一种云存储中基于干扰对齐的多节点精确修复方法

谢显中,黄 倩,王柳苏,马 彬
(重庆邮电大学宽带接入网络研究所,重庆 400065)

摘 要: 本文提出了一种基于干扰对齐的满足 MDS 性质的多节点精确修复码(MMSR).首先利用柯西矩阵构造 MMSR 码的生成矩阵,使其适用干扰对齐技术,并同时修复多个节点.然后讨论了 MMSR 码同步修复和异步修复方案的优缺点.最后证明了 MMSR 码的 MDS 性质,并通过一个(7,3,5)-MMSR 码的数据重建方案验证了 MMSR 码的 MDS 性质和可行性.

关键词: 云存储; 多节点修复; 干扰对齐; 再生码; 柯西矩阵
中图分类号: TP302.8 **文献标识码:** A **文章编号:** 0372-2112 (2014)10-1873-09
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2014.10.001

A Multi-Node Exact Repair Method in Cloud Storage Based on Interference Alignment

XIE Xian-zhong, HUANG Qian, WANG Liu-su, MA Bin
(*Institute of Broadband Access Technologies, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*)

Abstract: We proposed a Multi-node exact Minimum Storage Regenerating code (MMSR code) based on interference alignment which is satisfying MDS nature. First, we designed the generator matrix of the storage nodes of MMSR code by Cauchy matrix, for the purposes that using interference alignment technology and repair multi-node simultaneously. Then we discussed the advantages and disadvantages of the synchronous repair mode and asynchronous repair mode. Finally, we proved the MDS nature of MMSR code and proposed a specific data reconstruction scheme for the (7,3,5)-MMSR code to verify the MDS nature and feasibility of MMSR code.

Key words: cloud storage; multi-node repair; interference alignment; regenerating code; Cauchy matrix

1 引言

云存储(Cloud Storage)系统[1~3]凭借冗余信息保证其高可靠性.冗余信息的产生由简单复制发展到纠删码(Erasure Coding)[4],再到修复带宽更优的再生码(Regenerating Code)[5].再生码不但保持了纠删码的 MDS(Maximum Distance Separable)性质,同时降低了数据修复过程中的通信开销.

在精确修复中,新生节点的内容、功能和约束条件与原失效节点完全一致.精确修复可被看作非多播网络问题,其割集的理论最优边界值难以达到[6],仅使用线性网络编码并不能解决相关问题[7].为此,云存储中引入了干扰对齐(IA, Interference Alignment)技术[8].利用干扰对齐技术在节点修复时将干扰数据压缩到较小的空间维度,增大期望数据的维度,从而在确保高可靠性的同时降低修复开销,节约云存储成本.文献[9]提出了一种云存储中的干扰对齐技术;并证明了为使修复带宽最小,当超过三组单播类型的网络或每组的最小割集大于 1 时,干扰对齐是必不可少的.在 $k=2$ 的情况下,文献[10]将干扰对齐同时运用于系统节点和冗余节点的精确修复,但需要较大的域值,复杂度高.文献[11]给出的 MISER 码(MDS Interference-aligning Systematic Exact-Regenerating)说明了干扰对齐运用于再生码中的必要性,但该方案只适用于单系统节点失效的情况.

目前关于多失效节点的精确修复问题的文献,要么未设计详细的修复方案,只是提出一种理论接近最优的模型[12,13];要么只讨论了功能修复[14].Kenneth 等在文献[15]提出了一种与再生码相比,能进一步减少修复带宽的 MSCR 码(Minimum Storage Coordinated Regenerating),该码对纠删码中的节点数据进行再分块,各节点分工

收稿日期:2013-07-08;修回日期:2013-12-03;责任编辑:孙瑶
基金项目:国家自然科学基金(No.61271259,No.61301123);重庆市自然科学基金(No.CSTC2010BB2415,No.CTSC2011jjA40006);重庆市教委科学技术研究项目(No.KJ120501,No.KJ110530)

合作完成修复过程;Scouarnec 等在文献[16]中给出了另一种 MSCR 码方案,对参数要求与文献[15]相同,并导出了详细的设计过程.但是 MSCR 码只针对 $k=2$ 的情况有效,实际应用价值不大,同时没有采用干扰对齐技术.

根据以往存在的问题,本文提出了基于干扰对齐的多节点最小存储精确再生码(MMSR,Multi-node exact Minimum Storage Regenerating Code).本文第二部分详述了基于干扰对齐的节点修复模型,并介绍了同步修复和异步修复的一般模型;MMSR 码的一般构造和 MDS 性质证明将在第三部分给出;本文第四部分详细介绍了 MMSR 码的同步修复和异步修复方案;为了验证 MMSR 码的 MDS 性质和可行性,一个(7,3,5)-MMSR 码的具体数据重建方案将在第五部分给出;最后本文第六部分是对 MMSR 码的数值分析.

为方便,本文用元素表示云存储中的最小存储单位,即基本计算单位;将被选中用于修复失效节点的未失效节点称为帮助节点.本文中 MMSR 码的参数集定义如下:

$$\{[n,k,d],(\beta,\alpha,M)\}$$

其中,n 为存储节点数,k 为系统节点数(也是重建数据时,用户连接的节点数),d 为帮助节点数,$n-k$ 为冗余节点数;M 是源文件的大小,α 是存储节点的存储容量,β 是修复带宽(为修复失效节点,新节点需从帮助节点中下载一定量的数据,这些下载的数据需要在网络上传输,占用了网络带宽,因而称为修复带宽).

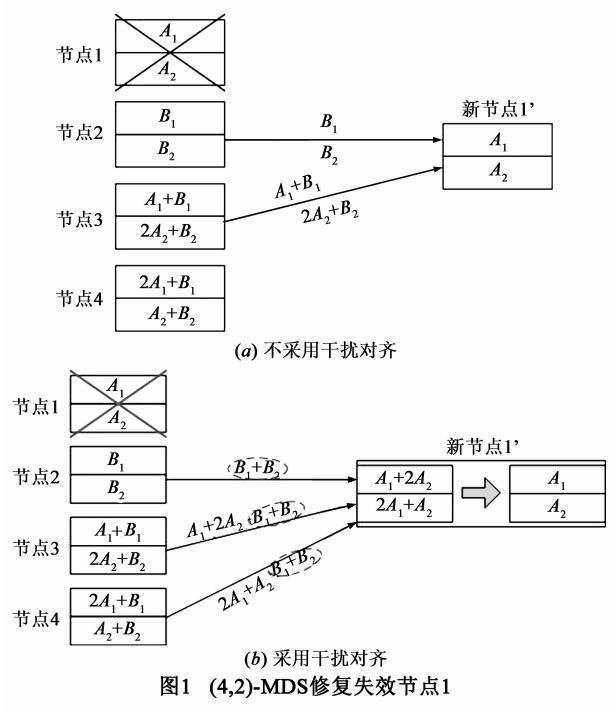
2 节点修复的一般模型

2.1 基于干扰对齐的节点修复模型

图 1 为(4,2)-MDS 修复模型.模型中有 4 个存储节点,不妨假设前两个节点为系统节点,后两个节点为冗余节点(校验节点),节点 1 失效.为修复节点 1,A_1、A_2 为期望数据,B_1、B_2 为干扰数据.其中,图 1(a)为不采用干扰对齐的修复模型,图 1(b)是采用干扰对齐的修复模型.

图 1(a)未用干扰对齐,在修复节点 1 时,从节点 2 和节点 3 传输了 4 个元素(B_1、B_2、A_1+B_1 和 $2A_2+B_2$)到新节点,然后在新节点处利用 4 个方程解出 4 个数据(A_1、A_2、B_1 和 B_2).图 1(b)在帮助节点数据传输到新节点前,先在发送端进行预编码处理,即将帮助节点的元素编码为 $2A_1+A_2+B_1+B_2$、$A_1+2A_2+B_1+B_2$ 和 B_1+B_2;这三个元素被传输到新节点后,在新节点处进行干扰消除,即把 B_1+B_2 从 $A_1+2A_2+B_1+B_2$ 和 $2A_1+A_2+B_1+B_2$ 中整体消去,解出期望数据 A_1、A_2.

由图 1 可见,在云存储环境中,修复失效节点时用干扰对齐技术把干扰数据对齐在同一个方向上后消除,不但可以减小修复带宽(传输数据减少),还降低了解码代价.



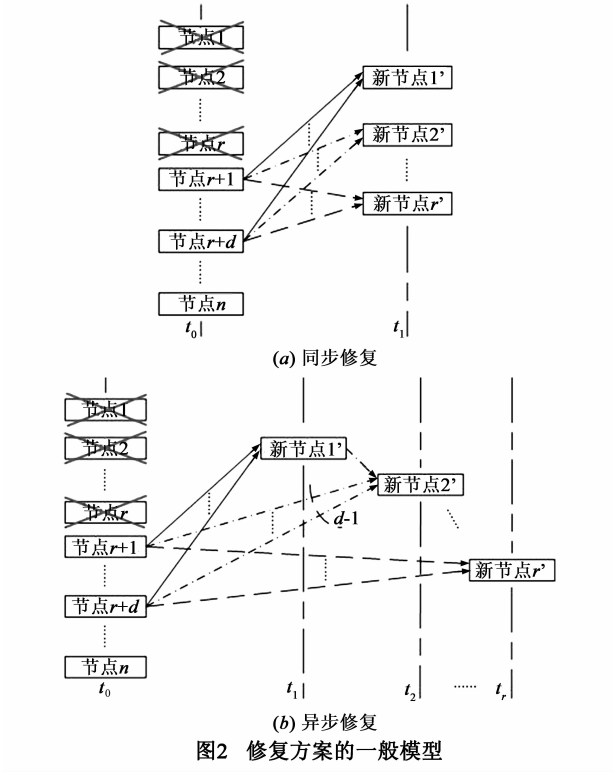
(a) 不采用干扰对齐

(b) 采用干扰对齐

图1 (4,2)-MDS修复失效节点1

2.2 修复方案的一般模型

图 2 是同步修复和异步修复的一般模型.模型中有 n 个节点,假设前 r 个节点失效.图 2(a)采用同步修复方案,系统在 t_1 时刻同时修复 r 个失效节点,并且 r 个失效



(a) 同步修复

(b) 异步修复

图2 修复方案的一般模型

节点的帮助节点集相同.图 2(*b*)采用异步修复方案,系统分别在 $t_1,t_2,\cdots,t_r$ 时刻修复 r 个节点,且每修复完一个失效节点后根据当前网络状况改变帮助节点集.

对比两种修复方式可以看出,同步修复在修复多失效节点过程中,因为不用改变帮助节点集的选择,节省了节点选择算法的代价,但单节点修复过程较复杂,运算量较大,增大了计算成本,降低了干扰对齐性能.异步修复在修复多失效节点时,每修复完一个节点后需要适当调整帮助节点集的选择.因为在当前网络环境允许的情况下,帮助节点中每增加一个系统节点(相应的减少一个冗余节点),即可减少计算一个干扰数据,所以异步修复保证了较小的单节点修复能耗,提高干扰对齐性能,逐步降低计算成本,更适应于多变的无线网络环境中.

3　MMSR 码的构造

3.1　MMSR 码的一般构造

文献[17]根据网络编码的割集原理推导出 MSR(Minimum Storage Regenerating)码的最小存储容量 α_{MSR} 和最小修复带宽 β_{MSR}:

$$(\alpha_{\mathrm{MSR}},\beta_{\mathrm{MSR}})=\left(\frac{M}{k},\frac{M}{k(d-k+1)}\right) \tag{1}$$

根据文献[11]的分析,若能构造出 $\beta=1$ 的 MSR 码,则可构造出任意 β 大小的 MSR 码.这里不妨取 $\beta=1$,由式(1)可得:

$$d=\alpha+k-1 \tag{2}$$

因此,MMSR 码的参数约束条件为:$2k\leqslant n$,$2k-1\leqslant d\leqslant n-1$,$k\leqslant\alpha$.

根据以上参数的约束条件,下面给出具有 MDS 性质的 MMSR 码的一般性编码方案.考虑大小为 M 的用户数据 $\boldsymbol{A}$,为方便叙述,将用户数据 $\boldsymbol{A}$ 均分为大小为 α 的 k 部分,记为 $\boldsymbol{A}_i(i=1,2,\cdots,k)$,则用户数据 $\boldsymbol{A}$ 可表示为:

$$\boldsymbol{A}=[\boldsymbol{A}_1\quad\boldsymbol{A}_2\quad\cdots\quad\boldsymbol{A}_k]^{\mathrm{T}}=[a_1\quad a_2\quad\cdots\quad a_{k\alpha}]^{\mathrm{T}} \tag{3}$$

这里上标$^{\mathrm{T}}$表示矩阵的转置(下同).

由系统模型图 1 可以看出,云存储是否可以高效、可靠的修复节点和重建数据,都依赖于节点生成矩阵的设计方案.

设矩阵 $\boldsymbol{G}^{(m)}$ 是节点 m 的生成矩阵(Generator Matrix),矩阵 $\boldsymbol{G}_i^{(m)}$ 是矩阵 $\boldsymbol{G}^{(m)}$ 的第 $i(1\leqslant i\leqslant k)$ 部分子矩阵.则 $M\times\alpha$ 阶生成矩阵 $\boldsymbol{G}^{(m)}$ 表示为:

$$\boldsymbol{G}^{(m)}=[\boldsymbol{G}_1^{(m)\mathrm{T}}\quad\boldsymbol{G}_2^{(m)\mathrm{T}}\quad\cdots\quad\boldsymbol{G}_k^{(m)\mathrm{T}}]^{\mathrm{T}} \tag{4}$$

则节点 m 中所存储元素(用户数据 $\boldsymbol{A}$ 中的连续 α 个数据)的线性组合为:

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{G}^{(m)}=\sum_{i=1}^{k}\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{G}_i^{(m)} \tag{5}$$

式(5)的每一列都是用户数据 $\boldsymbol{A}$ 的线性组合,表示了节点 m 中存储的一个元素.

3.1.1　系统节点的构造

当节点 $m(1\leqslant m\leqslant k)$ 为系统节点时,节点 m 生成节点的第 i 部分 $\boldsymbol{G}_i^{(m)}$ 表示为:

$$\boldsymbol{G}_i^{(m)}=\begin{cases}\boldsymbol{I}_\alpha, i=m\\ \boldsymbol{0}_\alpha, i\neq m\end{cases}\quad\forall\, i\in\{1,2,\cdots,k\} \tag{6}$$

其中,$\boldsymbol{I}_\alpha$ 为 α 阶的单位矩阵.通过式(4)~(6)的计算,节点 m 中存放着用户数据 $\boldsymbol{A}$ 的第 m 部分的数据 $\boldsymbol{A}_m$.

3.1.2　冗余节点的构造

定义 1　(柯西矩阵[18])　若 $m\times n$ 阶矩阵 $\boldsymbol{C}$ 的元素为 $c_{ij}=\dfrac{1}{(x_i-y_j)}$,$(1\leqslant i\leqslant m,1\leqslant j\leqslant n)$,则矩阵 $\boldsymbol{C}$ 是柯西(Cauchy)矩阵.其中 x_i,y_j 是有限域 $\mathbb{F}_q$ 上满足 $x_i-y_j\neq 0,(1\leqslant i\leqslant m,1\leqslant j\leqslant n)$ 的单射序列($x_i\neq x_j,y_i\neq y_j$).

考虑柯西矩阵的任意子矩阵都是满秩的特性,本文选取柯西矩阵构造冗余节点的生成矩阵.当节点 m $(k+1\leqslant m\leqslant n)$ 为冗余节点,由式(5)可得到冗余节点的生成矩阵:

$$\boldsymbol{G}^{(m)}=\begin{bmatrix}
c_1^{(m)} & 0 & 0 & \cdots & 0\\
c_2^{(m)} & c_{\alpha+1}^{(m)} & 0 & \cdots & 0\\
c_3^{(m)} & 0 & c_{\alpha+1}^{(m)} & \cdots & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
c_\alpha^{(m)} & 0 & 0 & \cdots & c_{\alpha+1}^{(m)}\\
c_\alpha^{(m)} & c_1^{(m)} & 0 & \cdots & 0\\
0 & c_2^{(m)} & 0 & \cdots & 0\\
0 & c_3^{(m)} & c_{\alpha+2}^{(m)} & \cdots & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
0 & c_\alpha^{(m)} & 0 & \cdots & c_{\alpha+2}^{(m)}\\
c_{\alpha+2}^{(m)} & 0 & c_1^{(m)} & \cdots & 0\\
0 & c_{\alpha+2}^{(m)} & c_2^{(m)} & \cdots & 0\\
0 & 0 & c_3^{(m)} & \cdots & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & c_\alpha^{(m)} & \cdots & c_{\alpha+3}^{(m)}\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
c_{\alpha+k-1}^{(m)} & 0 & 0 & \cdots & c_1^{(m)}\\
0 & c_{\alpha+k-1}^{(m)} & 0 & \cdots & c_2^{(m)}\\
0 & 0 & c_{\alpha+k-1}^{(m)} & \cdots & c_3^{(m)}\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & 0 & \cdots & c_\alpha^{(m)}
\end{bmatrix} \tag{7}$$

在式(7)中,元素 $c_i^{(m)}$ 由式(8)的柯西矩阵确定.

$$\boldsymbol{C}=\begin{bmatrix}
c_1^{(k+1)} & c_1^{(k+2)} & \cdots & c_1^{(n)}\\
\vdots & \vdots & \ddots & \vdots\\
c_\alpha^{(k+1)} & c_\alpha^{(k+2)} & \cdots & c_\alpha^{(n)}\\
c_{\alpha+1}^{(k+1)} & c_{\alpha+1}^{(k+2)} & \cdots & c_{\alpha+1}^{(n)}\\
\vdots & \vdots & \ddots & \vdots\\
c_{\alpha+k-1}^{(k+1)} & c_{\alpha+k-1}^{(k+2)} & \cdots & c_{\alpha+k-1}^{(n)}
\end{bmatrix} \tag{8}$$

其中,矩阵 $\boldsymbol{C}$ 每一项 $c_i^{(m)}$ 的上标 m 表明该项是节点 m 的生成矩阵系数,前 α 行确定冗余节点中期望数据部

分的参数,后 $k-1$ 行确定冗余节点中干扰数据部分的参数.为保证冗余节点在修复节点和重建数据时能准确解出全部期望数据,就要确保各冗余节点对应元素的独立性.

冗余节点 m 的生成矩阵的设计思想是:当修复系统节点 $l(1\leqslant l\leqslant k)$时,只需从各帮助节点输出第 l 个元素给失效节点 l 对应的新节点 l' 即可.因此矩阵 $\boldsymbol{G}^{(m)}$ 的第 l 部分的第 l 列为柯西矩阵的相应元素.为了将冗余节点中每个元素的干扰数据对齐到尽量少的方向上,矩阵 $\boldsymbol{G}^{(m)}$ 的第 l 列除第 l 部分外,其余部分均只有一个非零元素.并为达到多节点修复功能,每个部分的非零项均不同.

3.2 MMSR 码的 MDS 性

MDS 码要求用户或 DC(Data Collector)可以从任意 k 个节点中重建出大小为 M 的用户数据.下面首先证明由任意 k 个节点的生成矩阵 $\boldsymbol{G}^{(m)}$ 以列向量方式组成的矩阵的非奇异性.

定理 1 (非奇异性) 若矩阵 $\boldsymbol{G}^{(m)}$ 由式(4)给出,则由任意 k 个生成矩阵所组成的矩阵 $\boldsymbol{S}_1=[\boldsymbol{G}^{(i)}\ \boldsymbol{G}^{(i+j_1)}\ \boldsymbol{G}^{(i+j_2)}\ \cdots\ \boldsymbol{G}^{(i+j_{k-1})}]$为非奇异矩阵.

证明:设 $\delta_1,\delta_2,\cdots,\delta_x$ 为 DC 连接的 x 个冗余节点;$\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_{k-x}(\varepsilon_1\leqslant\cdots\leqslant\varepsilon_{k-x})$为 DC 连接的 $k-x$ 个系统节点;$\varphi_1,\varphi_2,\cdots,\varphi_x(\varphi_1\leqslant\cdots\leqslant\varphi_x)$为剩余的未被 DC 连接的 x 个系统节点.则有:

$$\boldsymbol{S}_1=\boldsymbol{R}_1=[\boldsymbol{G}^{(\varepsilon_1)}\ \cdots\ \boldsymbol{G}^{(\varepsilon_{k-x})}\ \boldsymbol{G}^{(\delta_1)}\ \cdots\ \boldsymbol{G}^{(\delta_x)}] \tag{9}$$

因为 DC 可直接从所连接的 $k-x$ 个系统节点中得到 $\alpha(k-x)$个数据,所以将系统节点 $\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_{k-x}$ 的生成矩阵从矩阵 $\boldsymbol{R}_1$ 中消去,并进行列变换得到 $\boldsymbol{R}_2$.将矩阵 $\boldsymbol{G}'^{(\delta_i)}$的第 $\varphi_j(j=1,\cdots,x)$列作为 $\boldsymbol{R}_2$ 的第 $x(j-1)+i$ 列.将矩阵 $\boldsymbol{G}'^{(\delta_i)}$的第 $\varepsilon_j(j=1,\cdots,x)$列作为 $\boldsymbol{R}_2$ 的第 $x^2+x(j-1)+i$ 列.则矩阵 $\boldsymbol{R}_2$ 可以表示为由 $\alpha\times x$ 阶子矩阵($\boldsymbol{T}$、$\boldsymbol{P}_{u,v}$和 $\boldsymbol{Q}_{u,v}$)组成的 $x\times\alpha$ 阶分块矩阵:

$$\begin{aligned}\boldsymbol{R}_2&=[\boldsymbol{G}'^{(\delta_1)}\ \boldsymbol{G}'^{(\delta_2)}\ \cdots\ \boldsymbol{G}'^{(\delta_x)}]\\&=\begin{bmatrix}\boldsymbol{G}_{\varphi_1}^{(\delta_1)}&\boldsymbol{G}_{\varphi_1}^{(\delta_2)}&\cdots&\boldsymbol{G}_{\varphi_1}^{(\delta_x)}\\\vdots&\vdots&\ddots&\vdots\\\boldsymbol{G}_{\varphi_x}^{(\delta_1)}&\boldsymbol{G}_{\varphi_x}^{(\delta_2)}&\cdots&\boldsymbol{G}_{\varphi_x}^{(\delta_x)}\end{bmatrix}\\&=\begin{bmatrix}\boldsymbol{T}&\boldsymbol{P}_{\varphi_2,1}&\cdots&\boldsymbol{P}_{\varphi_x,1}&\boldsymbol{Q}_{\varepsilon_1,1}&\cdots&\boldsymbol{Q}_{\varepsilon_{k-x},1}\\\boldsymbol{Q}_{\varphi_1,2}&\boldsymbol{T}&\cdots&\boldsymbol{P}_{\varphi_x,2}&\boldsymbol{Q}_{\varepsilon_1,2}&\cdots&\boldsymbol{Q}_{\varepsilon_{k-x},2}\\\vdots&\vdots&\ddots&\vdots&\vdots&\ddots&\vdots\\\boldsymbol{Q}_{\varphi_1,x}&\boldsymbol{Q}_{\varphi_2,x}&\cdots&\boldsymbol{T}&\boldsymbol{Q}_{\varepsilon_1,x}&\cdots&\boldsymbol{Q}_{\varepsilon_{k-x},x}\end{bmatrix}\end{aligned} \tag{10}$$

其中,矩阵 $\boldsymbol{T}$ 的元素 $t_{i,j}=c_i^{(\delta_j)},\forall i,j$;矩阵 $\boldsymbol{P}_{u,v}$的第 u 行为$[c_{\varphi_v+\alpha}^{(\delta_1)}\ \cdots\ c_{\varphi_v+\alpha}^{(\delta_x)}]$,其余行为 0;矩阵 $\boldsymbol{Q}_{u,v}$的第 u 行为$[c_{\varphi_v+\alpha-1}^{(\delta_1)}\ \cdots\ c_{\varphi_v+\alpha-1}^{(\delta_x)}]$,其余行为 0.

设 $x\times x$ 阶矩阵 $\boldsymbol{F}$,其元素 $f_{i,j}=c_{\varphi_i+\alpha}^{(\delta_j)},\forall i,j$.将矩阵 $\boldsymbol{R}_2$ 的后 $k-x$ 列块矩阵右乘矩阵 $\boldsymbol{F}^{-1}$:

$$\boldsymbol{Q}_{u,v}\boldsymbol{F}^{-1}=\boldsymbol{E}_{u,v} \tag{11}$$

其中,矩阵 $\boldsymbol{E}_{u,v}$的第(u,v)个项为 1,其余项为 0.通过式(11)的计算,由矩阵 $\boldsymbol{R}_2$ 得到矩阵 $\boldsymbol{R}_3$:

$$\boldsymbol{R}_3=\begin{bmatrix}\boldsymbol{T}&\boldsymbol{P}_{\varphi_2,1}&\cdots&\boldsymbol{P}_{\varphi_x,1}&\boldsymbol{E}_{\varepsilon_1,1}&\cdots&\boldsymbol{E}_{\varepsilon_{k-x},1}\\\boldsymbol{Q}_{\varphi_1,2}&\boldsymbol{T}&\cdots&\boldsymbol{P}_{\varphi_x,2}&\boldsymbol{E}_{\varepsilon_1,2}&\cdots&\boldsymbol{E}_{\varepsilon_{k-x},2}\\\vdots&\vdots&\ddots&\vdots&\vdots&\ddots&\vdots\\\boldsymbol{Q}_{\varphi_1,x}&\boldsymbol{Q}_{\varphi_2,x}&\cdots&\boldsymbol{T}&\boldsymbol{E}_{\varepsilon_1,x}&\cdots&\boldsymbol{E}_{\varepsilon_{k-x},x}\end{bmatrix} \tag{12}$$

矩阵 $\boldsymbol{R}_3$ 后 $k-x$ 列块矩阵对应的每个列向量都有且只有一个非零元素.DC 可从这些列向量解出对应的数据,并消除这些数据对其余编码数据的影响,得到 $x^2\times x^2$ 阶矩阵 $\boldsymbol{R}_4$:

$$\boldsymbol{R}_4=\begin{bmatrix}\boldsymbol{T}'&\boldsymbol{P}'_{\varphi_2,1}&\cdots&\boldsymbol{P}'_{\varphi_x,1}\\\boldsymbol{Q}'_{\varphi_1,2}&\boldsymbol{T}'&\cdots&\boldsymbol{P}'_{\varphi_x,2}\\\vdots&\vdots&\ddots&\vdots\\\boldsymbol{Q}'_{\varphi_1,x}&\boldsymbol{Q}'_{\varphi_2,x}&\cdots&\boldsymbol{T}'\end{bmatrix} \tag{13}$$

其中,矩阵 $\boldsymbol{T}'$、$\boldsymbol{P}'_{u,v}$和 $\boldsymbol{Q}'_{u,v}$为 $x\times x$ 阶方阵;矩阵 $\boldsymbol{T}'$的元素 $t_{i,j}=c_{\varphi_i}^{(\delta_j)},\forall i,j$;矩阵 $\boldsymbol{P}'_{u,v}$ 的第 u 行为$[c_{\varphi_v+\alpha}^{(\delta_1)}\ \cdots\ c_{\varphi_v+\alpha}^{(\delta_x)}]$,其余行为 0;矩阵 $\boldsymbol{Q}'_{u,v}$的第 u 行为$[c_{\varphi_v+\alpha-1}^{(\delta_1)}\ \cdots\ c_{\varphi_v+\alpha-1}^{(\delta_x)}]$,其余行为 0.此时矩阵 $\boldsymbol{R}_4$ 相当于(n,x,d)-MMSR 中 DC 利用 x 个冗余节点重建用户文件时所计算的矩阵.

将矩阵 $\boldsymbol{R}_4$ 右乘矩阵$\begin{bmatrix}\boldsymbol{T}'^{-1}&&\boldsymbol{0}\\&\ddots&\\\boldsymbol{0}&&\boldsymbol{T}'^{-1}\end{bmatrix}$,可得:

$$\boldsymbol{R}_5=\begin{bmatrix}\boldsymbol{I}&\boldsymbol{P}''_{\varphi_2,1}&\cdots&\boldsymbol{P}''_{\varphi_x,1}\\\boldsymbol{Q}''_{\varphi_1,2}&\boldsymbol{I}&\cdots&\boldsymbol{P}''_{\varphi_x,2}\\\vdots&\vdots&\ddots&\vdots\\\boldsymbol{Q}''_{\varphi_1,x}&\boldsymbol{Q}''_{\varphi_2,x}&\cdots&\boldsymbol{I}\end{bmatrix} \tag{14}$$

其中,矩阵 $\boldsymbol{I}$、$\boldsymbol{P}''_{u,v}$、$\boldsymbol{Q}''_{u,v}$为 $x\times x$ 阶方阵;矩阵 $\boldsymbol{I}$ 为单位矩阵;矩阵 $\boldsymbol{P}''_{u,v}$的第 u 行为$[c_{\varphi_v+\alpha}^{(\delta_1)}\ \cdots\ c_{\varphi_v+\alpha}^{(\delta_x)}]\boldsymbol{T}'^{-1}$,其余行为 0;矩阵 $\boldsymbol{Q}''_{u,v}$的第 u 行为$[c_{\varphi_v+\alpha-1}^{(\delta_1)}\ \cdots\ c_{\varphi_v+\alpha-1}^{(\delta_x)}]\boldsymbol{T}'^{-1}$,其余行为 0.此时,证明矩阵 $\boldsymbol{S}_1$ 的非奇异性等效于证明矩阵 $\boldsymbol{R}_5$ 的非奇异性.

由矩阵的特征值性质可知,若一个矩阵的特征值全不为 0,则该矩阵可逆.下面针对 x 取不同值时,根据矩阵 $\boldsymbol{R}_5$ 的特征值讨论其非奇异性.

当 $x=2$ 时,矩阵 $\boldsymbol{R}_5$ 的特征方程式可表示为$(1-\lambda)^4-ab(1-\lambda)^2=0$.由于特征值 λ 不能为 0,则 $ab\geqslant 0$ 且 $ab\neq 1$.其中 a、b 为矩阵 $\boldsymbol{P}''_{\varphi_2,1}$的非零元素.

当 $x=3$ 时,矩阵 $\boldsymbol{R}_5$ 的特征方程式可表示为$(1-\lambda)^9-abef(1-\lambda)^5=0$.由于特征值 λ 不能为 0,则 $abef\geqslant 0$ 且 $abef\neq 1$.其中 a、b、e、f 为矩阵 $\boldsymbol{P}''_{u,v}$、$\boldsymbol{Q}''_{u,v}$的非零元素.

当 $4\leqslant x\leqslant k$ 时,矩阵 $\boldsymbol{R}_5$ 的特征方程见式(15).其中 ξ_{ij}为矩阵 $\boldsymbol{P}''_{u,v}$、$\boldsymbol{Q}''_{u,v}$的非零元素.式(15)中第 $1,x+2,2x+3,\cdots,(x-1)x+x$ 行所有项除主对角线项外都为 0,则特征方程式可以简化为式(16).

由式(16)可知,当 $4\leqslant x\leqslant k$ 时,矩阵 $\boldsymbol{R}_5$ 在任意条件下满秩.

综上所述,当 DC 连接 $4\leqslant x\leqslant k$ 个冗余节点时,矩阵 $\boldsymbol{S}_1$ 为非奇异矩阵;当 DC 连接 $x=2$ 或 3 个冗余节点时,选取的柯西矩阵满足一定条件时,矩阵 $\boldsymbol{S}_1$ 为非奇异矩阵.特别地,若选取特殊柯西矩阵(如 Hilbert 矩阵),矩阵 $\boldsymbol{S}_1$ 在 $x=2$ 或 3 时满足非奇异性.

定理 2　(MDS 性质)　设 $\boldsymbol{C}$ 为柯西矩阵,以 $\boldsymbol{G}^{(m)}$ 为生成矩阵的 MMSR 码具有 MDS 性质.

证明:定理 1 已证得矩阵 $\boldsymbol{S}_1$ 为非奇异矩阵.由于矩阵 $\boldsymbol{S}_1$ 是由任意 k 个节点(系统节点和冗余节点)的生成矩阵 $\boldsymbol{G}^{(m)}$ 以列向量方式组成的 $M\times M$ 阶矩阵.因此,只要 DC 连接任意 k 个节点就可以重建出大小为 M 的源文件,即 MMSR 满足 MDS 性质.

$$\begin{vmatrix}
\underline{1-\lambda} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0\\
0 & 1-\lambda & \cdots & 0 & \xi_{11} & \xi_{12} & \cdots & \xi_{1x} & \cdots & 0 & 0 & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & \cdots & 1-\lambda & 0 & 0 & \cdots & 0 & \cdots & \xi_{11} & \xi_{12} & \cdots & \xi_{1x}\\
\xi_{11} & \xi_{12} & \cdots & \xi_{1x} & 1-\lambda & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0\\
0 & 0 & \cdots & 0 & 0 & \underline{1-\lambda} & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1-\lambda & \cdots & \xi_{21} & \xi_{22} & \cdots & \xi_{2x}\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
\xi_{(x-1)1} & \xi_{(x-1)2} & \cdots & \xi_{(x-1)x} & 0 & 0 & \cdots & 0 & \cdots & 1-\lambda & 0 & \cdots & 0\\
0 & 0 & \cdots & 0 & \xi_{(x-1)1} & \xi_{(x-1)2} & \cdots & \xi_{(x-1)x} & \cdots & 0 & 1-\lambda & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & \underline{1-\lambda}
\end{vmatrix}=0 \tag{15}$$

$$(1-\lambda)^{x}\begin{vmatrix}
1-\lambda & 0 & \cdots & 0 & \xi_{11} & \xi_{13} & \cdots & \xi_{1x} & \cdots & 0 & 0 & \cdots & 0\\
0 & 1-\lambda & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & \cdots & 1-\lambda & 0 & 0 & \cdots & 0 & \cdots & \xi_{11} & \xi_{12} & \cdots & \xi_{1(x-1)}\\
\xi_{12} & \xi_{13} & \cdots & \xi_{1x} & 1-\lambda & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0\\
0 & 0 & \cdots & 0 & 0 & 1-\lambda & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1-\lambda & \cdots & \xi_{21} & \xi_{22} & \cdots & \xi_{2(x-1)}\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
\xi_{(x-1)2} & \xi_{(x-1)3} & \cdots & \xi_{(x-1)x} & 0 & 0 & \cdots & 0 & \cdots & 1-\lambda & 0 & \cdots & 0\\
0 & 0 & \cdots & 0 & \xi_{(x-1)1} & \xi_{(x-1)3} & \cdots & \xi_{(x-1)x} & \cdots & 0 & 1-\lambda & \cdots & 0\\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & \cdots & 1-\lambda
\end{vmatrix}$$

$$=(1-\lambda)^{x^2}=0 \tag{16}$$

3.3　(7,3,5)-MMSR 码举例

前面给出了 MMSR 码一般性方案.这里将给出一个(7,3,5)-MMSR 例子验证 MMSR 码的可行性与有效性.

根据 MMSR 码参数的约束条件,取 $n=7$、$k=3$、$d=5$,并考虑 $M=9$、$\beta=1$、$\alpha=3$.不妨假设前三个节点为系统节点,后四个节点为冗余节点.设 9 个用户数据为 $a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9$:

系统节点 m 的生成矩阵的第 i 部分为:

$$\boldsymbol{G}_i^{(m)}=\begin{cases}\boldsymbol{I}_3, i=m\\ \boldsymbol{0}_3, i\neq m\end{cases}\quad \forall\, i\in\{1,2,3\} \tag{17}$$

考虑如下 5 阶 Hilbert 矩阵 $\boldsymbol{C}_5$:

$$\boldsymbol{C}_5=\begin{bmatrix}
1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5}\\
\frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6}\\
\frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7}\\
\frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8}
\end{bmatrix}^{\mathrm{T}} \tag{18}$$

由此可得到冗余节点 m 的生成矩阵:

$$\boldsymbol{G}^{(m)}=\begin{bmatrix}
c_1^{(m)} & c_2^{(m)} & c_3^{(m)} & c_4^{(m)} & 0 & 0 & c_5^{(m)} & 0 & 0\\
0 & c_4^{(m)} & 0 & c_1^{(m)} & c_2^{(m)} & c_3^{(m)} & 0 & c_5^{(m)} & 0\\
0 & 0 & c_4^{(m)} & 0 & 0 & c_5^{(m)} & c_1^{(m)} & c_2^{(m)} & c_3^{(m)}
\end{bmatrix}^{\mathrm{T}} \tag{19}$$

4 节点修复方案

云存储系统主要依靠数据冗余和节点修复方案保证云端服务器上存储的用户数据的可靠性.由于 MMSR 码限定其帮助节点的个数 $2k-1\leqslant d\leqslant n-1$,所以 MMSR 码至多允许 $n-2k+1$ 个节点失效.

图 3 为前面构造的(7,3,5)-MMSR 码的多节点修复实例图.不妨设节点 1、节点 2 失效,三个系统节点中的元素顺序存放,四个冗余节点的第 $i(1\leqslant i\leqslant 3)$ 个元素包含了修复系统节点 i 的期望数据 A_1、A_2(第 i 部分)和干扰数据 A_3(其余部分).

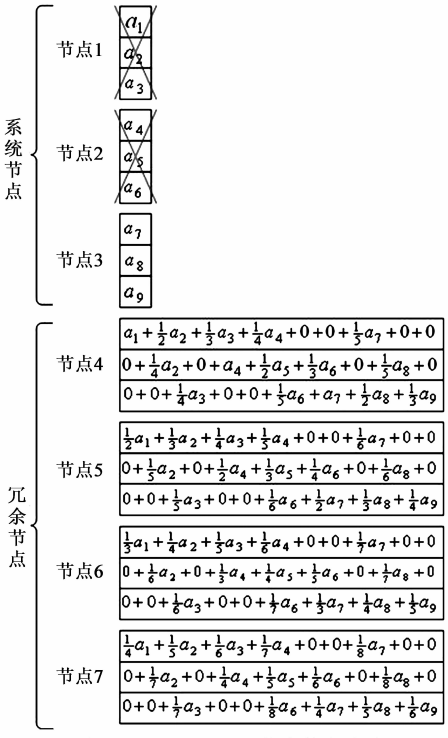


图3 (7,3,5)-MMSR节点修复方案

利用 MMSR 码修复系统节点 l 时,只需从帮助节点输出第 l 个元素给新节点 l'.

4.1 同步修复

图 4 是(7,3,5)-MMSR 同步修复方案的简化图.节点 1 和节点 2 失效,在余下的未失效节点中选择 5 个帮助节点用以修复节点 1 和节点 2.帮助节点集 $H=\{3,4,5,6,7\}$.新节点 1′和新节点 2′分别是节点 1 和节点 2 的再生节点.5 个帮助节点分别输出各自的第 1 个元素给新节点 1′,第 2 个元素给新节点 2′.由图 4 可知同步修复节点 1 和节点 2 的总修复带宽为 $\Gamma=2\gamma=2d\beta=10$.

在修复节点 1 时,5 个帮助节点分别输出各自的第 1 个元素,即生成矩阵的第一列给新节点 1′.式(20)中第一列是新节点 1′从系统帮助节点 3 收到的向量,后四列分别是从冗余帮助节点处收到的向量.且后四列向量的第 1 部分是期望数据(柯西矩阵 $\boldsymbol{C}_5$ 的相应元素);第 2、3 部分是干扰数据,干扰数据均按$[1\quad 0\quad 0]^{\mathrm{T}}$方向对齐.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ c_1^{(4)} & c_2^{(4)} & c_3^{(4)} & c_4^{(4)} & 0 & 0 & c_5^{(4)} & 0 & 0\\ c_1^{(5)} & c_2^{(5)} & c_3^{(5)} & c_4^{(5)} & 0 & 0 & c_5^{(5)} & 0 & 0\\ c_1^{(6)} & c_2^{(6)} & c_3^{(6)} & c_4^{(6)} & 0 & 0 & c_5^{(6)} & 0 & 0\\ c_1^{(7)} & c_2^{(7)} & c_3^{(7)} & c_4^{(7)} & 0 & 0 & c_5^{(7)} & 0 & 0 \end{bmatrix}^{\mathrm{T}} \tag{20}$$

由式(20)可见,节点 1′可以通过冗余帮助节点的第 1 个元素减去节点 3 的第 1 个元素的相应倍数,消去冗余帮助节点 m 中第 3 部分的干扰数据.由于本例中节点 2 失效,故冗余帮助节点的第 2 部分中的干扰数据不能被消除,而需与期望数据一起解出.消除部分干扰后,节点 1′处的参数矩阵为:

$$\boldsymbol{D}_1=\begin{bmatrix} c_1^{(4)} & c_2^{(4)} & c_3^{(4)} & c_4^{(4)} & 0 & \cdots & 0\\ c_1^{(5)} & c_2^{(5)} & c_3^{(5)} & c_4^{(5)} & 0 & \cdots & 0\\ c_1^{(6)} & c_2^{(6)} & c_3^{(6)} & c_4^{(6)} & 0 & \cdots & 0\\ c_1^{(7)} & c_2^{(7)} & c_3^{(7)} & c_4^{(7)} & 0 & \cdots & 0 \end{bmatrix}^{\mathrm{T}} \tag{21}$$

设柯西矩阵 $\boldsymbol{C}_5$ 的 4 阶子矩阵为 $\boldsymbol{C}_4$.矩阵 $\boldsymbol{D}_1$ 右乘上 $\boldsymbol{C}_4^{-1}$ 即可恢复出数据 a_1,a_2,a_3,a_4.其中 a_1,a_2,a_3 为节点 1′所要再生的期望数据.至此,节点 1 被精确修复在节点 1′处.经历相同过程后,节点 2 也被精确修复在节点 2′处.

4.2 异步修复

假设使用异步修复方案顺序修复节点 1 和节点 2.从图 3 可见修复节点 2 时,若节点 1 和节点 3 都参与修复,则会增大干扰对齐性能,降低修复计算量和复杂度.图 5 是(7,3,5)-MMSR 异步修复方案的简化图.修复节点 1 时的帮助节点集 $H_1=\{3,4,5,6,7\}$,节点 1 的修复过程与同步修复方式相同.

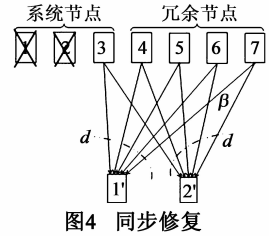


图4 同步修复

待节点 1 修复完成后,云存储系统根据当前网络状况,调整修复节点 2 的帮助节点集.经调整后,节点 2 的帮助节点集可增加一个系统帮助节点(新节点 1′),减少一个冗余帮助节点(节点 7).因此,修复节点 2 的帮助节点集 $H_2=\{1',3,4,5,6\}$.由图 5 可知异步修复节点 1 和节点 2 的总修复带宽 $\Gamma=10$.

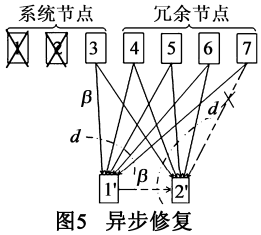


图5　异步修复

在修复节点 2 时,5 个帮助节点分别输出各自的第 2 个元素,即生成矩阵的第二列给新节点 $2'$.式(22)中第一、二列是新节点 $2'$ 从系统帮助节点 $1'$、3 收到的向量,后三列分别是从冗余帮助节点处收到的向量.且后三列向量的第 2 部分是期望数据;第 1、3 部分是干扰数据,干扰数据均按$[0\quad 1\quad 0]^{\mathrm{T}}$方向对齐.此时有系统节点 $1'$ 的加入,在消除干扰数据时,第 1 部分和第 3 部分的干扰数据可同时消除.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & c_4^{(4)} & 0 & c_1^{(4)} & c_2^{(4)} & c_3^{(4)} & 0 & c_5^{(4)} & 0 \\ 0 & c_4^{(5)} & 0 & c_1^{(5)} & c_2^{(5)} & c_3^{(5)} & 0 & c_5^{(5)} & 0 \\ 0 & c_4^{(6)} & 0 & c_1^{(6)} & c_2^{(6)} & c_3^{(6)} & 0 & c_5^{(6)} & 0 \end{bmatrix}^{\mathrm{T}} \tag{22}$$

由式(22)可见,冗余帮助节点 m 的第 2 个元素减去节点 $1'$ 和节点 3 的第 2 个元素相应倍数,即可达到同时消除所有干扰的目地.消除干扰后,节点 $2'$ 处的参数矩阵为:

$$\begin{aligned} \boldsymbol{D}_2 &= \begin{bmatrix} 0 & 0 & 0 & c_1^{(4)} & c_2^{(4)} & c_3^{(4)} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_1^{(5)} & c_2^{(5)} & c_3^{(5)} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_1^{(6)} & c_2^{(6)} & c_3^{(6)} & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \\ &= [\boldsymbol{0}_3^{\mathrm{T}} \quad \boldsymbol{C}_3^{\mathrm{T}} \quad \boldsymbol{0}_3^{\mathrm{T}}]^{\mathrm{T}} \end{aligned} \tag{23}$$

矩阵 $\boldsymbol{D}_2$ 右乘上 3 阶子矩阵 $\boldsymbol{C}_3$ 的逆矩阵 $\boldsymbol{C}_3^{-1}$ 即可恢复出 a_4, a_5, a_6.至此,节点 2 被精确修复在节点 $2'$ 处.

5　数据重建方案

(n,k)-MDS 码的设计原理是将用户数据存储在 k 个节点中,再通过预编码把用户数据扩散到 n 个节点中.前面已证得 MMSR 码具有 MDS 性质.下面针对(7,3,5)-MMSR 码,讨论重建数据时选择节点的 $k+1$ 种可能,以验证 MMSR 码的 MDS 性质.

5.1　三个系统节点

假设 DC 重建数据时连接节点 1、节点 2 和节点 3.由这三个系统节点的生成矩阵组成的矩阵 $\boldsymbol{S}_1 = \boldsymbol{I}_9$.其中,$\boldsymbol{I}_9$ 是 9×9 的单位矩阵.可得到全部未编码用户数据 A_1、A_2、A_3.

5.2　三个冗余节点

假设 DC 连接节点 4、节点 5 和节点 6,则矩阵 $\boldsymbol{S}_1 = [\boldsymbol{G}^{(4)}\ \boldsymbol{G}^{(5)}\ \boldsymbol{G}^{(6)}]$,改变 $\boldsymbol{S}_1$ 各列的顺序(由于矩阵 $\boldsymbol{S}_1$ 的每一列都单独代表一个存储元素,故可任意改变 $\boldsymbol{S}_1$ 各列的排列)得到 $\boldsymbol{S}_2$:

$$\boldsymbol{S}_2 = \begin{bmatrix} c_1^{(4)} & c_1^{(5)} & c_1^{(6)} & 0 & 0 & 0 & 0 & 0 & 0 \\ c_2^{(4)} & c_2^{(5)} & c_2^{(6)} & c_4^{(4)} & c_4^{(5)} & c_4^{(6)} & 0 & 0 & 0 \\ c_3^{(4)} & c_3^{(5)} & c_3^{(6)} & 0 & 0 & 0 & c_4^{(4)} & c_4^{(5)} & c_4^{(6)} \\ c_4^{(4)} & c_4^{(5)} & c_4^{(6)} & c_1^{(4)} & c_1^{(5)} & c_1^{(6)} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_2^{(4)} & c_2^{(5)} & c_2^{(6)} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_3^{(4)} & c_3^{(5)} & c_3^{(6)} & c_5^{(4)} & c_5^{(5)} & c_5^{(6)} \\ c_5^{(4)} & c_5^{(5)} & c_5^{(6)} & 0 & 0 & 0 & c_1^{(4)} & c_1^{(5)} & c_1^{(6)} \\ 0 & 0 & 0 & c_5^{(4)} & c_5^{(5)} & c_5^{(6)} & c_2^{(4)} & c_2^{(5)} & c_2^{(6)} \\ 0 & 0 & 0 & 0 & 0 & 0 & c_3^{(4)} & c_3^{(5)} & c_3^{(6)} \end{bmatrix} \tag{24}$$

由于此时 DC 未连接系统节点,DC 不能直接得到任何用户数据.矩阵 $\boldsymbol{S}_2$ 右乘上矩阵 $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{C}_3^{-1} & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{C}_3^{-1} & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{C}_3^{-1} \end{bmatrix}$ 得到非奇异矩阵 $\boldsymbol{S}_3$.矩阵 $\boldsymbol{S}_3$ 右乘 $\boldsymbol{S}_3^{-1}$ 即可解出全部用户数据.

5.3　两个系统节点和一个冗余节点

假设 DC 连接节点 1、节点 2 和节点 4,则矩阵 $\boldsymbol{S}_1 = [\boldsymbol{G}^{(1)}\ \boldsymbol{G}^{(2)}\ \boldsymbol{G}^{(4)}]$.DC 可以从节点 1 和节点 2 中($\boldsymbol{S}_1$ 的前 6 个列向量)得到 A_1、A_2,并利用 A_1、A_2 消除节点 4 中相应的数据($\boldsymbol{G}_1^{(4)}$、$\boldsymbol{G}_2^{(4)}$).因此只需从 $\boldsymbol{G}_3^{(4)}$ 中解出 A_3 即可.

$$\boldsymbol{S}_2 = [\boldsymbol{G}_3^{(4)}] = \begin{bmatrix} c_5^{(4)} & 0 & c_1^{(4)} \\ 0 & c_5^{(4)} & c_2^{(4)} \\ 0 & 0 & c_3^{(4)} \end{bmatrix} \tag{25}$$

由三角矩阵的性质可知矩阵 $\boldsymbol{S}_2$ 是满秩.因此 $\boldsymbol{S}_2$ 右乘 $\boldsymbol{G}_3^{(4)-1}$ 可得 A_3.

5.4　一个系统节点和两个冗余节点

假设 DC 连接节点 1、节点 4 和节点 5,则 $\boldsymbol{S}_1 = [\boldsymbol{G}^{(1)}\ \boldsymbol{G}^{(4)}\ \boldsymbol{G}^{(5)}]$.DC 可以从节点 1 中得到 A_1,并用 A_1 消除节点 4 和节点 5 中相应的数据.因此只需从节点 4 和节点 5 的第 2、3 部分中解出 A_2、A_3 即可.将矩阵 $\boldsymbol{S}_1$ 简化并调整列顺序变化为矩阵 $\boldsymbol{S}_2$:

$$\boldsymbol{S}_2 = \begin{bmatrix} \boldsymbol{G}_2^{(4)} & \boldsymbol{G}_2^{(5)} \\ \boldsymbol{G}_3^{(4)} & \boldsymbol{G}_3^{(5)} \end{bmatrix} = \begin{bmatrix} c_1^{(4)} & c_1^{(5)} & 0 & 0 & c_4^{(4)} & c_4^{(5)} \\ c_2^{(4)} & c_2^{(5)} & 0 & 0 & 0 & 0 \\ c_3^{(4)} & c_3^{(5)} & c_5^{(4)} & c_5^{(5)} & 0 & 0 \\ 0 & 0 & c_1^{(4)} & c_1^{(5)} & c_5^{(4)} & c_5^{(5)} \\ c_5^{(4)} & c_5^{(5)} & c_2^{(4)} & c_2^{(5)} & 0 & 0 \\ 0 & 0 & c_3^{(4)} & c_3^{(5)} & 0 & 0 \end{bmatrix} \tag{26}$$

由式(26)知,可从矩阵 $\boldsymbol{S}_2$ 的后两个列向量直接解出 a_4、a_7,并消除它们对其余列的影响.将 $\boldsymbol{S}_2$ 简化为 $\boldsymbol{S}_3$:

$$\boldsymbol{S}_3=\begin{bmatrix} c_2^{(4)} & c_2^{(5)} & 0 & 0 \\ c_3^{(4)} & c_3^{(5)} & c_5^{(4)} & c_5^{(5)} \\ c_5^{(4)} & c_5^{(5)} & c_2^{(4)} & c_2^{(5)} \\ 0 & 0 & c_3^{(4)} & c_3^{(5)} \end{bmatrix} \tag{27}$$

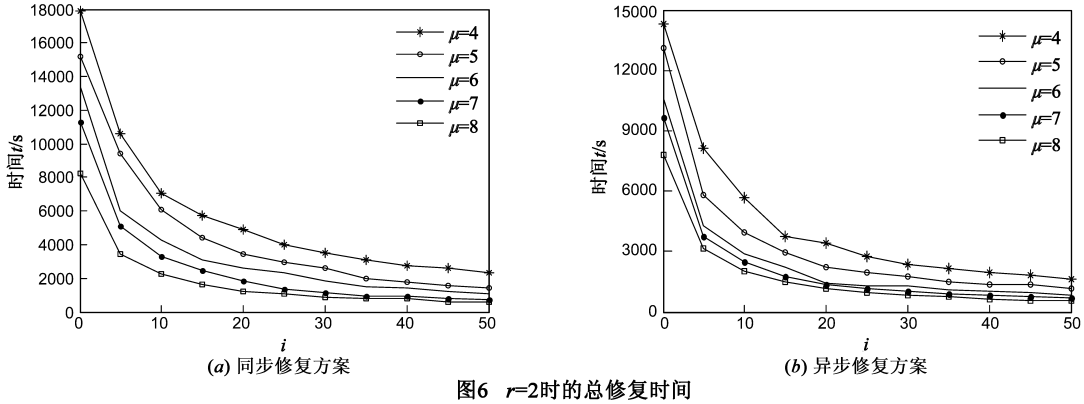
式(27)的处理与式(24)类似,通过线性变化可解出余下的 4 个数据(a_5, a_6, a_8, a_9).

6 数值分析

取用户文件大小 $M=10$GB,系统节点数 $k_i=3+i$,存储节点数 $n_i=2k_i+1$.假设失效节点数 $r=2$,帮助节点数 $d_i=n_i-r$.修复多个节点的总时间为 t.各新生节点从帮助节点处下载数据的速率服从区间(0Mbps, 10Mbps)上的正态分布,且每修复完一个节点,下载速率更新一次.由于解码与更换节点的时间远小于数据传输的时间,这里我们不考虑解码与更换节点所带来的时间消耗.同步和异步修复在不同网络状况(链路速率)、不同存储节点数(i 值不同)下的修复时间如图 6 所示,其中 μ 是正态分布的期望值.

从图 6 可以看出修复时间随着网络状况变好而减少,但变化程度也在逐步降低.并且当存储节点越多时,新节点从每个帮助节点中下载的数据量变少,则修复时间减少.但随着节点数的增多,需要的修复链路数也增多,总的链路状况将变坏.因此当修复节点数达到一定值后,虽然下载的数据量变少,但是修复时间变化不大.

图 7 为同步修复与异步修复的修复时间比较图.从图 7 可以看出,异步修复方案比同步修复方案所用的总修复时间少.但是当网络状况较好时,两种方案没有明显差别.



(a) 同步修复方案 (b) 异步修复方案

图6 r=2时的总修复时间

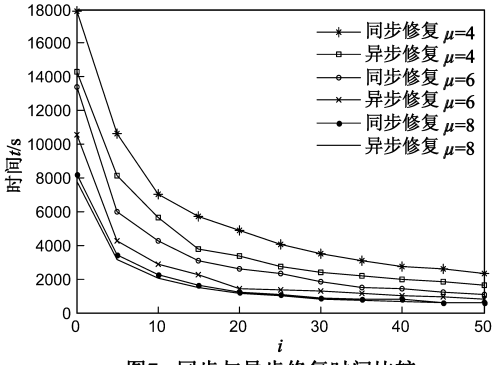


图7 同步与异步修复时间比较

7 总结

本文针对云存储中节点失效的问题,提出了一种多节点同时修复的 MMSR 码,并证明了 MMSR 码的 MDS 性质.MMSR 码不仅能解决多节点同时失效的问题,而且利用干扰对齐技术把修复带宽和存储容量控制在尽可能小的范围.下一步的工作将对多个冗余节点失效的情况或系统节点和冗余节点同时失效的情况进行详细编码设计.

参考文献

[1] 吴吉义,傅建庆,平玲娣,谢琪.一种对等结构的云存储系统研究[J].电子学报,2011,39(5):1100-1107.
WU Ji-yi, Fu Jian-qing, PING Ling-di, XIE Qi. Study on the P2P cloud storage system[J]. Acta Electronica Sinica, 2011, 39(5): 1100-1107. (in Chinese)

[2] 李建敦,彭俊杰,张武.云存储中一种基于布局的虚拟磁盘节能调度方法[J].电子学报,2012,40(11):2247-2254.
LI Jian-dun, PENG Jun-jie, ZHANG Wu. A layout-based energy-aware approach for virtual disk scheduling in cloud storage [J]. Acta Electronica Sinica, 2012, 40(11): 2247-2254. (in Chinese)

[3] 王丽娜,等.一种适于云存储的数据确定性删除方法[J].电子学报,2012,40(2):266-272.
WANG Li-na, et al. A data assured deletion approach adapted for cloud storage[J]. Acta Electronica Sinica, 2012, 40(2): 266

– 272.(in Chinese)
[4] Papailiopoulos D S, Dimakis A G, Cadambe V R. Repair optimal erasure codes through hadamard designs[J]. IEEE Transactions on Information Theory, 2013, 59(5): 3021 – 3037.
[5] 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.
[6] R W Yeung. Information Theory and Network Coding[M]. New York: Springer, 2008.
[7] Dougherty R, Freiling C, Zeger K. Insufficiency of linear coding in network information flow[J]. IEEE Transactions on Information Theory, 2005, 51(8): 2745 – 2759.
[8] Wu Y, Dimakis A G. Reducing repair traffic for erasure coding-based storage via interference alignment[A]. IEEE International Symposium on Information Theory (ISIT)[C]. Seoul: IEEE, 2009. 2276 – 2280.
[9] A Ramakrishnan, A Das, H Maleki, A Markopoulu, S A Jafar. Network coding for three unicast sessions: interference alignment approaches[A]. The 48th Annual Allerton Conference on Communication, Control and Computing (Allerton)[C]. Allerton, IL: IEEE, 2010. 1054 – 1061.
[10] C Suh, d K Ramchandran. Exact-repair MDS code construction using interference alignment[J]. IEEE Transactions on Information Theory, 2011, 57(3): 1425 – 1442.
[11] Shah N B, Rashmi K V, Kumar P V, et al. Interference alignment in regenerating codes for distributed storage: Necessity and code constructions[J]. IEEE Transactions on Information Theory, 2012, 58(4): 2134 – 2158.
[12] Shum K W. Cooperative regenerating codes for distributed storage systems[A]. IEEE International Conference on (ICC)[C]. USA: IEEE, 2011. 1-5.
[13] Shum K W, Hu Y. Exact minimum repair bandwidth cooperative regenerating codes for distributed storage systems[A]. IEEE International Symposium on Information Theory Proceedings (ISIT)[C]. USA: IEEE, 2011. 1442 – 1446.
[14] Kermarrec A M, Le Scouarnec N, Straub G. Repairing multiple failures with coordinated and adaptive regenerating codes[A]. IEEE International Symposium on Network Coding (NetCod)[C]. USA: IEEE, 2011. 1 – 6.
[15] Kenneth W Shum. Cooperative regenerating codes for distributed storage systems[A]. IEEE International Conference on Communications (ICC)[C]. Tokyo, Japan: IEEE, 2011. 1 – 5.
[16] Le Scouarnec N. Exact scalar minimum storage coordinated regenerating codes[A]. Proceedings of IEEE International Symposium on Information Theory (ISIT)[C]. USA: IEEE, 2012. 1197 – 1201.
[17] Wu Y, Dimakis A G, Ramchandran K. Deterministic regenerating codes for distributed storage[A]. The 45th Annual Allerton Conference on Control, Computing, and Communication (Allerton)[C]. Monticello IL: IEEE, 2007. 1 – 8.
[18] Bloemer J, Kalfane M, Karp R, et al. An XOR-Based Erasure-Resilient Coding Scheme[R]. CA USA: ICSI Technical, (Report No. TR-95-048), 1995.

作者简介

谢显中　男. 1966 年出生, 四川通江人. 博士、教授. 2000 年毕业于西安电子科技大学, 获博士学位. 现任重庆邮电大学产学研合作办公室主任, 重庆邮电大学个人通信研究所所长. 主要从事无线和移动通信技术方面的研究与开发.
E-mail: xiexzh@cqupt.edu.cn

黄　倩　女. 1988 年 8 月出生, 重庆北碚人. 2007 年在重庆邮电大学获工学学士学位. 现为在读硕士研究生. 主要从事云存储、个人通信等方面的研究工作.
E-mail: huangq_2011@163.com