pmatrix} 1&1&1&1&1\\ 1&2&2&2&2\\ 2&1&1&2&2\\ 2&2&2&1&1\\ 3&1&2&1&2\\ 3&2&1&2&1\\ 4&1&2&2&1\\ 4&2&1&1&2 \end{pmatrix}$$

(a) $L_{12}(6\times 2^2)$ (b) $L_8(4\times 2^4)$

图 1 混合型正交表 $L_{12}(6\times 2^2)$ 和 $L_8(4\times 2^4)$

Fig. 1 Mixed orthogonal array$L_{12}(6\times 2^2)$ and $L_8(4\times 2^4)$

元素属于集合$\{N_1,N_2,\cdots,N_n\}$中的ρ个子集。

在 FR 码 $C:(n,\theta,\alpha,\rho)$中,当任意一个节点故障时,可以从 $d=\alpha$ 个存活节点中分别下载 $\beta=1$ 个数据块来修复该故障节点,而任意 k 个节点中的所有数据块可以重构原文件,k 称为重构度。如果 FR 码中数据块的重复度 ρ 不同或节点存储容量 α 不同,则称这个 FR 码为异构 FR 码,也称为 GFR 码[23]。定义异构 FR 码的重复度为 ρ_j $(j=1,2,\cdots,\theta)$,节点存储容量为 $\alpha_i(i=1,2,\cdots,n)$,根据 $n\alpha=\rho\theta$ 可得:

$$\sum_{i=1}^{n}\alpha_i=\sum_{j=1}^{\theta}\rho_j \tag{1}$$

根据定义 2 可将 FR 码的节点和数据块的关系用矩阵的形式来表示,称为 FR 码的关联矩阵,用矩阵 $\boldsymbol{G}$ 来表示。

定义 3[19]一个 FR 码 C,其关联矩阵 $\boldsymbol{G}=[g_{ij}]_{n\times\theta}$ 为一个 0-1 矩阵$(0<i\leqslant n,0<j\leqslant\theta)$,则:

$$g_{ij}=\begin{cases}1, & x_j\in N_i\\ 0, & x_j\notin N_i\end{cases} \tag{2}$$

FR 码的关联矩阵 $\boldsymbol{G}$ 中,第 i 行中元素 1 的个数为 α_i,表示第 i 个节点中数据块的个数;第 j 列中元素 1 的个数为 ρ_j,表示第 j 个数据块的重复度。

图 2 所示为(7,6,2;3,3)异构 FR 码,其中共有 7 个节点,6 个数据块,每个数据块的重复度为 3,节点的存储容量为 2 和 3。式(3)为(7,6,2;3,3)异构 FR 码的关联矩阵 $\boldsymbol{G}_{7\times 6}$,矩阵 $\boldsymbol{G}_{7\times 6}$ 的行对应异构 FR 码的节点,列对应异构 FR 码的数据块。

$$\boldsymbol{G}_{7\times 6}=\begin{pmatrix} 1&1&0&0&0&0\\ 0&0&1&1&0&0\\ 0&0&0&0&1&1\\ 1&0&1&0&1&0\\ 0&1&1&0&0&1\\ 1&0&0&1&0&1\\ 0&1&0&1&1&0 \end{pmatrix} \tag{3}$$

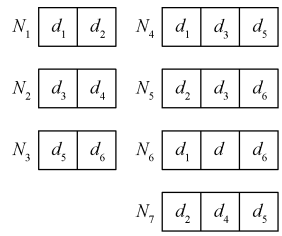

图 2 (7,6,2;3,3)异构 FR 码

Fig. 2 (7,6,2;3,3) heterogeneous FR code

从 FR 码 C 中的任意 k 个节点下载其中的数据块可以重构原文件,根据(θ,m)MDS 码的性质,k 个节点中不同数据块的个数不小于 m。定义任意 k 个节点中不同数据块的最小数量为 $M(k)$,则需满足 $m\leqslant M(k)$,即 $M(k)$ 为 FR 码在取重构度 k 时原文件分解的最大数据块数,因此原文件分解的数据块数 m 取决于重构度 k。这里称 $M(k)$ 为 FR 码的最大文件尺寸[10],则:

$$M(k)=\min_{K\subset[n],|K|=k}|\cup_{i\in K}N_i| \tag{4}$$

如图 2 所示的(7,6,2;3,3)异构 FR 码,若取 $k=3$,由式(4)可得 $M(3)=5$。首先可将文件分为 5 个数据块,经(6,5)MDS 码编码得到 6 个编码块,将编码块复制 $\rho=3$ 次得到 18 个数据块,并按照图 2 所示将所有数据块排列在 7 个节点中,任取 $k=3$ 个节点下载其中的数据块可以重构原文件。

2 基于混合型正交表构造 FR 码

本小节基于混合型正交表构造异构 FR 码,所构造的异构 FR 码有 $n=t_1$ 个节点,$\theta=t_1t_2-t_2(t_2+1)/2$ 个数据块,每个数据块的重复度为 $\rho=2$,节点存储容量有两种,分别为 $\alpha'_1=t_1-1$ 和 $\alpha'_2=t_2$。选择有两种不同水平的混合型正交表 $L_s(t_1^p\times t_2^q)$ 且满足 $t_1>t_2\geqslant 2$,通过选取水平不同的任意两列所组成的完全对来构造异构 FR 码的关联矩阵,进而构造异构 FR 码。由满足上述条件的混合型正交表引出构造异构 FR 码的一般性构造算法,具体步骤如下。

1)选择一个有两种不同水平的混合型正交表 $L_s(t_1^p\times t_2^q)$ 且满足 $t_1>t_2\geqslant 2$,取水平数不同的任意两列组成完全对$([t_1],[t_2])$。

2)在步骤 1)得到的完全对$([t_1],[t_2])$中筛选满足如下条件的水平对:

$$B=\{(x,y)\mid x\in[t_1],y\in[t_2],x>y\} \tag{5}$$

3)定义矩阵 $\boldsymbol{G}=[g_{ij}]_{n\times\theta}$,将水平数 $t_1=n$ 作为矩阵 $\boldsymbol{G}$ 的行,B 中的水平对作为矩阵 $\boldsymbol{G}$ 的列,记第 j 个水平对为 (x_j,y_j)。若 $x_j=i$,则记 $g_{ij}=1$,y_j 亦如此,列中其他位置记为 0。

4)通过关联矩阵 $\boldsymbol{G}$ 及式(6)来构造异构 FR 码:

$$N_i = \{d_j \mid g_{ij} = 1\} \tag{6}$$

即任意节点 N_i 中所存储的数据块 d_j 为第 i 行中元素 1 所在列对应的数据块。

将关联矩阵 $\boldsymbol{G}$ 的行作为节点 $\{N_1, N_2, \cdots, N_n\}$,则共有 $t_1 = n$ 个节点,且行重为 $t_1 - 1$ 或 t_2,即节点的存储容量为 $t_1 - 1$ 或 t_2;将关联矩阵的列作为存储的数据块,即每个水平对对应一个数据块。因此,所构造的 FR 码为($n = t_1, \theta = t_1 t_2 - t_2(t_2+1)/2, \alpha'_1 = t_1 - 1; \alpha'_2 = t_2, \rho = 2$)异构 FR 码。

$$\boldsymbol{G}_{6\times 9} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{7}$$

选取图 1(a)的混合型正交表 $L_{12}(6\times 2^2)$构造 FR 码,表中 $t_1 = 6, t_2 = 2$。根据算法中步骤 1)和 2)可得 $B = \{(2,1),(3,1),(3,2),(4,1),(4,2),(5,1),(5,2),(6,1),(6,2)\}$,其中有 9 个水平对,则 $\theta = 9$;再经过步骤 3)转换为矩阵 $\boldsymbol{G}_{6\times 9}$,如式(7)所示;最后按照步骤 4)可得(6,9,5;2,2)异构 FR 码,其节点存储结构如图 3 所示。

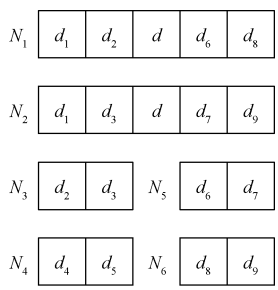

图 3　(6,9,5;2,2)异构 FR 码

Fig. 3　(6,9,5;2,2) heterogeneous FR code

在基于混合型正交表的异构 FR 码中,当任意一个节点发生故障时,可以通过故障节点中数据块对应的水平对来修复该故障节点。在 FR 码的关联矩阵中,不同水平对对应不同的数据块,水平对中的所有元素对应该数据块存放的节点下标。当任意一个节点发生故障时,找到故障节点中每个数据块对应的水平对,水平对中所共有的元素为该故障节点的下标 i,其他元素为修复该故障节点时需要连接的节点的下标,因此从这些节点中下载水平对对应的数据块即可修复故障节点。整个修复过程是精确的数据块传输过程,不需要编解码操作。

图 3 中(6,9,5;2,2)异构 FR 码中,当节点 N_3 发生故障,考虑到节点中数据块 d_2 和 d_3 所对应的水平对为(3,1)和(3,2),两个数对中共有的元素为 3,即故障节点 N_3 的下标,而数对中除元素 3 之外还有 1 和 2,那么只需从下标为 1 和 2 的存活节点 N_1 和 N_2 中分别下载水平对对应的数据块 d_2 和 d_3,即可修复故障节点 N_3。

若 $t_1 = t_2 + 1$,利用上述算法所构造的 FR 码为同构 FR 码,即所有节点的存储容量相同。本文主要构造异构 FR 码,部分利用常用混合型正交表构造的异构 FR 码的相关参数如表 1 所示。

表 1　不同混合型正交表构造的异构 FR 码的相关参数

Table 1　Related parameters of heterogeneous FR codes constructed by different mixed orthogonal arrays

	n	θ	α_i
$L_8(4\times 2^4)$	4	5	2,2,3,3
$L_{20}(5\times 2^8)$	5	7	2,2,2,4,4
$L_{18}(6\times 3^6)$	6	12	3,3,3,5,5,5
$L_{32}(8\times 4^8)$	8	22	4,4,4,4,7,7,7,7
$L_{27}(9\times 3^9)$	9	21	3,3,3,3,3,3,8,8,8

3　基于混合型正交表构造分组 FR 码

由表 1 可知,选取的混合型正交表水平数较大时,节点的存储开销也随之增大,因此当某些节点发生故障时,修复带宽开销也会增大。为了降低故障节点的修复带宽开销,可利用分组的方法对原始数据块进行分组,再在组内构造 FR 码。分组构造使故障节点只需在组内连接少量存活节点就可修复,可同时降低故障节点的修复带宽开销和修复局部性。利用水平数最小的混合型正交表来构造异构 FR 码,即图 1(b)的混合型正交表 $L_8(4\times 2^4)$,并结合分组的方法构造分组 FR 码。

3.1　基于混合型正交表构造分组 FR 码

将原文件分解为若干个数据块并分组存储在若干个局部修复组中,每个局部修复组中的节点存储的都是由混合型正交表 $L_8(4\times 2^4)$ 所构造的 (4,5,3;2,2) 异构 FR 码,具体的构造算法如下。

1)选取混合型正交表 $L_8(4\times 2^4)$,表中 $t_1 = 4, t_2 = 2$,根据基于混合型正交表的异构 FR 码构造算法中步骤 1)~3)得到矩阵 $\boldsymbol{G}_{4\times 5}$,如式(8)所示。

2)原始文件分为大小相同的 m 个数据块,将这 m 个数据块分组得到 g 组,每组中有 3 个数据块,则有如下 3 种情况。

(1)若 m 可以被 $c = 3$ 整除,此时 $m = cg$,则在每个分组中采用(5,3)MDS 码编码,然后利用矩阵 $\boldsymbol{G}_{4\times 5}$ 构造分组 FR 码。

(2)若 m 不能被 $c = 3$ 整除,即 $m = cg + r$,且 $r = 1$,则在前 $g-1$ 个分组中采用(5,3)MDS 码编码,第 g 个分组采用(5,4)MDS 码编码,最后将所有分组中的编码块根据矩阵 $\boldsymbol{G}_{4\times 5}$ 构造分组 FR 码。

(3)若 m 不能被 $c=3$ 整除,即 $m=cg+r$,且 $r=2$,则在前 $g-2$ 个分组中采用(5,3)MDS 码编码,第 $g-1$ 个分组和第 g 个分组采用(5,4)MDS 码编码,最后同样将所有分组中的编码块根据矩阵 $\boldsymbol{G}_{4\times5}$ 构造分组 FR 码。

$$\boldsymbol{G}_{4\times5}=\begin{pmatrix}1&1&0&1&0\\1&0&1&0&1\\0&1&1&0&0\\0&0&0&1&1\end{pmatrix} \tag{8}$$

若原文件分为 $m=10$ 个数据块,$m=3g+1$,满足上述第 2 种情况,且共有 3 个局部修复组。第 1 个和第 2 个局部修复组采用(5,3)MDS 码编码,第 3 个局部修复组采用(5,4)MDS 码编码,最后将 3 个分组中的编码数据块分别根据矩阵 $\boldsymbol{G}_{4\times5}$ 构造分组 FR 码。构造得到的分组 FR 码的节点存储结构如图 4 所示。

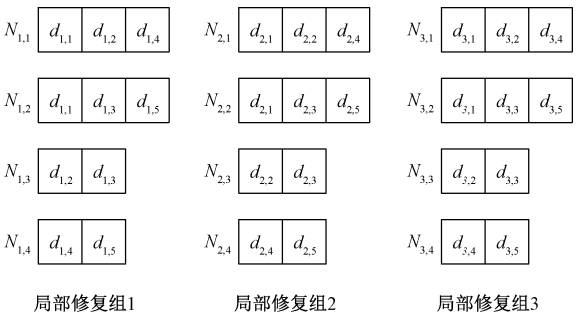

图 4 分组 FR 码的节点存储结构

Fig. 4 Node storage construction of grouping FR code

3.2 故障节点修复

由于局部修复组中 FR 码的重复度 $\rho=2$,所以局部修复组内最多可以允许两个节点故障,修复过程如下。

1)当只有一个节点发生故障时,可以按照第 2 节中故障节点的修复方法在局部修复组内进行修复,即通过故障节点中数据块对应的水平对来寻找存活节点,并从中分别下载水平对所对应的数据块来完成修复。

2)当两个节点发生故障时,有如下两种情况:

(1)若两个故障节点在不同的局部修复组中,则需要分别在这两个局部修复组中修复故障节点,具体修复过程如过程 1)。

(2)若两个故障节点在同一个局部修复组中,如果两个故障节点没有相同的数据块,则可按照过程 1)中的修复方法分别对故障节点进行修复;如果两个故障节点有相同的数据块,则需要从其余所有存活节点中下载数据块来恢复组内全部数据,再进行故障节点的修复。如图 4 所示的分组 FR 码,当第 3 个局部修复组中节点 $N_{3,2}$ 和 $N_{3,4}$ 发生故障,则需要从剩下的节点 $N_{3,1}$ 和 $N_{3,3}$ 中下载数据块恢复组内全部数据,从而修复故障节点 $N_{3,2}$ 和 $N_{3,4}$。

4 性能分析

对基于混合型正交表构造的异构 FR 码及分组 FR 码的性能进行分析,包括异构 FR 码的最大文件尺寸 $M(k)$、分组 FR 码的修复局部性和修复带宽开销,并将分组 FR 码与 RS 码和简单再生码作比较,通过仿真实验得出结论。

4.1 异构 FR 码的最大文件尺寸 $M(k)$

由于 FR 码外部采用 MDS 码,原文件分解的数据块数取决于重构度 k。定义任意 k 个节点中不同数据块的个数为 $m(k)$。

定理 1 在基于混合型正交表的异构 FR 码中,将节点按照不同的存储容量分为 N'_1 和 N'_2 两部分,两部分的存储容量分别为 $\alpha'_1=t_1-1$ 和 $\alpha'_2=t_2$,则任意 k 个节点所能得到的不同数据块数为:

$$m(k)=k_1\alpha'_1-\binom{k_1}{2}+k_2\alpha'_2-k_1k_2 \tag{9}$$

式中:k_1 表示从 N'_1 中取得的节点个数,k_2 表示从 N'_2 中取得的节点个数,且满足 $k_1+k_2=k$。

证明:节点 N'_1 部分中任意两个节点有一个相同的数据块,因此任意 k_1 个节点中有 $k_1\alpha'_1-\binom{k_1}{2}$ 个不同的数据块;节点 N'_2 部分中任意两个节点没有相同的数据块,则任意 k_2 个节点中有 $k_2\alpha'_2$ 个不同的数据块;节点 N'_1 部分中的任意一个节点和 N'_2 部分中任意一个节点都有一个相同的数据块,则 k_1 个节点和 k_2 个节点中有 k_1k_2 个相同的数据块。综上可得,任意 k 个节点中有 $k_1\alpha'_1-\binom{k_1}{2}+k_2\alpha'_2-k_1k_2$ 个不同的数据块。

由式(4)可知,任意 k 个节点中不同数据块的最小数量为 FR 码的最大文件尺寸 $M(k)$,则基于混合型正交表的异构 FR 码的最大文件尺寸为:

$$M(k)=\min_{K\subset[n],|K|=k_1+k_2}\left\{k_1\alpha'_1-\binom{k_1}{2}+k_2\alpha'_2-k_1k_2\right\} \tag{10}$$

文献[13,23]给出了异构 FR 码中最大文件尺寸的限。具体地,在节点存储容量不同的异构 FR 码中,任意两个节点中相同的数据块数至多为 1,假设节点的存储容量 α_i 满足 $\alpha_1\leqslant\alpha_2\leqslant\cdots\leqslant\alpha_n$,则异构 FR 码的最大文件尺寸 $M(k)$ 满足式(11)。

$$\left(\sum_{i=1}^{k}\alpha_i\right)-\binom{k}{2}\leqslant M(k)\leqslant\sum_{i-1}^{k}\alpha_i \tag{11}$$

那么称该异构 FR 码为一般好的 GFR 码。

定理 2 在基于混合型正交表的异构 FR 码中,假设

节点的存储容量 α_i 满足 $\alpha_1 \leqslant \alpha_2 \leqslant \cdots \leqslant \alpha_n$，则该异构 FR 码的最大文件尺寸 $M(k)$ 满足式(11)，因此基于混合型正交表的异构 FR 码是一般好的 GFR 码。

证明：在基于混合型正交表的异构 FR 码中，任意两个数据块所对应的水平对相比，若没有相同元素，则两个数据块不会同时出现在任意一个节点中；若有一个相同元素，则两个数据块只会同时出现在一个节点中。因此，任意两个数据块至多同时出现在一个节点中，即任意两个节点中相同的数据块数至多为 1，那么任意 k 个节点中至多有 $\binom{k}{2}$ 个重复的数据块。若 α_i 满足 $\alpha_1 \leqslant \alpha_2 \leqslant \cdots \leqslant \alpha_n$，则 k 个节点中至少有 $(\sum_{i=1}^{k}\alpha_i)-\binom{k}{2}$ 个不同的数据块，至多 $\sum_{i=1}^{k}\alpha_i$ 个数据块。

如图 3 所示的基于混合型正交表 $L_{12}(6\times 2^2)$ 构造(6,9,5;2,2)异构 FR 码，根据式(10)可得该异构 FR 码在不同重构度 k 时的最大文件尺寸为：

$$M(k)=\begin{cases}2, & k=1\\ 4, & k=2\\ 6, & k=3\\ 8, & k=4\\ 9, & k=5,6\end{cases} \tag{12}$$

若令 $k=4$，由式(12)得 $M(4)=8$。可将原文件分为 8 个数据块，经(9,8)MDS 码编码得到 9 个数据块并复制 $\rho=2$ 次得到 18 个数据块；然后排列在 6 个节点中，任取 $k=4$ 个节点下载数据块都可以重构原文件。同时，当 $k=4$ 时异构 FR 码的最大文件尺寸也满足式(11)。事实上，对于任意 $k\leqslant n$ 都满足式(11)，因此该异构 FR 码是一般好的 GFR 码。

4.2　修复局部性

修复故障节点时所连接的存活节点数称为故障节点的修复局部性。分别采用 RS 码、简单再生码和本文构造的基于混合型正交表的分组 FR 码来存储大小相同的文件。当只有一个节点发生故障时，若采用 (n,m) RS 码，需要恢复原文件才能修复故障节点，所以修复局部性为 m；若采用 (n,m,f) 简单再生码，则需要 $2f$ 个存活节点来修复故障节点，若取 $f=4$，则修复局部性为 8；如果采用基于混合型正交表的分组 FR 码，故障节点在组内进行修复，修复局部性为 2 或 3，计算组内每个节点的修复局部性，取平均值得(3+3+2+2)/4=2.5。

若有两个节点同时发生故障，采用 (n,m) RS 码同样需要恢复原文件，所以修复局部性同样为 m；对于 (n,m,f) 简单再生码，只有满足一定条件才能通过存活节点来修复两个故障节点，否则需要先恢复原文件，修复局部性为 m；而对于基于混合型正交表的分组 FR 码，如果两个故障节点在两个不同的局部修复组内，则分别在这两个局部修复组内对故障节点进行修复，修复局部性最小为 4，最大为 6；若两个故障节点在同一个局部修复组内，如果两个故障节点没有相同的数据块，修复局部性为 2，如果两个故障节点有相同的数据块，则首先需要恢复组内所有数据，修复局部性也为 2。综上所述，当两个节点发生故障时，RS 码和简单再生码都需要先连接 m 个存活节点恢复原文件进而修复两故障节点，而基于混合型正交表的分组 FR 码只需连接故障节点所在局部修复组中的若干存活节点就能实现两故障节点的修复，因此其修复局部性优于 RS 码和简单再生码。

图 5 所示为采用 3 种编码方式对同一文件分解的 m 个数据块编码后，修复单故障节点的平均修复局部性的比较，其中基于混合型正交表的分组 FR 码的平均修复局部性明显低于 RS 码和简单再生码。

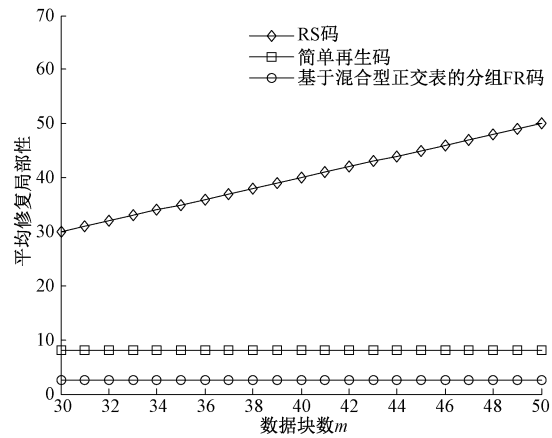

图 5　单故障节点的平均修复局部性对比

Fig. 5　Comparison of average repair locality for single failure node

4.3　修复带宽开销

修复带宽开销是指修复故障节点时从存活节点下载数据量的大小。假设一个文件的大小为 $M=500$ Mb，当只有一个节点发生故障时，若采用 (n,m) RS 码，则需要恢复原文件，修复带宽开销为 M；对于 (n,m,f) 简单再生码，每个节点中存储 $f+1$ 个数据块，且每个数据块的大小为 M/fm，而修复一个数据块需要下载 f 个数据块来修复，所以修复带宽开销为 $(f+1)M/m$，若取 $f=4$，则修复带宽开销为 $5M/m$；如果采用基于混合型正交表的分组 FR 码，每个数据块的大小为 M/m，由于局部修复组内节点所存储数据块的数量不同，所以单故障节点的修复带宽开销为 $2M/m$ 或 $3M/m$，计算每个节点的修复带宽开销并取平均值为 $(3+3+2+2)M/(4m)=5M/(2m)$。

若有两个节点同时发生故障，对于 (n,m) RS 码，仍然需要恢复原文件，修复带宽开销仍为 M；如果采用 $(n,$

m,f)简单再生码,若两个故障节点中有相同的数据信息,则需要先恢复原文件,进而修复两个故障节点,修复带宽开销为 M,若两个故障节点中没有相同的数据信息,修复带宽开销为 $2(f+1)M/m$,当 $f=4$ 时为 $10M/m$;如果采用基于混合型正交表的分组 FR 码,若两个故障节点在两个局部修复组中,则分别在两个局部修复组中进行修复,修复带宽开销最大为 $6M/m$,最小为 $4M/m$,若两个故障节点在同一局部修复组中,如果两个故障节点没有相同的数据块,则修复带宽开销为 $4M/m$,如果两个故障节点有相同的数据块,则首先需要恢复组内全部数据,修复带宽开销为 $3M/m$ 或 $4M/m$。由上述可得,对于两故障节点的修复,RS 码和简单再生码都需要大量的修复带宽开销才能实现,而基于混合型正交表的分组 FR 码只需在故障节点所在的局部修复组中连接存活节点并下载数据块即可完成两故障节点的修复,因此其修复带宽开销优于 RS 码和简单再生码。

在图 6 所示的 3 种编码方式修复单故障节点时的平均修复带宽开销中,基于混合型正交表的分组 FR 码的平均修复带宽开销要优于 RS 码和简单再生码。

表 2 为 RS 码、简单再生码和本文基于混合型正交表构造的分组 FR 码在故障节点修复时的性能比较。可以看到,基于混合型正交表的分组 FR 码在修复故障节点时的修复局部性和修复带宽开销较低,修复效率较高。

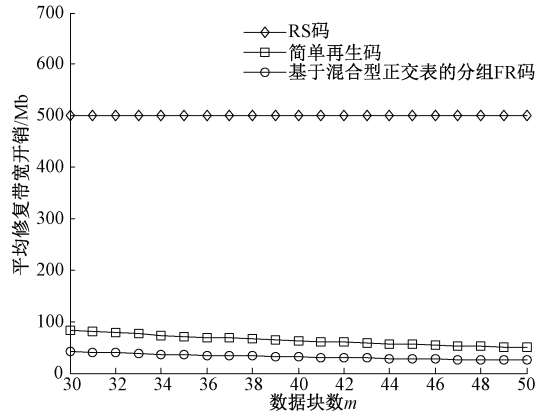

图 6 单故障节点的平均修复带宽开销对比

Fig. 6 Comparison of average repair bandwidth cost for single failure node

表 2 三种编码方式修复故障节点的性能比较

Table 2 Comparison of the performance of the three coding methods to repair the failed nodes

		RS 码	简单再生码($f\geq 2$)	基于混合正交表的分组 FR 码	
修复局部性	单节点故障	m	$2f$	2 或 3	
	两节点故障	m	m	一个修复组中	2
				两个修复组中	4 或 5 或 6
修复带宽开销	单节点故障	M	$(f+1)M/m$	$2M/m$ 或 $3M/m$	
	两节点故障	M	M 或 $2(f+1)M/m$	一个修复组中	$3M/m$ 或 $4M/m$
				两个修复组中	$4M/m$ 或 $5M/m$ 或 $6M/m$

5 结 论

本文提出一种基于混合型正交表的异构 FR 码构造方法,与传统 FR 码基于表的故障节点修复方式相比,该 FR 码是通过构造过程中的水平对来实现的。对于目前分布式存储系统中修复故障节点的修复带宽开销较高、修复局部性较大等问题,本文还在基于混合型正交表的异构 FR 码的基础上构造了分组 FR 码,其可以在局部修复组内对故障节点进行修复。理论分析证明基于混合型正交表的异构 FR 码是一般好的 GFR 码,进一步与 RS 码以及简单再生码相比,构造的分组 FR 码可以通过连接较少的存活节点来实现故障节点的修复,且修复带宽开销相对较低。

参考文献

[1] SIDDIQA A, KARIM A, GANI A. Big data storage technologies: A survey [J]. Frontiers of Information Technology & Electronic Engineering, 2017, 18(8): 1040-1070.

[2] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google file system [J]. ACM SIGOPS Operating Systems Review, 2003, 37(5): 29-43.

[3] WANG Y J, XU F L, PEI X Q. Research on erasure code-based fault-tolerant technology for distributed storage [J]. Chinese Journal of Computers, 2017, 40(1): 238-257.

[4] DIMAKIS A G, GODFREY P B, WU Y, et al. Network coding for distributed storage systems [J]. IEEE Transactions on Information Theory, 2010, 56(9): 4539-4551.

[5] RASHMI K V, SHAN N B, KUMAR P V. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction[J]. IEEE Transactions on Information Theory, 2011, 57(8):5227-5239.

[6] PAPAILIOPOULOS D S, DIMAKIS A G. Locally repairable codes[J]. IEEE Transactions on Information

Theory, 2014, 60(10): 5843-5855.
[7] KAMATH G M, PRAKASH N, LALITHA V, et al. Codes with local regeneration and erasure correction[J]. IEEE Transactions on Information Theory, 2014, 60(8): 4637-4660.
[8] WANG J, YAN Z Y, LI K C, et al. Local codes with cooperative repair in distributed storage of cyber-physical-social systems [J]. IEEE Access, 2020, 8: 38622-38632.
[9] SHAH N B, RASHMI K V, KUMAR P V, et al. Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff[J]. IEEE Transactions on Information Theory, 2012,DOI: 10. 1109/TIT. 2011. 2173792.
[10] ROUAYHEB S E, RAMCHANDRAN K. Fractional repetition codes for repair in distributed storage systems[C]. 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2010: 1510-1517.
[11] NAM M, KIM J, SONG H. Locally repairable fractional repetition codes[C]. 7th International Workshop on Signal Design and its Applications in Communications (IWSDA), 2015: 128-132.
[12] ZHU B, SHUM K W, LI H, et al. General Fractional Repetition Codes for Distributed Storage Systems[J]. IEEE Communications Letters, 2014, 18(4): 660-663.
[13] ZHU B, SHUM K W, LI H. Heterogeneity-aware codes with uncoded repair for distributed storage systems[J]. IEEE Communications Letters, 2015, 19(6): 901-904.
[14] ZHU B. A study on universally good fractional repetition codes [J]. IEEE Communications Letters, 2018, 22(5): 890-893.
[15] ZHU B, SHUM K W, LI H, et al. On the optimal reconstruction degree of fractional repetition codes[C]. IEEE International Symposium on Information Theory (ISIT), 2019: 1557-1561.
[16] ZHU B, SHUM K W, LI H. Fractional repetition codes with optimal reconstruction degree [J]. IEEE Transactions on Information Theory, 2020, 66(2): 983-994.
[17] ZHU B, SHUM K W, WANG W, et al. On the optimal minimum distance of fractional repetition codes[C]. IEEE Global Communications Conference, 2020: 1-6.
[18] AYDINIAN H, BOCHE H. Fractional repetition codes based on partially ordered sets[C]. IEEE Information Theory Workshop (ITW), 2017: 51-55.
[19] SU Y S. Pliable fractional repetition codes for distributed storage systems: Design and analysis [J]. IEEE Transactions on Communications, 2018, 66(6): 2359-2375.
[20] SU Y S. Optimal pliable fractional repetition codes[C]. 2018 IEEE International Symposium on Information Theory (ISIT), 2018: 2077-2081.
[21] SU Y S. Optimal pliable fractional repetition codes that are locally recoverable: A bipartite graph approach[J]. IEEE Transactions on Information Theory, 2019, 65(2): 985-999.
[22] 杨子胥. 正交表的构造[M]. 济南: 山东人民出版社, 1978.
YANG Z X. The Construction of Orthogonal Array[M]. Jinan: Shandong People's Publishing House, 1978.
[23] GOPAL K. On Heterogeneous distributed storage systems: Bounds and code constructions [D]. Gandhinagar: Dhirubhai Ambani Institute of Information and Communication Technology, 2019.

作者简介

王静,2004 年于西安电子科技大学获得学士学位,2009 年于西安电子科技大学获得博士学位,现为长安大学教授,主要研究方向为分布式存储、再生码和局部性修复编码。
E-mail:jingwang@ chd. edu. cn

Wang Jing received her B. S. and Ph. D. degrees from Xidian University in 2004 and 2009, respectively. She is currently a professor at Chang′an University. Her research interests include distributed storage, regenerating codes and locally repairable codes.

王相隆,2018 年于山西农业大学获得学士学位,现为长安大学硕士生,主要研究方向为分布式存储和部分重复码。
E-mail:15135406976@ 163. com

Wang Xianglong received his B. Sc. degree from Shanxi Agricultural University in 2018. He is currently a M. Sc. candidate in Chang′an University. His research interests include distributed storage and fractional repetition codes.