• 通信与信息工程 •

基于 Hadamard 矩阵构造部分重复码

王 静[1]，孙 伟[1*]，何亚锦[1]，沈克勤[1]，张鑫楠[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】针对分布式存储系统故障节点修复问题，提出一种部分重复 (FR) 码的构造算法。由 Hadamard 矩阵经过简单变换直接构造 FR 码。随后引入了分组思想，由 8 阶 Hadamard 矩阵构造分组 FR 码 (HGFR)，构造更加简洁直观，实现多故障节点在局部修复组内进行精确无编码修复。理论分析发现，与 RS 码和 SRC 简单再生码相比，设计的 HGFR 码在分布式存储系统节点发生故障时的修复局部性、修复复杂度和修复带宽开销都降低，且修复效率提高，减少了故障节点的修复时间。

关 键 词 分布式存储； 部分重复码； Hadamard 矩阵； 局部修复

中图分类号 TN911.2 文献标志码 A doi:10.12178/1001-0548.2020028

Construction of Fractional Repetition Codes Based on Hadamard Matrix

WANG Jing[1], SUN Wei[1*], HE Ya-jin[1], SHEN Ke-qin[1], ZHANG Xin-nan[1], and LIU Xiang-yang[2]

(1. School of Information Engineering, Chang'an University Xi'an 710064;

2. College of Information and Communication, National University of Defense Technology Xi'an 710106)

Abstract In order to solve the problem of fault node repair in distributed storage system, a construction algorithm of fractional repetition (FR) code is proposed. Specifically, the FR code is constructed directly by Hadamard matrix through simple transformation. Then, the grouping idea is introduced and the 8-order Hadamard matrix is used to construct the grouping FR code, which is more concise and intuitive and can realize the precise non-coding repair of multiple fault nodes in the local repair group. Compared with Reed-Solomon (RS) codes and simple regenerating codes (SRC), theoretical analysis shows that designed FR codes have lower repair locality, repair bandwidth overhead and repair complexity. In addition, this method has high repair efficiency and reduces the repair time of failed nodes.

Key words distributed storage; fractional repetition codes; hadamard matrix; local repair

近年来，由于数据量的快速上升，急需一种适宜的大数据存储系统。分布式存储系统由许多廉价磁盘组成，以其突出优势成为海量数据存储的有效系统，并被广泛部署和使用[1]。但在分布式存储系统中，节点容易发生故障，造成数据丢失。因此，故障节点的快速修复研究成为了分布式存储系统可靠性的重中之重。

目前，分布式存储系统主要通过复制和纠删码策略来恢复节点故障。复制策略中三副本复制最为常见，故障节点修复具有较低的修复带宽开销，但需要存储大量的副本数据，存储开销较大。纠删码策略通过增加校验数据块来确保数据存储的可靠性，实现故障节点修复，且存储开销较小。虽然纠删码弥补了复制策略存储开销大的缺点，但是纠删码在修复故障节点时的修复带宽开销过大[2]。

鉴于复制和纠删码策略存在上述局限性，文献 [3] 将网络编码应用到分布式存储中，提出了再生码的概念，降低了故障节点的修复带宽开销。现在再生码研究重点在最小存储再生 (minimum storage regeneration, MSR) 码和最小带宽再生 (minimum bandwidth regeneration, MBR) 码[4-5]。再生码在修复故障节点时，需要连接大量存活节点以获得较低的修复带宽开销，且在修复过程中涉及有限域运算，计算复杂度相对较高。随后，文献 [6] 提出了局部

收稿日期：2020－02－04；修回日期：2020－12－08

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)

作者简介：王静 (1982－)，女，博士，教授，主要从事网络编码及分布式存储编码等方面的研究.

通信作者：孙伟，E-mail：1277874948@qq.com

修复码 (locally repairable codes, LRC)，使修复过程中需要连接的存活节点数较小，修复带宽开销较低，具有较好的修复局部性。将再生码和局部修复码结合，文献 [7-8] 提出了局部再生码的概念，达到存储−带宽开销的最佳折中。其中，基于系统 MSR 码的局部再生码，故障局部码可以通过相邻局部码进行协作修复[9]。

文献 [10] 提出了一种精确最小带宽再生码——部分重复 (fractional repetition, FR) 码，故障节点修复过程中的计算复杂度和修复带宽开销都有所降低，可以实现故障节点的精确无编码修复。近年来，许多研究人员对 FR 码进行了研究，文献 [11] 利用组合设计来构造 FR 码。随后，学者们又相继给出了几种基于组合结构的 FR 码构造，包括基于射影几何的 FR 码构造[12]、基于正则图的 FR 码构造[13] 及基于可分组设计的 FR 码构造[14]。

现有 FR 码对于故障节点的修复，特别是多节点故障修复，其修复带宽开销和修复局部性较高，同时修复复杂度也较高。文献 [13] 提出了基于正则图构造 FR 码，随后文献 [15] 提出了运用网格构造 FR 码，但其都只能修复单节点故障。基于相对差集的 FR 码构造，能够修复分布式存储系统中的多节点故障，但是随着参数的增大，其节点存储容量和构造复杂度会随之增大[16]。而且常见的 FR 码构造随着系统规模和参数的增大，其节点数或节点容量增大，构造复杂度也会增大。本文提出基于 Hadamard 矩阵构造 FR 码，同时对其进行分组，运用 8 阶 Hadamard 矩阵提出了分组构造 FR(Hadamard grouping fractional repetition, HGFR) 码的一般算法，实现故障节点的局部修复。基于 Hadamard 矩阵分组构造 FR 码可以对多个节点故障进行快速精确无编码修复，有效降低了运算复杂度，同时运用了分组可以实现组内修复，复杂度进一步降低，实现故障节点的快速修复。理论分析发现，与 RS 码和 SRC 简单再生码相比，设计的 FR 码在分布式存储系统节点发生故障时，修复局部性、修复复杂度和修复带宽开销进一步降低，且修复效率提高，减少了故障节点的修复时间。

1 Hadamard 矩阵和部分重复码

1.1 Hadamard 矩阵

Hadamard 矩阵是在工程技术上运用较多的一类矩阵，Hadamard 矩阵是特殊的 1、−1 二元矩阵，具体定义如下：

定义 1[17-18] n阶方阵$\boldsymbol{H}_n$，其元素为 1 或−1，并且满足：

$$\boldsymbol{H}_n\boldsymbol{H}_n^{\mathrm{T}} = n\boldsymbol{I}_n$$

称$\boldsymbol{H}_n$为n阶 Hadamard 矩阵，其中$\boldsymbol{I}_n$是n阶单位矩阵。

若$\boldsymbol{H}_n$的第 1 行和第 1 列全是 1，该$\boldsymbol{H}_n$为 Hadamard 矩阵的标准型。以下所涉及的 Hadamard 矩阵$\boldsymbol{H}_n$均为 Hadamard 标准型矩阵。

Hadamard 矩阵有如下一些性质：

1) 将 Hadamard 矩阵的任意两行 (或两列) 交换，Hadamard 矩阵的任意一行 (或一列) 的所有元素乘以−1，得到的矩阵仍然为 Hadamard 矩阵。

2) 若 $\boldsymbol{H}_n$是 n阶 Hadamard 矩阵 ($n>2$)，则 n是 4 的倍数。

定义 2 [19] 令：

$$\boldsymbol{K}_n = \frac{\boldsymbol{J}_n+\boldsymbol{H}_n}{2}$$

其中$\boldsymbol{J}_n$表示元素全为 1 的n阶矩阵，则得到n阶 0-1 矩阵$\boldsymbol{K}_n$。

性质 1　矩阵$\boldsymbol{K}_n$中除第一行之外，每一行都有$n/2$个 1 和$n/2$个 0。

1.2 FR 码

FR 码是在 MBR 码基础上提出的，典型的 FR 码由两部分组成，外部的 MDS 码和内部的重复码[20]。

定义 3(FR 码)[21] 分布式存储系统中参数为(n,k,d)的部分重复码$C=(\eta,M)$，将数据块复制ρ倍 (即重复度为ρ)，特定地，n个子集的集合$M=\{M_1,M_2,\cdots,M_n\}$，子集中的元素都来自于集合$\eta=\{1,2,\cdots,\theta\}$。

同时应满足以下两个条件：

1) 每个子集的大小均为d；

2) η中的每个元素属于M中的ρ个子集。

上述定义中，数据块是经过 MDS 编码后的数据块。FR 码的实质是将数据块复制ρ倍，然后将其排列到n个节点，同时使相同的数据块不出现在同一个节点上。图 1 为经过(12,9)MDS 编码后构成(12,9,4)FR 码，其中ρ为 2，数据复制 2 倍，每个节点存放 4 个编码数据块。

FR 码修复故障节点时直接从未失效节点中下载丢失的所需数据块，不进行编译码操作即可完成故障节点的迅速修复。FR 码可实现精确无编码修复，修复带宽开销和修复时间较低，修复复杂度较小，同时能够容忍$\rho-1$个节点故障。

N_1	1	2	3	4	N_4	3	7	10	11
N_2	1	5	6	7	N_5	4	8	10	12
N_3	2	6	8	9	N_6	5	9	11	12

图 1　(12,9,4)FR 码

2　基于 Hadamard 矩阵构造 FR 码

本节基于 Hadamard 矩阵构造 FR 码。首先选取一个$4t(t=1,2,3,\cdots)$阶的 Hadamard 矩阵，对其进行简单的变换得到所需要的矩阵；再将矩阵与分布式存储节点和编码数据块相对应，矩阵的行代表存储节点，矩阵中不同的列表示不同的编码数据块。由 Hadamard 矩阵引出 FR 码一般性构造算法，其具体步骤如下：

1) 将原始文件$\boldsymbol{M}$分成k个原始数据块，这里$k\geqslant 2$。对该k个原始数据块采用(n,k)MDS 编码$(n\geqslant k)$，得到n个编码数据块$c_1,\cdots,c_{k-1},c_k,c_{k+1},\cdots,c_n$，$n$个编码数据块包括$k$个原始数据块和$n-k$个校验数据块；

2) 取一个 n+1 阶标准型 Hadamard 矩阵$\boldsymbol{H}_{n+1}$，其中 n+1=1,2，或 n+1=0(mod4)；

3) 根据公式：

$$\boldsymbol{K}'_{n+1}=\frac{\boldsymbol{J}_{n+1}+\boldsymbol{H}_{n+1}}{2} \tag{1}$$

得到 0-1 矩阵$\boldsymbol{K}'_{n+1}(n\geqslant k)$，其中$\boldsymbol{J}_{n+1}$表示元素全为 1 的$n+1$阶矩阵，$\boldsymbol{H}_{n+1}$为标准 Hadamard 矩阵，需满足$n+1$为 4 的倍数；

4) 对 0-1 矩阵$\boldsymbol{K}'_{n+1}$进行变换，将第一行第一列删去得到新矩阵$\boldsymbol{K}_n$；

5) 通过矩阵$\boldsymbol{K}_n$构造 FR 码，具体过程为：

① 矩阵$\boldsymbol{K}_n$中每一行代表一个节点，用矩阵$\boldsymbol{K}_n$中的第i行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，$i=1,2,\cdots,n$；

② 由以下公式构造 FR 码：

$$N_i=\{j:a_{ij}=1\} \tag{2}$$

式中，$j=1,2,\cdots,n$；i表示第i个 FR 节点；a_{ij}表示矩阵第 i 行第 j 列的值；N_i表示 FR 码的存储节点。N_i中包含的数据块为矩阵$\boldsymbol{K}_n$中第i行所有 1 所对应的列数，将列数提取出即得到一个节点所存储的数据块，构成了所需 FR 码。

采用上述 FR 码的一般性构造算法，在包含$n=11$个存储节点的分布式存储系统中，构造 FR 码。选取如式 (3) 所示的 12 阶 Hadamard 矩阵，利用式 (1) 对该 Hadamard 矩阵进行变换，进一步得到所需的 11 阶矩阵$\boldsymbol{K}_{11}$，如式 (4) 所示。采用该$\boldsymbol{K}_{11}$矩阵，按照步骤 5) 构造得到 FR 码，该 FR 码的节点存储结构具体如图 2 所示。其中矩阵的行代表分布式存储系统中的存储节点，矩阵中不同的列表示不同的编码数据块，每个存储节点存储$d=5$个编码块，编码块的重复度$\rho=5$。

$$\boldsymbol{H}_{12}=\begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1\\
1&-1&1&-1&1&1&1&-1&-1&-1&1&-1\\
1&-1&-1&1&-1&1&1&1&-1&-1&-1&1\\
1&1&-1&-1&1&-1&1&1&1&-1&-1&-1\\
1&-1&1&-1&-1&1&-1&1&1&1&-1&-1\\
1&-1&-1&1&-1&-1&1&-1&1&1&1&-1\\
1&-1&-1&-1&1&-1&-1&1&-1&1&1&1\\
1&1&-1&-1&-1&1&-1&-1&1&-1&1&1\\
1&1&1&-1&-1&-1&1&-1&-1&1&-1&1\\
1&-1&1&1&1&-1&-1&-1&1&-1&-1&1\\
1&1&-1&1&1&1&-1&-1&-1&1&-1&-1
\end{bmatrix} \tag{3}$$

$$\boldsymbol{K}_{11}=\begin{bmatrix}
0&1&0&1&1&1&0&0&0&1&0\\
0&0&1&0&1&1&1&0&0&0&1\\
1&0&0&1&0&1&1&1&0&0&0\\
0&1&0&0&1&0&1&1&1&0&0\\
0&0&1&0&0&1&0&1&1&1&0\\
0&0&0&1&0&0&1&0&1&1&1\\
1&0&0&0&1&0&0&1&0&1&1\\
1&1&0&0&0&1&0&0&1&0&1\\
1&1&1&0&0&0&1&0&0&1&0\\
0&1&1&1&0&0&0&1&0&0&1\\
1&0&1&1&1&0&0&0&1&0&0
\end{bmatrix} \tag{4}$$

N_1	2	4	5	6	10	N_2	3	5	6	7	11
N_3	1	4	6	7	8	N_4	2	5	7	8	9
N_5	3	6	8	9	10	N_6	4	7	9	10	11
N_7	1	5	8	10	11	N_8	1	2	6	9	11
N_9	1	2	3	7	10	N_{10}	2	3	4	8	11
N_{11}	1	3	4	5	9						

图 2　FR 码的节点存储结构图

当一个节点发生故障时，可以运用其他节点对故障节点进行修复，在其他节点中找到并下载含有故障节点的数据块，故障节点即完成了修复，且该过程不需要进行编码译码操作。对于图 2 中的 FR 码，若节点N_1发生故障，可以分别从存活节点N_3中

下载数据块 4 和 6，从存活节点N_4中下载数据块 2 和 5，以及存活节点N_5中下载数据块 10，实现节点N_1的修复。对于两个节点发生故障的情况，与一个节点故障修复的方法相同。

3 基于 Hadamard 矩阵构造分组 FR 码

运用 Hadamard 矩阵构造 FR 码发现，随着 Hadamard 矩阵阶数的增加，FR 码的重复度和节点存储容量也会随之增加，从而导致存储开销也相应地增加，故障节点修复性能受到一定限制。为此，本节引入分组构造的思想构造部分重复码。当 Hadamard 矩阵的阶数为 12 时构造的 FR 码的重复度为 5，重复度较大；当阶数为 8 时 FR 码的重复度降为 3，更满足实际存储需求，所以本文利用 8 阶标准 Hadamard 矩阵构造分组 FR 码。

3.1 基于 Hadamard 矩阵构造分组部分重复码

对分布式存储系统节点进行分组，并利用 8 阶 Hadamard 矩阵构造分组 FR 码 (hadamard grouping fractional repetition, HGFR)。分组 FR 码的具体构造算法如下：

1) 取一个 8 阶标准 Hadamard 矩阵$\boldsymbol{H}_8$；由上节构造部分重复码方法的步骤 1)～步骤 4) 得到新矩阵$\boldsymbol{K}_7$，并利用新矩阵$\boldsymbol{K}_7$构造 FR 码；

2) 记原始文件 M 包含s个原始数据块，将原始数据块进行分组，得到t个局部修复组，每组分配k= 5个原始数据块。包含以下几种情况：

① 如果s可以被k= 5整除，即$s = tk$，此时每个局部修复组内首先进行 (7, 5)MDS 码编码，然后采用步骤 3) 中的矩阵$\boldsymbol{K}_7$构造 FR 码；

② 如果s不能被k= 5整除，即$s = tk + m$且 m=1，前 t–1 个局部修复组采用 (7, 5)MDS 码以及矩阵$\boldsymbol{K}_7$构造 FR 码，第 t 个局部修复组采用 (7,6)MDS 码以及矩阵$\boldsymbol{K}_7$构造 FR 码；

③ 若$s = tk + m$且$m = 2$，前$t - 2$个局部修复组采用 (7, 5)MDS 码以及矩阵$\boldsymbol{K}_7$构造 FR 码，第 t–1 个和第 t 个局部修复组采用 (7, 6)MDS 码以及矩阵$\boldsymbol{K}_7$构造 FR 码；

④ 若$s = tk + m$且$m = 3$或者 4 时，前 t–1 个局部修复组采用 (7,5)MDS 码以及矩阵$\boldsymbol{K}_7$构造 FR 码，第 t 个局部修复组采用 (7,m)MDS 码以及矩阵$\boldsymbol{K}_7$构造 FR 码。

以包含 $s = 11$个数据块的原始文件为例，$s = 2k + 1$，这里 m=1，满足情况②。第 1 个局部修复组采用 (7,5)MDS 码以及矩阵$\boldsymbol{K}_7$构造 FR 码，第 2 个局部修复组采用 (7,6)MDS 码以及矩阵$\boldsymbol{K}_7$构造 FR 码。构造的分组 FR 码如图 3 所示。

N_1	2	4	6	N_8	9	11	13
N_2	1	4	5	N_9	8	11	12
N_3	3	4	7	N_{10}	10	11	14
N_4	1	2	3	N_{11}	8	9	10
N_5	2	5	7	N_{12}	9	12	14
N_6	1	6	7	N_{13}	8	13	14
N_7	3	5	6	N_{14}	10	12	13

图 3 分组 FR 码的节点存储结构图

3.2 故障节点修复

下面考虑分布式存储系统采用基于 Hadamard 矩阵的分组 FR 码，其故障节点的修复。考虑到 3 个以上节点同时故障的概率很低，本文只考虑分布式存储系统中 1 个、2 个或者 3 个节点故障，且在同一个或者不同局部修复组内的情况。具体故障节点修复过程如下：

1) 当分布式存储系统只有 1 个节点故障，此时只需考虑在一个局部修复组内对单一故障节点进行修复。具体地，在该局部修复组内其他存活节点中找到含有故障节点的数据块，直接下载故障节点丢失的数据块，即可完成故障节点修复，该过程不需要进行编码操作。如图 2 所示，如第一个局部修复组中第一个节点发生故障，可以下载节点N_2、N_4、N_6中的 2、4 和 6 这 3 个数据块完成第一个节点的修复。

2) 当分布式存储系统中有 2 个节点同时发生故障时，存在以下两种情况：① 2 个故障节点在同一局部修复组内，可以从剩余的 5 个存活节点中下载故障节点所含有的数据块，完成故障节点的快速修复；② 发生故障的 2 个节点不在同一局部修复组，则需要在两个修复组内分别对其组内的故障节点进行修复。在图 2 中，如节点N_2和节点N_{10}发生故障导致数据块丢失，可以在第一个局部修复组内从节点N_1、N_4、N_5中分别下载数据块 4、1 和 5 完成节点N_2的修复，在第二个局部修复组内从节点N_9、N_{11}、N_{13}下载数据块 11、10 和 14 完成节点N_{10}的快速修复。

3) 当分布式存储系统中有 3 个节点同时发生故障，存在以下两种情况：①发生故障的 3 个节点不在同一局部修复组，该情况下的故障节点修复与前

面两种修复过程完全一样，只需直接从局部修复组中下载所需的数据块；②发生故障的 3 个节点在同一局部修复组中，如果 3 个故障节点中没有同时含有同一个数据块，则可以直接从其他存活节点中下载丢失的数据块，完成故障节点的修复；否则，需要运用 MDS 编码进行修复。如图 2 中节点N_1、N_2、N_3发生故障，剩余存活节点无法直接修复，需要从存活节点下载数据块进行 MDS 编码修复数据块 4，即完成故障节点N_1、N_2、N_3的精确修复。

4　性能分析

本节对提出的基于 Hadamard 矩阵构造分组 FR 码进行性能分析，修复带宽开销、修复局部性、修复复杂度为分析的主要性能，并与 SRC 简单再生码以及 RS 码进行比较。取文件大小M=1 000 Mb，SRC 简单再生码的子文件数$f=4$。

4.1　修复带宽开销

当节点发生故障时，修复故障节点被下载的数据量大小称为修复带宽开销。在分布式存储系统中，原文件大小为M，当发生单节点故障时，若采用(n,k)RS 码，则需要下载全部的原文件来修复故障节点，所以采用(n,k)RS 码修复单节点故障的修复带宽开销为文件大小M；对于(n,k,f)SRC 简单再生码，每个节点存储$f+1$个数据块，每个数据块大小为M/fk，当一个数据块发生故障，f个数据块被下载进行修复，所以 1 个节点发生故障的修复带宽开销为$(f+1)M/k$；如果采用基于 Hadamard 矩阵的分组 FR 码，每个数据块大小为M/k，每个节点含有 3 个数据块，所以基于 Hadamard 矩阵的分组 FR 码的单节点故障的修复带宽开销为$3M/k$。单节点故障的 3 种编码方式修复带宽开销对比如图 4 所示。

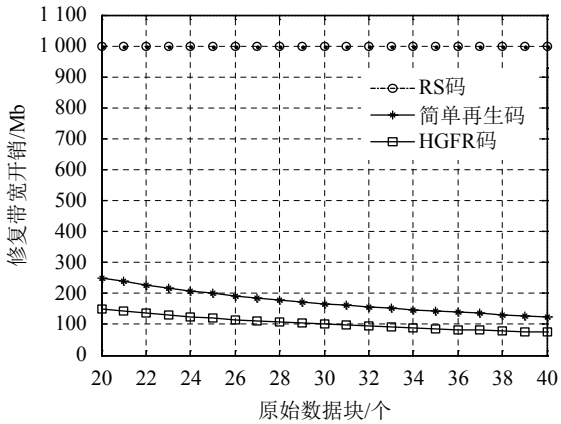

图 4　单节点故障时的修复带宽开销

当 2 个节点发生故障时，若采用(n,k)RS 码，仍然需要下载全部的原文件来修复故障节点，所以(n,k)RS 码修复 2 个故障节点的修复带宽开销仍为M；对于(n,k,f)SRC 简单再生码，如果 2 个故障节点数大于$f-1$，则 2 个节点发生故障的修复带宽开销为$2(f+1)M/k$，如果 2 个故障节点数小于$f-1$，需要先恢复原文件，所以修复带宽开销为M，这里f=4，所以 2 个节点故障时，SRC 简单再生码的修复带宽开销为 M；如果采用基于 Hadamard 矩阵的分组 FR 码，2 个故障节点在同一修复组时，修复带宽开销为$3M/k$，不在同一修复组时，修复带宽开销为$6M/k$。修复带宽开销对比如图 5 所示。

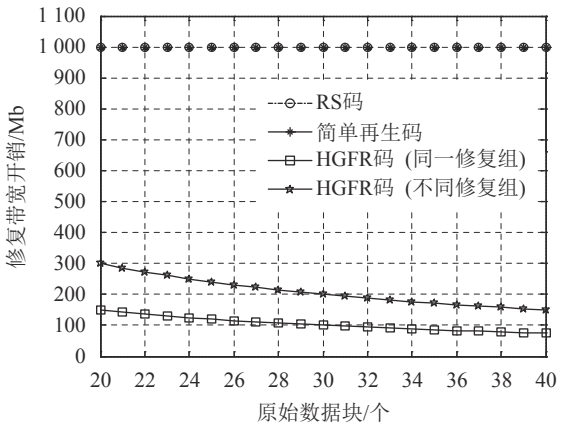

图 5　2 个节点故障时的修复带宽开销

从图 4 和图 5 可以看出，基于 Hadamard 矩阵的分组 FR 码，单节点和两节点故障时的修复带宽开销性能明显优于 RS 码和 SRC 简单再生码，且在图 5 中 RS 码和 SRC 简单再生码由于修复带宽开销一样，所以曲线重合。

4.2　修复局部性

修复局部性是指故障节点修复过程中需要连接的存活节点数目，即修复过程中的磁盘 I/O 开销。首先对于单节点失效，简单再生码修复时需要连接$2f$个存活节点，所以局部修复性为$2f$；RS 码发生单节点故障时，需要连接k个节点恢复原文件修复故障节点，所以 RS 码修复局部性为k；基于 Hadamard 矩阵的分组 FR 码，连接 3 个未发生故障的节点来恢复单个节点故障，所以其修复局部性为 3。单节点故障时的修复局部性对比如图 6 所示。

对于 2 个节点发生故障，SRC 简单再生码需要先恢复原文件才可以修复故障节点，所以 SRC 简单再生码的修复局部性为k；对于 RS 码，仍然需要恢复原文件，所以修复 2 个节点的修复局部性也为k；基于 Hadamard 矩阵的分组 FR 码，分两种

情况：1) 如果一个修复组中发生 2 个节点故障，从 3 个未失效节点中下载所需数据，修复局部性为 3；2) 2 个故障节点不在同一个局部修复组内，其修复两个故障节点需要连接 6 个存活节点，修复局部性为 6。

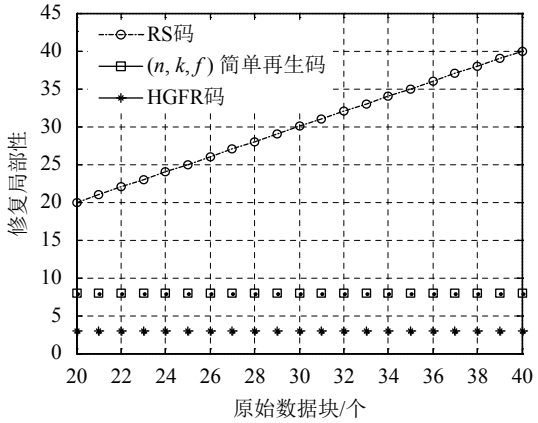

图 6 单节点故障时的修复局部性

修复故障节点时，可以看出基于 Hadamard 矩阵的分组 FR 码的修复局部性，优于简单再生码和 RS 码的修复局部性。

4.3 修复复杂度

本文提出的基于 Hadamard 矩阵的分组 FR 码，当发生节点故障时，可以直接从其他存活节点下载数据块进行修复，无需进行编码运算。对于 RS 码修复故障节点需要先恢复原文件，之后再编码生成所需要的数据块，编码数据块由k个数据块在有限域$GF(q)$上运算得到。所以 RS 码在修复单节点故障时需要k^2+k次有限域乘法运算和k^2-1次有限域加法运算。对于 SRC 简单再生码，其修复一个故障节点需要进行$(f-1)(f+1)$次异或运算。可以看出，本文构造的 FR 码修复复杂度明显低于 RS 码和 SRC 简单再生码的修复复杂度，可以快速修复故障节点，修复时间减少。

表 1 给出了分布式存储系统中，RS 码、简单再生码和基于 Hadamard 矩阵的分组 FR 码分别在修复带宽开销、修复局部性及节点存储开销三方面的性能比较。从表 1 可以看出，基于 Hadamard 矩阵的分组 FR 码其修复局部性和带宽开销较低。而且，基于 Hadamard 矩阵的分组 FR 码在修复节点故障时可以实现精确无编码修复，运算复杂度较低，修复时间减少，可靠性提高。

表 1 3 种编码方案的故障节点修复性能比较

开销		简单再生码	RS码	基于Hadamard矩阵的分组FR码
节点存储开销		$(f+1)M/k$	M/k	$3M/k$
修复带宽开销	单节点故障	$(f+1)M/k$	M	$3M/k$
	两节点故障	M	M	$3M/k$或$6M/k$
修复局部性	单节点故障	$2f$	k	3
	两节点故障	k	k	3或6

5 结 束 语

现有传统编码方式对于故障节点的修复，特别是对多故障节点的修复，其修复带宽开销和修复局部性能受限，同时修复复杂度较高。为此，本文提出了基于 Hadamard 矩阵的 FR 码，同时对其进行分组，运用 8 阶 Hadamard 矩阵提出了分组 FR 码的一般性构造算法。理论分析发现，基于 Hadamard 矩阵的分组 FR 码可以对多故障节点进行快速精确无编码修复，有效降低了运算复杂度，同时运用了分组可以实现组内修复，复杂度进一步降低。与 RS 码和 SRC 简单再生码相比，发生故障时的修复局部性、修复复杂度和修复带宽开销进一步降低，且修复效率提高，减少了故障节点的修复时间。

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] WANG Yi-jie, XU Fang-liang, PEI Xiao-qiang. Research on erasure code-based fault-tolerant technology for distributed storage[J]. Chinese Journal of Computers, 2017, 40(1): 238-257.

[3] 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.

[4] YANG Bin, TANG Xiao-hu, LI Jie. A Systematic piggybacking design for minimum storage regenerating codes[J]. IEEE Transactions on Information Theory, 2015, 61(11): 5779-5786.

[5] RRWAT A S, SILBERSTEIN N, KOYLUOGLU O O, et al. Optimal locally repairable codes with local minimum storage regeneration via rank-metric codes[C]//2013 Information Theory and Applications Workshop (ITA). [S.l.]: IEEE, 2013: 1-8.

[6] PAPAILIOPOULOS D S, DIMAKIS A G. Locally repairable codes[C]//IEEE International Symposium on Information Theory (ISIT). [S.l.]: IEEE, 2012: 2771-2775.

[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] RAWAT A S, KOYLUOGLU O O, SILBERSTEIN N, et al. Optimal locally repairable and secure codes for distributed storage systems[J]. IEEE Transactions on Information Theory, 2014, 60(1): 212-236.

[9] WANG Jing, YAN Zhi-yuan, 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.

[10] ROUAYHEB S E, RAMCHANDRAN K. Fractional repetition codes for repair in distributed storage systems[C]//Proceedings of 2010 48th Annual Allerton Conference on Communication, Control, and Computing. Allerton, IL, USA: IEEE, 2010: 1510-1517.

[11] OLMEZ O, RAMAMOORTHY A. Repairable replication-based storage systems using resolvable designs[C]//Proceedings of 2012 50th Annual Allerton Conference on Communication, Control, and Computing. Monticello, IL, USA: IEEE, 2012: 1174-1181.

[12] KOO J C, GILL J T. Scalable constructions of fractional repetition codes in distributed storage systems[C]// Proceedings of 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton). Monticello, IL, USA: IEEE, 2011: 1366-1373.

[13] SILBERSTEIN N, ETZION T. Optimal fractional repetition codes based on graphs and designs[J]. IEEE Transactions on Information Theory, 2015, 61(8): 4164-4180.

[14] ZHU Bing, LI Hui, SHUM K W. Repair efficient storage codes via combinatorial configurations[C]//Proceedings of IEEE International Conference on Big Data. [S.l.]: IEEE, 2014: 80-81.

[15] OLMEZ O, RAMAMOORTHY A. Fractional repetition codes with flexible repair from combinatorial designs[J]. IEEE Transactions on Information Theory, 2016, 62(4): 1565-1591.

[16] KIM Y S, PARK H, NO J S. Construction of new fractional repetition codes from relative difference sets with $\lambda = 1$[J]. Entropy, 2017, 19(10): 563.

[17] SANGIAMCHIT P, FAKCHAROENPHOL J. Practical differential privacy for location data aggregation using a hadamard matrix[C]//The 16th International Joint Conference on Computer Science and Software Engineering (JCSSE). Chonburi, Thailand: [s.n.], 2019: 79-84.

[18] PETR L, KOCHEN S. Sets and Hadamard matrices[J]. Theoretical Computer Science, 2019, 800: 142-145.

[19] 秦小二, 胡双年, 姜灏, 等. 用 Hadamard 矩阵构造线性码[J]. 四川大学学报 (自然科学版), 2015, 52(6): 1221-1224.
QIN Xiao-er, HU Shuang-nian, JIANG Hao, et al. Constructing linear codes by Hadamard matrices[J]. Journal of Sichuan University (Natural Science Edition), 2015, 52(6): 1221-1224.

[20] 朱兵, 李挥, 陈俊, 等. 基于可分组设计的部分重复码研究[J]. 通信学报, 2015, 36(2): 102-109.
ZHU Bing, LI Hui, CHEN Jun, et al. Research on fractional repetition codes based on group divisible designs[J]. Journal on Communications, 2015, 36(2): 102-109.

[21] SU Yi-Sheng. Pliable fractional repetition codes for distributed storage systems: Design and analysis[J]. IEEE Transactions on Communications, 2018, 66(6): 2359-2375.

编　辑　税　红