

基于差集矩阵的部分重复码构造

王静^{*①} 何亚锦^① 雷珂^① 刘向阳^②

^①(长安大学信息工程学院 西安 710064)

^②(西北工业大学电子信息学院 西安 710129)

摘要: 针对最小带宽再生码的有效修复问题, 该文提出一种基于差集矩阵的部分重复(FR)码的构造算法。利用差集矩阵和克罗内克(Kronecker)和来构造正交排列, 根据正交排列每一列取相同元素所在行作为节点的编码块, 得到相应的FR码。构造的FR码可以划分成多个平行类, 同时还能调整数据块的重复度和节点的存储容量。仿真结果表明, 与传统的里德-所罗门(RS)码和简单再生码(SRC)相比, 构造的FR码在修复复杂度、修复带宽开销和修复局部性方面具有更好的性能, 修复选择度上虽然是基于表格的修复方案, 但选择度依旧可以达到很高。

关键词: 部分重复码; 分布式存储系统; 差集矩阵; 正交排列

中图分类号: TP301

文献标识码: A

文章编号: 1009-5896(2022)11-4025-09

DOI: 10.11999/JEIT210829

Construction of Fractional Repetition Codes Using Difference Set Matrix

WANG Jing^① HE Yajin^① LEI Ke^① LIU Xiangyang^②

^①(School of Information Engineering, Chang'an University, Xi'an 710064, China)

^②(School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract: Considering the problem of effective repair of minimum bandwidth regenerating codes, a construction algorithm of Fractional Repetition (FR) codes based on difference set matrix is proposed. The orthogonal array is constructed by using the difference set matrix and Kronecker sum. According to the orthogonal array, each row of the same element is taken as the coding blocks of the node to obtain the corresponding FR codes. As a result, the constructed FR codes can be divided into multiple parallel classes, and at the repetition of the data blocks and the storage capacity of the node can be adjusted. The simulation results show that compared with the traditional Reed-Solomon (RS) codes and Simple Regenerating Codes (SRC), the constructed FR codes have better performance in terms of repair complexity, repair bandwidth overhead, and repair locality. Although the repair selectivity is a table-based repair scheme, the selectivity can still reach high.

Key words: Fractional Repetition(FR) codes; Distributed storage systems; Difference set matrix; Orthogonal array

1 引言

由于社交媒体和网络信息的兴起, 数据呈现爆炸式增长, 传统的集中式存储已经无法满足需求。分布式存储系统以其价格低廉、性能优异等优点, 日益成为主流存储系统。在分布式存储系统中, 即使存在故障节点的情况下, 也需要保证数据存储的

完整性和数据访问的速度。这个问题通常使用复制和纠删码策略^[1,2]来解决。

复制策略就是将数据块复制若干倍, 然后将它们分别存放在不同的存储节点上。其中最常见的是3副本复制, 如Google文件系统(Google File System, GFS)^[3]。在节点故障的时候, 由于副本的存在, 只需简单复制数据即可修复。整个修复过程简单易实现, 但是却增加了分布式存储系统的存储负担。为了进一步优化存储开销, 提出纠删码策略。纠删码策略是将原始文件分成若干块进行编码产生冗余校验块来保证数据存储的可靠性, 目前在Windows^[4], Facebook的Hadoop集群(Facebook's Hadoop cluster)^[5]等已经投入使用。虽然纠删码减少了存储

收稿日期: 2021-08-13; 改回日期: 2022-05-07; 网络出版: 2022-05-11

*通信作者: 王静 jingwang@chd.edu.cn

基金项目: 国家自然科学基金(62001059), 陕西省重点研发计划(2021GY-019)

Foundation Items: The National Natural Science Foundation of China (62001059), The Key Research and Development Project of Shaanxi Province (2021GY-019)

开销,但是在节点故障修复的时候,需要下载整个原文件,过程复杂,且修复带宽开销大。

为了进一步优化存储系统,文献[6]提出了再生码(Regenerating Codes, RC),在单节点故障时,再生码不需重构原文件就能修复,降低了修复带宽开销,但是在修复过程中涉及的计算量比较大。Papailiopoulos等人[7]利用线性组合的思想构造了无参数限制的最小带宽再生码(Minimum Bandwidth Regenerating, MBR),结构简单。Rashmi等人[8]提出了锯齿状最小存储再生码(Minimum Storage Regenerating, MSR)构造,在有限域 F_3 上计算开销较小。随后,Guan等人[9]在较小的有限域内构造了MSR码,构造复杂度低。为了降低故障节点的修复局部性,Gopalan等人[10]和Papailiopoulos等人[11]提出了局部修复码(Locally Repairable Codes, LRC)的概念,即单个故障节点可以通过访问最多 r 个存活节点来实现数据恢复。文献[12]将再生码和局部修复码结合,提出了基于系统MSR码的局部再生码,节点故障修复时利用相邻局部码进行协作修复。文献[13]给出具有 (r, t) 局部性的局部修复码的最小距离上界,并且构造的LRC能容忍多节点故障,在修复过程中可以并行读取数据,减少修复时间。

El Rouayheb等人[14]提出了部分重复(Fractional Repetition, FR)码,它是基于最小带宽点的精确修复再生码,采用外部的最大距离可分(Maximum Distance Separable, MDS)码和内部的重复码构成。FR码在修复故障节点时,只需要复制相应的编码块,无需编码计算就可完成修复[15]。文献[16]给出了基于Steiner系的FR码的构造和仿射几何等可分解组合设计,并利用Kronecker和构造出具有较低修复局部性的新FR码。文献[17]提出了基于部分有序集的普遍好的FR码的简单构造,构造的FR码易于实现且可扩展,与MBR码相比,允许存储更大的文件。Zhu等人[18-20]研究了FR码与其对偶码之间的关系,进一步利用正则图和组合设计构造达到最小距离上限的FR码,并给出了FR码重构度的下限。

此后,还有很多学者进行了这方面的研究,FR码的构造一直是一个重要的研究课题。文献[21]引入一种新的组合模型,构造具有均衡访问频率的最优FR码。文献[22]基于二部图的任意周长构造出参数可调节的FR码,并证明其最优性。Prajapati等人[23]使用部分正则图构造异构的(每个节点上存储的数据包数量不同)普遍好的FR码,对于部分参数,使用环构造和 t -设计构造普遍好的FR码。上述构造的FR码都是基于图构造的,过程相对复杂,并且重复度通常是由结构固定,不能对任意参数的FR码进行构造。

考虑到现有的FR码的构造算法复杂且不能灵活地选择构造参数,本文提出一种基于差集矩阵的FR码的构造算法,可以同时调整数据块的重复度和节点的存储容量,且在参数选取上不受限制。进一步可以通过调整差集矩阵的参数 λ ,构造局部性较小的FR码。实验结果表明,构造的FR码具有更低的修复局部性,修复简单且效率高,减少了故障节点的修复时间。

2 预备知识

2.1 FR码

(n, k, d) 分布式存储系统(Distributed Storage System, DSS)中,其中 n 表示的是系统中的存储节点总数, k 表示重构原始文件需要连接的节点数, $d(\geq k)$ 表示修复单个故障节点需要连接的节点数。当存储节点故障时,新生节点需要从其他存活节点下载 β 个数据块进行精确修复,即修复后的数据和失效时的数据是完全一样的。文献[23]指出在精确修复模式下,当 $\beta = 1$ 时,系统的存储容量 C_{MBR} 为

$$C_{\text{MBR}} = kd - \binom{k}{2} \quad (1)$$

在DSS中,FR码外部采用MDS码,将MDS码编码后的 θ 个编码块复制 ρ 倍,再将它们分别存放在 n 个不同的存储节点上,每个节点的存储大小为 α ,连接任意 k 个节点就可以重构原始文件,记为FR $(n, \theta, \rho, \alpha)$ 。考虑到节点之间会存储相同编码块,本文给出FR码的文件大小 $M(k)$ 的定义,即任意 k 个节点 $V_i(i \in \{1, 2, \dots, k\})$ 所获得的不同数据块的个数即求并集

$$M(k) = \min |U_{i \in k} V_i|, k \in \{1, 2, \dots, n\}, |k| = k \quad (2)$$

文献[13]指出如果满足 $M(k) \geq C_{\text{MBR}}(n, k, d)$,则称该FR码是最优的。

2.2 差集矩阵

如果矩阵 \mathbf{G} 表示一个 p 阶可加群,其元素为 $0, 1, \dots, p-1$ 。给定一个矩阵 \mathbf{D} ,如果矩阵 \mathbf{D} 中的任意两列的有序差中, \mathbf{G} 的每个元素都出现且出现的次数相同,本文称矩阵 \mathbf{D} 为差集矩阵(difference matrix),用 $\mathbf{D}(\lambda p, q; p)$ 表示。它是元素属于 \mathbf{G} 的一个具有 p 水平的 $\lambda p \times q$ 的差集矩阵, λ 表示 \mathbf{D} 中的任意两列的有序差中, \mathbf{G} 中的每个元素出现的次数。

2.3 Kronecker和及正交排列

\mathbf{G} 表示一个 n 阶可加群,其元素为 $0, 1, \dots, n-1$ 。假设矩阵 $\mathbf{A} = (a_{ij})_{n \times r}$, $\mathbf{B} = (b_{ij})_{m \times s}$,则矩阵 \mathbf{A} 和 \mathbf{B} 的Kronecker和为

$$\mathbf{A} \oplus \mathbf{B} = (a_{ij} + \mathbf{B})_{mn \times rs} \text{ mod } n \quad (3)$$

其中, $a_{ij} + \mathbf{B}$ 表示 a_{ij} 与 \mathbf{B} 中的每个元素做模 n 加法

运算。接下来, 为了后面表述更加方便, 用 $\mathbf{1}_r$ 表示 $r \times 1$ 阶的全1矩阵, (\mathbf{r}) 表示 $r \times 1$ 矩阵 $(0, 1, \dots, r-1)^T$ 。

一个 $\lambda n^2 \times l$ 的矩阵, 如果满足矩阵的任意两列由 \mathbf{G} 的 n 个元素所构成的 n^2 个序对, 任一序对都出现在这两列的 λ 个行中, 则称这个矩阵为正交排列, 记为 $\mathbf{OA}(n, l; \lambda)$ 。

3 基于差集矩阵的FR码构造

本节利用差集矩阵构造FR码, 构造的FR码可以调整编码块的重复度, 并且可以通过设计参数 λ , 进一步调节节点之间共有的编码块数, 达到减

设 $\alpha = [0, 1, \dots, p-1]^T$, $\alpha_i = [i, i, \dots, i]^T (i = 0, 1, \dots, p-1)$, 定义函数 $M^j(\alpha) = [0+j, 1+j, \dots, p-1+j]^T$, 其中 $i+j (i = 0, 1, \dots, p-1)$ 表示模 p 加, 构造如式(5)的正交矩阵

$$\mathbf{OA}(p, (q+1); 1) = \begin{bmatrix} M^{0 \times 0}(\alpha) & M^{0 \times 1}(\alpha) & \dots & M^{0 \times (q-1)}(\alpha) & \alpha_0 \\ M^{1 \times 0}(\alpha) & M^{1 \times 1}(\alpha) & \dots & M^{1 \times (q-1)}(\alpha) & \alpha_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ M^{(p-1) \times 0}(\alpha) & M^{(p-1) \times 1}(\alpha) & \dots & M^{(p-1) \times (q-1)}(\alpha) & \alpha_{p-1} \end{bmatrix} \quad (5)$$

矩阵 \mathbf{OA} 可以进一步表示成

$$\mathbf{OA}(p, (q+1); 1) = \begin{bmatrix} 0 \times 0 & 0 \times 1 & \dots & 0 \times (q-1) \\ 1 \times 0 & 1 \times 1 & \dots & 1 \times (q-1) \\ \vdots & \vdots & \ddots & \vdots \\ (p-1) \times 0 & (p-1) \times 1 & \dots & (p-1) \times (q-1) \end{bmatrix} \oplus (\mathbf{p}), (\mathbf{p}) \oplus \mathbf{0}_p \quad (6)$$

根据差集矩阵的定义, $\mathbf{OA}(p, (q+1); 1) = [\mathbf{D}(p, q; p) \oplus (\mathbf{p}), (\mathbf{p}) \oplus \mathbf{0}_p]$, 可以验证得

$$\mathbf{D}(p, q; p) = \begin{bmatrix} 0 \times 0 & 0 \times 1 & \dots & 0 \times (q-1) \\ 1 \times 0 & 1 \times 1 & \dots & 1 \times (q-1) \\ \vdots & \vdots & \ddots & \vdots \\ (p-1) \times 0 & (p-1) \times 1 & \dots & (p-1) \times (q-1) \end{bmatrix} \quad (7)$$

是一个模 p 加法群上的差集矩阵。这里的 $\mathbf{OA}(p, (q+1); 1)$, $\mathbf{D}(p, q; p)$ 中的 $(p-1)$ 和 $(q-1)$ 分别表示数 $p-1$ 和数 $q-1$, 而 $(\mathbf{p}) = (0, 1, \dots, p-1)^T$ 。

定理1 正交排列 $\mathbf{OA}(p, (q+1); \lambda)$ 构造FR码, 共有 $p(q+1)$ 个节点, λp^2 个编码块, 重复度为 $q+1$, 节点存储大小为 λp , 即 $\text{FR}(\lambda p^2, p(q+1), q+1, \lambda p)$ 。节点故障时, 需要连接 p 个节点修复。

证明 正交排列 $\mathbf{OA}(p, (q+1); \lambda) = [\mathbf{D}(\lambda p, q; p) \oplus (\mathbf{p}), (\mathbf{p}) \oplus \mathbf{0}_{\lambda p}]$ 矩阵是一个 $\lambda p^2 \times (q+1)$ 矩阵, \mathbf{OA} 的 λp^2 行对应 λp^2 个不同的编码块, \mathbf{OA} 的 $q+1$ 列对应着FR码的重复度。由构造过程可知, \mathbf{OA} 的任一列都包含了所有的编码块, 且每个编码块都正好与唯一的节点相关联, 因此由正交排列 \mathbf{OA} 矩阵构造的FR码有 $q+1$ 个平行类, 即正交排列 \mathbf{OA} 矩阵的每一列都是一个平行类, 本文称FR码是可分解的。每列取 p 个不同的元素, 每个元素在该列中出现 λp 次。针对某列, 将取相同元素所在行编码块放置

小修复局部性的目的。

3.1 基于差集矩阵的FR码的构造算法

设 \mathbf{G} 为 p 阶可加群, 元素为 $0, 1, \dots, p-1$ 。若存在元素属于 \mathbf{G} 的一个差集矩阵 $\mathbf{D}(\lambda p, q; p)$, 构造矩阵为

$$\mathbf{OA}(p, (q+1); \lambda) = [\mathbf{D}(\lambda p, q; p) \oplus (\mathbf{p}), (\mathbf{p}) \oplus \mathbf{0}_{\lambda p}] \quad (4)$$

具体地, 基于差集矩阵 $\lambda = 1$ 的FR码的具体构造步骤如下所示:

当 $l > 1, p > 1$ 且 $l \leq q+1$ (q 为 p 的最小素因数)时, 给出了 $\mathbf{OA}(p, (q+1); 1)$ 的通用构造方法, 并在此基础上构造了普遍好的FR码。

在一个节点中, 节点之间有 λ 个相同的编码块。故单节点故障时, 只需从 p 个节点上下载 λ 个编码块即可修复。考虑多节点故障时, 因为每个平行类含有 p 个节点, 故多节点故障时同样仅需从 p 个节点中下载数据即可完成修复。证毕

定理2 由上述 $\lambda = 1$ 的差集矩阵构造的FR码是最优的FR码。

证明 $\lambda = 1$ 时, 正交排列 $\mathbf{OA}(p, (q+1); 1)$, 矩阵 \mathbf{OA} 任两列的元素所组成的有序对仅出现1次, 即任意两列中的两行的元素对不会完全相同。由FR码的构造方法可知, 每一列对应一个平行类, 所以平行类之间的存储节点有一个编码块相同。假定FR码每个节点存储大小为 d , 即修复度, 连接任意 k 个节点可以重构原文件。因为两个节点之间最多相交一个编码块, 所以当连接 k 个不同的节点时, 最多可以得到 $\binom{k}{2}$ 个重复数据块, 根据FR码原文件大小的定义可得 $M(k) \geq kd - \binom{k}{2} = C_{\text{MBR}}$ 。因

此, 由 $\lambda = 1$ 的差集矩阵构造的FR码是最优的FR码。证毕

引理1 当 $\lambda = 1$ 且 p 为素数时, 构造的FR码的重复度 $\rho = 2$ 。

证明 p 为素数时, 其最小素因子 $q = 1$, 则构造的差集矩阵为 $D(p, 1; p)$, 进一步得到一个 $p \times 2$ 的正交矩阵 $OA(p, 2; 1)$, 已知正交矩阵有两列, 故构造的FR码的重复度 $\rho = 2$ 。证毕

引理2 当 p 为合数时, 构造的FR码的重复度

$$\left. \begin{aligned} M(k) &\geq 2kp - 2 \left[\binom{k}{2} - v \right] + 1, & w < 2 \\ M(k) &\geq 2kp - 2 \left[\binom{k}{2} - v \binom{w+1}{2} - (3-v) \binom{w}{2} \right] + 1, & w \geq 2 \end{aligned} \right\} \quad (8)$$

证明 考虑 $\lambda = 2$, 故 λp 的最小素因子为2, 即 $q = 2$, 则进一步构造基于 $\lambda = 2$ 的差集矩阵的FR($2p^2, 3p, 3, 2p$)码。FR码每个节点存储大小为 $2p$, 一个平行类含有 p 个节点, 不论单节点还是多节点的节点故障, 仅需连接 p 个节点即可修复。

考虑节点故障修复的局部性, 令 $k = p + 1$, 其中 $k = 3w + v$, 文件大小为 $M(k) = \min |U_{i \in k} V_i|$, $k \in \{1, 2, \dots, p\}$, $|k| = k$

$$\left. \begin{aligned} |V_1 \cup V_2 \cup \dots \cup V_k| &= \sum_{i=1}^k |V_i| - \sum_{1 \leq i < j \leq k} |V_i \cap V_j| + \sum_{1 \leq i < j < m \leq k} |V_i \cap V_j \cap V_m| \\ &\quad - \dots + (-1)^{k-1} |V_1 \cap V_2 \cap \dots \cap V_k| \\ M(k) = |U_{i \in k} V_i| &\geq \sum_{i=1}^k |V_i| - \sum_{i < j \leq k} |V_i \cap V_j| + \sum_{1 \leq i < j < m \leq k} |V_i \cap V_j \cap V_m| \end{aligned} \right\} \quad (9)$$

考虑两个节点相交的编码块数, 优先考虑不在一个平行类中的节点。任取 k 个节点所获得的编码块为 $2kp$, 根据 $k = 3w + v$ 可知, 有 $3 - v$ 个平行类取 w 个节点, 有 v 个平行类取 $w + 1$ 个节点。由于同一平行类内的节点之间无相交编码块, 而平行类间的两节点之间有一个相同的编码块。当 $0 < w < 2$ 时, k 个节点至少有 $2[\binom{k}{2} - v]$ 个编码块, $w \geq 2$ 时, k 个节点至少有 $2[\binom{k}{2} - v\binom{w+1}{2} - (3-v)\binom{w}{2}]$ 个相同编码块。考虑 p 是每个节点存储的编码块数, p 最小取2, 故 k 最小为3。由构造过程可知, 分别从3个平行类中取1个节点, 则3个节点最少相交1个编码块。故

$$\left. \begin{aligned} M(k) &\geq 2kp - 2 \left[\binom{k}{2} - v \right] + 1, & w < 2 \\ M(k) &\geq 2kp - 2 \left[\binom{k}{2} - v \binom{w+1}{2} - (3-v) \binom{w}{2} \right] + 1, & w \geq 2 \end{aligned} \right\} \quad (10)$$

证毕

3.2 构造法的应用

例1 当 p 为素数且 $p = 3$ 时, 其最小素因数 $q = 1$, 则构造正交矩阵 $OA(3, 2; 1) = [D(3, 1; 3)] \oplus (\mathbf{3})$, $(\mathbf{3}) \oplus \mathbf{0}_3$ 如式(11)所示, 矩阵 $OA(3, 2; 1)$ 后面为所对应的编码块。

$$OA(3, 2; 1) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 2 \\ 2 & 0 & 3 \\ 0 & 1 & 4 \\ 1 & 1 & 5 \\ 2 & 1 & 6 \\ 0 & 2 & 7 \\ 1 & 2 & 8 \\ 2 & 2 & 9 \end{bmatrix} \quad (11)$$

$$\left. \begin{aligned} p_1 \quad V_1 = \{1, 4, 7\} \quad V_2 = \{2, 5, 8\} \quad V_3 = \{3, 6, 9\} \\ p_2 \quad V_4 = \{1, 2, 3\} \quad V_5 = \{4, 5, 6\} \quad V_6 = \{7, 8, 9\} \end{aligned} \right\} \quad (12)$$

由于 $\mathbf{OA}(3, 2; 1)$ 矩阵的每一列都包含了所有的编码块，且每个编码块只对应唯一的一个节点。因此每一列都是一个平行类。考虑矩阵的最后一列，前3个元素取第1水平0，即水平“0”，于是把“1,2,3”放到 p_2 的第“1”个节点中。类似地，把“4,5,6”放到 p_2 的第“2”个节点中，把“7,8,9”放到 p_2 的第“3”个节点中，这3个节点构成平行类 p_2 。同理可以得到平行类 p_1 的3个节点，最终得到6个节点。

将原文件 M 分成6个数据块，采用(9, 6)MDS码编码，得到9个编码块。用户连接任意3个节点至少可以获得6个不同的数据块，即可重构原文件 M 。又由 $C_{\text{MBR}} = 3 \times 3 - C_3^2 = 6$ ，所以构造的FR码是最优的。构造的FR码的重复度 $\rho = 2$ ，并且有2个平行类，每个平行类包含了所有的数据块。

例2 令 $p = 4$ ，其最小素因子 $q = 2$ ，可以得到一个 4×2 的差集矩阵 $\mathbf{D}(4, 2; 4)$ 。利用差集矩阵 $\mathbf{D}(4, 2; 4)$ ，构造得到矩阵 $\mathbf{OA}(4, 3; 1) = [\mathbf{D}(4, 2; 4) \oplus (\mathbf{4}), (\mathbf{4}) \oplus \mathbf{0}_4]$ ，如式(13)所示。矩阵 $\mathbf{OA}(4, 3; 1)$ 后面附的是相对应的编码块

$$\mathbf{D}(4, 2; 4) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 2 \\ 0 & 3 \end{bmatrix} \quad \mathbf{OA}(4, 3; 1) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 2 \\ 2 & 2 & 0 & 3 \\ 3 & 3 & 0 & 4 \\ 0 & 1 & 1 & 5 \\ 1 & 2 & 1 & 6 \\ 2 & 3 & 1 & 7 \\ 3 & 0 & 1 & 8 \\ 0 & 2 & 2 & 9 \\ 1 & 3 & 2 & 10 \\ 2 & 0 & 2 & 11 \\ 3 & 1 & 2 & 12 \\ 0 & 3 & 3 & 13 \\ 1 & 0 & 3 & 14 \\ 2 & 1 & 3 & 15 \\ 3 & 2 & 3 & 16 \end{bmatrix} \quad (13)$$

$$\left. \begin{aligned} p_1 \quad V_1 = \{1, 5, 9, 13\} \quad V_2 = \{2, 6, 10, 14\} \\ V_3 = \{3, 7, 11, 15\} \quad V_4 = \{4, 8, 12, 16\} \\ p_2 \quad V_5 = \{1, 8, 11, 14\} \quad V_6 = \{2, 5, 12, 15\} \\ V_7 = \{3, 6, 9, 16\} \quad V_8 = \{4, 7, 10, 13\} \\ p_3 \quad V_9 = \{1, 2, 3, 4\} \quad V_{10} = \{5, 6, 7, 8\} \\ V_{11} = \{9, 10, 11, 12\} \quad V_{12} = \{13, 14, 15, 16\} \end{aligned} \right\} \quad (14)$$

采用系统(16, 13)MDS码，每个节点存储4个编码块。如果选取其中的任意两个平行类，则可以得到一个重复度 $\rho = 2$ 的FR码；如果取3个平行类，就得到了重复度 $\rho = 3$ 的FR码。

例3 考虑 $\lambda = 2$ ，利用差集矩阵 $\mathbf{D}(6, 2; 3)$ ，构造得到矩阵 $\mathbf{OA}(3, 3; 2) = [\mathbf{D}(6, 2; 3) \oplus (\mathbf{3}), (\mathbf{3}) \oplus \mathbf{0}_6]$ ，如下所示。通过该设计，我们可以得到一个 $\rho = 3$ 的FR码，并且有3个平行类。更具体地说，采用系统(18, 15)MDS码，对15个数据块编码，可以生成3个校验块，其中每个节点存储6个数据包。如果节点 V_1 故障，则新生节点可以从节点 V_4 下载{1,10}，从 V_5 下载{4,13}，从 V_6 下载{7,16}，来修复故障节点 V_1 ；或者从 V_7 下载{1,4}，从 V_8 下载{7,10}，从 V_9 下载{13,16}，实现故障节点 V_1 的修复。进一步可以看出，任意连接 $k = 4$ 个节点，用户都可以恢复原文件。且当 $\lambda = 2$ 时，上述构造出来的FR码是局部修复码，单节点的修复局部性为3

$$\mathbf{D}(6, 2; 3) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 2 \\ 0 & 0 \\ 0 & 1 \\ 0 & 2 \end{bmatrix} \quad \mathbf{OA}(3, 3; 2) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 2 \\ 2 & 2 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 1 & 2 & 0 & 5 \\ 2 & 0 & 0 & 6 \\ 0 & 2 & 1 & 7 \\ 1 & 0 & 1 & 8 \\ 2 & 1 & 1 & 9 \\ 0 & 0 & 1 & 10 \\ 1 & 1 & 1 & 11 \\ 2 & 2 & 1 & 12 \\ 0 & 1 & 2 & 13 \\ 1 & 2 & 2 & 14 \\ 2 & 0 & 2 & 15 \\ 0 & 2 & 2 & 16 \\ 1 & 0 & 2 & 17 \\ 2 & 1 & 2 & 18 \end{bmatrix} \quad (15)$$

$$\left. \begin{aligned} p_1 \quad V_1 = \{1, 4, 7, 10, 13, 16\} \quad V_2 = \{2, 5, 8, 11, 14, 17\} \\ V_3 = \{3, 6, 9, 12, 15, 18\} \\ p_2 \quad V_4 = \{1, 6, 8, 10, 15, 17\} \quad V_5 = \{2, 4, 9, 11, 13, 18\} \\ V_6 = \{3, 5, 7, 12, 14, 16\} \\ p_3 \quad V_7 = \{1, 2, 3, 4, 5, 6\} \quad V_8 = \{7, 8, 9, 10, 11, 12\} \\ V_9 = \{13, 14, 15, 16, 17, 18\} \end{aligned} \right\} \quad (16)$$

3.3 故障节点修复

分布式存储系统中出现节点故障，在修复过程中需要连接的节点集合称为修复组。本节主要从单节点故障、两节点故障以及多节点故障进行分类讨论，具体修复过程如下：

当单节点故障的时候，只需要连接任意一个完整平行类即可进行精确无编码修复。如例2构造的FR码，当第1个平行类 p_1 中的节点 V_1 故障时，可以从平行类 p_2 中的节点 V_5 下载编码块1，从节点 V_6 下载编码块5，从节点 V_7 下载编码块9，从节点 V_8 下载编码块13，进行无编码精确修复。节点 V_1 故障也可以从平行类 p_3 中的4个节点 $\{V_9, V_{10}, V_{11}, V_{12}\}$ 进

行修复, 或者利用平行类 p_2 和 p_3 混合修复如修复集 $\{V_5, V_6, V_{11}, V_{12}\}$ 等多种修复选择方案。连接任意修复组中的4个节点通过复制相关编码块即可进行无编码精确修复故障节点。

当系统中出现两个节点同时故障时, 分为以下情况: 如果重复度 $\rho = 2$, 两个故障节点在同一个平行类中, 则可以利用另一个平行类进行修复; 如果两个故障节点不在同一个平行类, 且两个故障节点有一个编码块相同, 这种情况下就不能采用复制的方式进行修复, 需要利用其余存活节点恢复原文件, 进一步修复故障节点。如例1中的节点 V_1 和 V_2 故障, 两个故障节点都在平行类 p_1 , 则可以用另一个平行类 p_2 中的3个节点 $\{V_4, V_5, V_6\}$ 来修复故障节点。如果节点 V_1 和节点 V_4 同时故障, 由于两个故障节点中含有相同编码块, 故不能采用简单复制的方式, 可以连接其余两个节点 V_2 和 V_3 进行MDS码编码恢复原文件, 进一步精确修复故障节点 V_1 和 V_4 。针对重复度 $\rho > 2$ 的情况, 不论是两个故障节点在一个平行类, 还是其他情况, 均可以利用其他平行类中的节点下载相应编码块进行精确修复。如例2中的节点 V_1 和 V_5 故障, 可以利用平行类 p_3 中的4个节点 $\{V_9, V_{10}, V_{11}, V_{12}\}$ 进行修复。

对于修复多个故障节点, 可以分为以下4种情况: (1) 若多个故障节点都在同一个平行类, 则可以利用其他平行类进行修复; (2) 对于多个故障节点不在同一个平行类的情况, 若还存在不包括故障节点的平行类, 那么就可以直接用不包括故障节点的平行类进行修复; (3) 若多个故障节点分布在所有平行类中, 且不存在所有平行类中的故障节点都包含同一个或者多个编码块的情况, 此时可以通过连接多个平行类中的存活节点来修复故障节点。(4) 若多个故障节点分布在所有平行类中, 且有一个或者多个编码块无法从现有存活节点中获得, 此时无法再采用复制方式进行修复, 需要利用存活节点先恢复原文件, 从而修复故障节点。

4 性能分析

基于差集矩阵构造的FR码的性能分析主要集

中在节点修复选择度、修复复杂度、修复带宽开销和修复局部性这几方面, 并将其与里德-所罗门(Reed-Solomon, RS)码和简单再生码(Simple Regenerating Codes, SRC)进行性能比较。

表1给出了相同文件情况下RS码、SRC和基于 $\lambda = 1$ 差集矩阵构造的FR码的修复带宽开销和修复局部性。本文构造的FR码外部采用 $(k + 3, k)$ MDS码。

4.1 节点修复选择度

节点修复选择度指的是故障节点修复过程中的修复选择方案。考虑单节点故障的修复选择度, 对于 (n, k) RS码, 从 $n - 1$ 个存活节点随机选取 k 个来修复故障节点。因此, (n, k) RS码的修复选择度为 C_{n-1}^k 。对于SRC, 从 $n - 1$ 个存活节点随机选取 f 个存活节点来修复故障节点, 故其修复选择度为 C_{n-1}^f 。基于 $\lambda = 1$ 差集矩阵构造的FR码, 其选择度主要与重复度 ρ 和节点存储大小 α 有关。如果构造的FR码的重复度 $\rho = 2$, 那么节点故障时的修复选择度仅为1, 只需要连接具有相同编码块的存活节点即可修复。如果构造的FR码的重复度 $\rho > 2$ 时, 故障节点修复可以有多种修复方案。对于任意单节点故障, 都可以从其他 $\rho - 1$ 个存活节点选择修复。对于重复度为 ρ 、节点存储大小为 α 的FR码, 节点故障的时候, 只需要从其他副本中下载相应的数据块即可。由于任意两个节点之间最多相交1个编码块, 修复一个节点中的第1个编码块和第2个编码块均有 $\rho - 1$ 种选取方案, 故其修复选择度为 $(\rho - 1)^\alpha$ 。从图1看出, 当FR码节点存储大小 $\alpha = 3$ 时, 基于 $\lambda = 1$ 差集矩阵构造的FR码节点修复选择度随着重复度的增加呈指数倍增加。

4.2 修复复杂度

对于 (n, k) RS码, 在修复过程中需要连接 k 个节点恢复原文件, 再重新编码生成故障节点。整个故障节点修复过程涉及大量的有限域运算, 需要 $k^2 + k$ 次有限域乘法运算和 $k^2 - 1$ 次有限域加法运算, 因此RS码在单节点故障时的修复复杂度为 $O(2k^2 + k - 1)$ 。对于SRC, 修复一个编码块需要进行 $f - 1$ 次异或运算, 由于每个节点存储 $f + 1$ 个

表1 RS码、SRC和FR码的故障节点修复性能比较

修复带宽开销和修复局部性	编码方案		
	RS码	SRC	FR码
单节点修复带宽开销	M	$(f + 1)M/k$	$\sqrt{(k + 3)M/k}$
两节点修复带宽开销	M	$2(f + 1)M/k$ 或 M	$\sqrt{(k + 3)M/k}$
单节点修复局部性	k	$2f$	$\sqrt{(k + 3)}$
两节点修复局部性	k	k	$\sqrt{(k + 3)}$

编码块，因此SRC的修复过程需要 $(f + 1)(f - 1)$ 次异或运算，其修复复杂度为 $O(f^2 - 1)$ 。基于差集矩阵构造的FR码，整个过程只涉及文件的读取，无需编码，过程简单，修复复杂度明显低于RS码和SRC这两种编码方案。

4.3 修复带宽开销

分布式存储系统中出现节点故障，在修复过程中下载的数据量大小称为修复带宽开销。针对单节点故障，传统的 (n, k) RS码在修复过程中需要下载整个原文件来修复故障节点，故单节点修复带宽开销为 M ；对于 (n, k, f) SRC，SRC在修复过程中需要连接 f 个存活节点，每个节点存储 $f + 1$ 个数据块，且单个数据块大小为 M/fk ，故单节点修复带宽开销为 $(f + 1)M/k$ ；针对本文基于 $\lambda = 1$ 差集矩阵构造的FR码，由构造可知一个平行类中有 λp^2 个编码块，节点存储大小为 λp ，由于外部采用系统 $(k + 3, k)$ MDS码，故有 $\lambda p^2 = k + 3$ ，经计算可知，一个平行类中含有 $p = \sqrt{(k + 3)}$ 个节点，每个数据块大小为 M/k ，故基于 $\lambda = 1$ 差集矩阵构造的FR码的修复带宽开销为 $\sqrt{(k + 3)}M/k$ 。考虑到基于 $\lambda = 1$ 构造的FR码是最优的，由文献[13]可知， $M(k) \geq C_{\text{MBR}}(n, k, d)$ ，故可以得出 $M(k) \geq k\sqrt{k + 3} - (k - 1)k/2$ 。因此可以计算出FR码的最小修复带宽开销为 $k + 3 - \sqrt{k + 3}(k - 1)/2$ ，可计算出 $k = 1$ 时最小，修复带宽开销为4，此时 $M(k) \geq 2$ 。

在性能分析部分提前设定相关参数，文件大小恒为 $M = 1000 \text{ Mb}$ ，SRC的子文件数 $f = 4$ 。单节点故障时，SRC的修复带宽开销为 $5M/k$ ，采用例2中构造的FR码的修复带宽开销为 $4M/k$ ，由于文件大小为 $M = 1000 \text{ Mb}$ ，并且基于 $\lambda = 1$ 构造的FR码是最优的，满足 $M(k) \geq C_{\text{MBR}}(n, k, d)$ ，因此 k 存在最大值，当 k 取最大值时，修复带宽开销最小。单节点故障修复带宽开销如图2所示。

当两节点故障时，传统的 (n, k) RS码在修复过程中需要下载整个原文件来修复故障节点，故两节点修复带宽开销依旧为 M ；对于 (n, k, f) SRC，其

修复带宽受两个故障节点之间的节点数影响。如果节点数大于 $f - 1$ ，那么这两个故障节点可以分别连接 f 个存活节点来修复，此时的修复带宽开销为 $2(f + 1)M/k$ ，反之如果节点数小于 $f - 1$ ，不能按照单节点故障修复方法进行修复，需要先恢复原文件，进一步修复故障节点，故修复带宽为 M 。针对本文基于 $\lambda = 1$ 差集矩阵构造的FR码，由于一个平行类含有 $\sqrt{(k + 3)}$ 个节点，故两个节点故障时的修复带宽开销依旧为 $\sqrt{(k + 3)}M/k$ 。

两个节点故障时，采用例2中FR码的修复带宽开销为 $4M/k$ ，而RS码和SRC的修复带宽开销为 M 。两节点故障修复带宽开销如图3所示，其中RS码和SRC的曲线重合。

从图2和图3可以看出，不论是单节点故障还是多节点故障，基于 $\lambda = 1$ 差集矩阵构造的FR码的修复带宽开销明显优于RS码和SRC简单再生码。

4.4 修复局部性

考虑单节点故障时， (n, k) RS码在修复过程中，需要连接 k 个节点，修复局部性为 k ；而SRC需要连接 $2f$ 个节点，修复局部性为 $2f$ ；针对基于差集矩阵构造的FR码，由构造可知一个平行类中共有 λp^2 个编码块，节点存储大小为 λp ，一个平行类中包含 p 个节点，故单节点故障修复局部性为 p 。采用系统 $(k + 3, k)$ MDS码，故有 $\lambda p^2 = k + 3$ ，经计算可知，单节点故障的修复局部性为 $p = \sqrt{(k + 3)}/\lambda$ 。

考虑两节点故障，SRC和 (n, k) RS码的修复局部性均为 k 。而本文提出的基于差集矩阵的FR码，

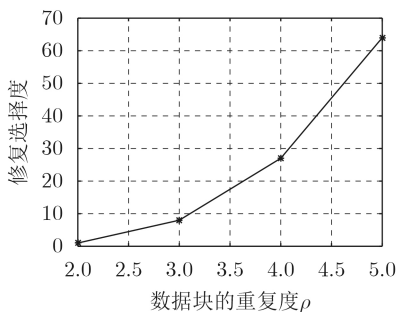


图1 节点修复选择度与重复度 ρ 之间的关系，其中 $\alpha = 3$

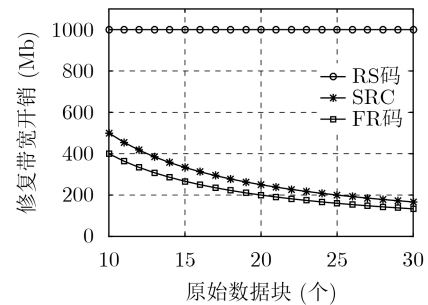


图2 $\lambda = 1$ 时单节点故障修复带宽开销性能对比

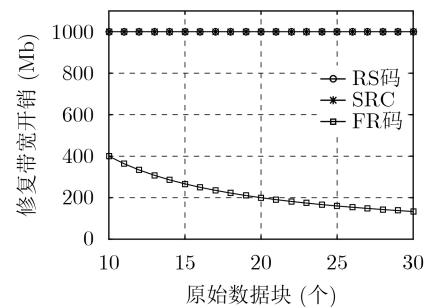


图3 $\lambda = 1$ 时两节点故障修复带宽开销性能对比

一个平行类含有 p 个节点,故 p 个节点即可以修复任意节点故障,因此修复局部性恒为 $p = \sqrt{(k+3)/\lambda}$ 。

假设FR码外部采用 $(k+3, k)$ MDS码中的 $k=13$,则 $\lambda=1$ 构造的FR码外部采用 $(16, 13)$ MDS码。如图4所示,考虑单节点故障,(16, 13)RS码的修复局部性为13;当 f 取4时, SRC的修复局部性为8;采用例2中构造的FR码的修复局部性为4。当两个节点故障时,(16, 13)RS码和SRC的修复局部性均为13,采用例2中FR码的修复局部性仍为4,具体如图4所示。与RS码和SRC相比,无论单节点故障还是两节点故障,基于 $\lambda=1$ 的差集矩阵构造的FR码都具有较优的修复局部性。

假设FR码外部采用 $(k+3, k)$ MDS码中的 $k=15$,则基于 $\lambda=2$ 的差集矩阵构造的FR码外部采用 $(18, 15)$ MDS码。如图5所示,单节点故障时,(18, 15)RS码的修复局部性为15;当 f 取4时, SRC的修复局部性为8;采用例3中FR码的修复局部性为3。考虑两节点故障时,(18, 15)RS码和SRC的修复局部性均为15,采用例3中FR码的修复局部性仍为3。与RS码和SRC相比,无论单节点故障还是两节点故障,基于差集矩阵 $\lambda > 1$ 的FR码都具有较小的修复局部性。

5 结论

针对目前基于部分图构造的FR码,构造过程复杂,本文提出了一种基于差集矩阵的FR码的构造算法,可以容忍多节点故障,同时可以任意调整节点间的共有编码块数,在参数选取上比较灵活。进一步证明了提出的基于差集矩阵 $\lambda=1$ 的FR码是

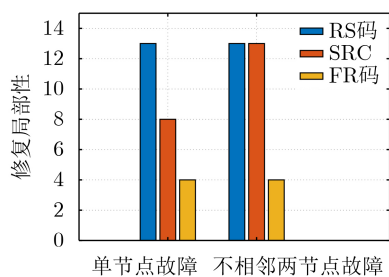


图4 $\lambda=1$ 时单节点故障修复局部性性能对比

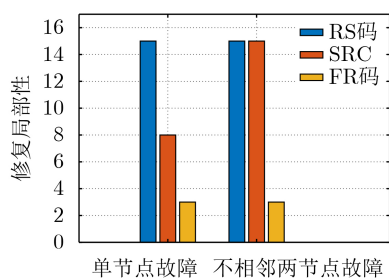


图5 $\lambda=2$ 时两节点故障修复局部性性能对比

最优的,并给出了基于差集矩阵 $\lambda=2$ 构造的FR码的文件大小,且其修复局部性明显减小。性能分析和仿真结果表明,与传统的RS码和SRC相比,构造的FR码虽然牺牲了存储开销,但是可以显著减少故障节点的修复局部性和修复带宽开销,降低修复复杂度。

参考文献

- [1] LIU Ying and VLASSOV V. Replication in distributed storage systems: State of the art, possible directions, and open issues[C]. 2013 International Conference on Cyber-enabled Distributed Computing and Knowledge Discovery, Beijing, China, 2013: 225–232. doi: 10.1109/CyberC.2013.44.
- [2] LI Jun and LI Baochun. Erasure coding for cloud storage systems: A survey[J]. *Tsinghua Science and Technology*, 2013, 18(3): 259–272. doi: 10.1109/TST.2013.6522585.
- [3] GHEMAWAT S, GOBIOFF H, and LEUNG S T. The Google file system[J]. *ACM SIGOPS Operating Systems Review*, 2003, 37(5): 29–43. doi: 10.1145/1165389.945450.
- [4] HUANG Cheng, SIMITCI H, XU Yikang, et al. Erasure coding in windows azure storage[C]. The 2012 USENIX Annual Technical Conference, Boston, USA, 2012: 15–26.
- [5] SATHIAMOORTHY M, ASTERIS M, PAPAILIOPOULOS D, et al. XORing elephants: Novel erasure codes for big data[J]. *Proceedings of the VLDB Endowment*, 2013, 6(5): 325–336. doi: 10.14778/2535573.2488339.
- [6] DIMAKIS A G, GODFREY P B, WU Yunnan, et al. Network coding for distributed storage systems[J]. *IEEE Transactions on Information Theory*, 2010, 56(9): 4539–4551. doi: 10.1109/TIT.2010.2054295.
- [7] PAPAILIOPOULOS D S and DIMAKIS A G. Distributed storage codes through Hadamard designs[C]. 2011 IEEE International Symposium on Information Theory Proceedings, St. Petersburg, Russia, 2011: 1230–1234. doi: 10.1109/ISIT.2011.6033731.
- [8] RASHMI K V, SHAH N B, and 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. doi: 10.1109/TIT.2011.2159049.
- [9] GUAN Sheng, KAN Haibin, WEN Jie, et al. A new construction of exact-repair MSR codes using linearly dependent vectors[J]. *IEEE Communications Letters*, 2017, 21(8): 1691–1694. doi: 10.1109/LCOMM.2017.2700862.
- [10] GOPALAN P, HUANG Cheng, SIMITCI H, et al. On the locality of codeword symbols[J]. *IEEE Transactions on Information Theory*, 2012, 58(11): 6925–6934. doi: 10.1109/TIT.2012.2208937.
- [11] PAPAILIOPOULOS D S, LUO Jianqiang, DIMAKIS A G,

- et al.* Simple regenerating codes: Network coding for cloud storage[C]. 2012 Proceedings IEEE INFOCOM, Orlando, USA, 2012: 2801–2805. doi: [10.1109/INFCOM.2012.6195703](https://doi.org/10.1109/INFCOM.2012.6195703).
- [12] WANG Jing, YAN Zhiyuan, 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. doi: [10.1109/ACCESS.2020.2975577](https://doi.org/10.1109/ACCESS.2020.2975577).
- [13] HAO Jie, SHUM K W, XIA Shutao, *et al.* Optimal locally repairable codes for parallel reading[J]. *IEEE Access*, 2020, 8: 80447–80453. doi: [10.1109/ACCESS.2020.2992188](https://doi.org/10.1109/ACCESS.2020.2992188).
- [14] EL ROUAYHEB S and RAMCHANDRAN K. Fractional repetition codes for repair in distributed storage systems[C]. 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, USA, 2010: 1510–1517. doi: [10.1109/ALLERTON.2010.5707092](https://doi.org/10.1109/ALLERTON.2010.5707092).
- [15] ZHOU Tai, LI Hui, ZHU Bing, *et al.* STORE: Data recovery with approximate minimum network bandwidth and disk I/O in distributed storage systems[C]. 2014 IEEE International Conference on Big Data, Washington, USA, 2014: 33–38. doi: [10.1109/BigData.2014.7004381](https://doi.org/10.1109/BigData.2014.7004381).
- [16] OLMEZ O and RAMAMOORTHY A. Fractional repetition codes with flexible repair from combinatorial designs[J]. *IEEE Transactions on Information Theory*, 2016, 62(4): 1565–1591. doi: [10.1109/TIT.2016.2531720](https://doi.org/10.1109/TIT.2016.2531720).
- [17] AYDINIAN H and BOCHE H. Fractional repetition codes based on partially ordered sets[C]. 2017 IEEE Information Theory Workshop (ITW), Kaohsiung, China, 2017: 51–55. doi: [10.1109/ITW.2017.8277958](https://doi.org/10.1109/ITW.2017.8277958).
- [18] ZHU Bing, SHUM K W, LI Hui, *et al.* On the duality and file size hierarchy of fractional repetition codes[J]. *The Computer Journal*, 2019, 62(1): 150–160. doi: [10.1093/comjnl/bxy094](https://doi.org/10.1093/comjnl/bxy094).
- [19] ZHU Bing, SHUM K W, WANG Weiping, *et al.* On the optimal minimum distance of fractional repetition codes[C]. GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, China, 2020: 1–6. doi: [10.1109/GLOBECOM42002.2020.9322318](https://doi.org/10.1109/GLOBECOM42002.2020.9322318).
- [20] ZHU Bing, SHUM K W, and LI Hui. Fractional repetition codes with optimal reconstruction degree[J]. *IEEE Transactions on Information Theory*, 2020, 66(2): 983–994. doi: [10.1109/TIT.2019.2929788](https://doi.org/10.1109/TIT.2019.2929788).
- [21] YU Wenjun, ZHANG Xiande, and GE Gennian. Optimal fraction repetition codes for access-balancing in distributed storage[J]. *IEEE Transactions on Information Theory*, 2021, 67(3): 1630–1640. doi: [10.1109/TIT.2020.3039901](https://doi.org/10.1109/TIT.2020.3039901).
- [22] SU Yisheng. Optimal pliable fractional repetition codes that are locally recoverable: A bipartite graph approach[J]. *IEEE Transactions on Information Theory*, 2019, 65(2): 985–999. doi: [10.1109/TIT.2018.2876284](https://doi.org/10.1109/TIT.2018.2876284).
- [23] PRAJAPATI S A, DEB S, and GUPTA M K. On some universally good fractional repetition codes[C]. 2020 International Conference on COMmunication Systems & NETworks (COMSNETS), Bengaluru, India, 2020: 404–411. doi: [10.1109/COMSNETS48256.2020.9027326](https://doi.org/10.1109/COMSNETS48256.2020.9027326).
- 王 静：女，博士，教授，研究方向为网络编码、再生码和大数据存储。
何亚锦：女，硕士，研究方向为分布式存储、网络编码和部分重复码。
雷 珂：女，硕士生，研究方向为分布式存储、再生码和部分重复码。
刘向阳：男，博士，副教授，研究方向为分布式存储、信号处理。

责任编辑：余 蓉