

DOI: 10.7652/xjtuxb201705008

稀疏随机纠删码：一种大规模数据存储容灾方法

滕鹏国^{1,2}，陈亮^{1,2}，袁德砦^{1,2}，王晓京¹

(1. 中国科学院成都计算机应用研究所，610041，成都；2. 中国科学院大学，100049，北京)

摘要：针对海量数据存储容灾系统中对扩展性、可靠性及高效性方面的需求，提出了一种高容灾可扩展且能够高概率译码恢复的高效大数据存储容灾算法。该算法利用等行重稀疏随机矩阵高概率行满秩的性质，用来实现数据高效可靠的存储容灾。首先，根据存储系统规模及容灾需求设置相应的编码参数；然后，采用等行重稀疏随机矩阵构造校验矩阵，并且产生相应的生成矩阵；最后，将数据文件分块编码到 n 个存储节点上，实现不同规模、不同容灾需求下的数据容灾存储，并通过设置合理的随机冗余，从而实现对译码成功率的控制。实验和理论分析表明：算法所提存储容灾技术可实现容灾能力不受素数或有限域大小的限制，而是根据存储规模及容灾需求灵活扩展；基于合理的随机冗余，译码成功率趋于 1，实现了高可靠的数据容灾存储；在较大规模存储系统中，算法编译码速率是相应经典 RS 和 CRS 编码方案的 2 倍以上，并在较大码长下具有近似最大距离可分(MDS)的性质，可达到近似最优的存储空间利用率。

关键词：存储系统；容灾技术；纠删码；稀疏随机矩阵

中图分类号：TP333 **文献标志码：**A **文章编号：**0253-987X(2017)05-0048-06

Sparse Random Erasure Code: a Fault Tolerance Scheme for Large-Scale Storage System

TENG Pengguo^{1,2}，CHEN Liang^{1,2}，YUAN Dezhai^{1,2}，WANG Xiaojing¹

(1. Chengdu Computer Applications Institute, Chinese Academy of Sciences, Chengdu 610041, China;

2. Chinese Academy of Sciences University, Beijing 100049, China)

Abstract: To meet the requirements for the scalability, reliability and high efficiency of data storage in large-scale storage system, a new kind of efficient storage fault-tolerance algorithm with flexible scalability and high probability of decoding is proposed, named sparse random erasure code(SREC). The property of sparse random matrix with equal row weight is used in the algorithm. Firstly, the coding parameters are designed according to the scale of storage system and the requirement of fault tolerance; then, the parity check matrix and the corresponding generator matrix are constructed by using sparse random matrix; finally, the files are segmented into blocks and the parity blocks are created by encoding procedure, thus the data storage with different fault-tolerance requirements is realized. Meanwhile, by setting proper random redundancy, the probability of successful decoding is made controllable. Simulation results and theoretical analysis show that the proposed algorithm makes the fault-tolerance capability not have to be restricted by the prime or the size of finite field, and that based on the proper random redundancy, the successful decoding probability trends to 1, thus achieving the high reliability of

收稿日期：2016-11-06。 作者简介：滕鹏国(1986—)，男，博士生；王晓京(通信作者)，男，研究员。 基金项目：国家自然科学基金资助项目(61501064)；四川省科技厅支撑计划资助项目(2015GZ0088)。

data storage. Furthermore, the speeds of encoding and decoding are twice or more than those of classic RS and CRS algorithms, and when the code length is long enough, this algorithm has approximate optimal storage efficiency.

Keywords: storage system; fault-tolerance technology; erasure code; sparse random matrix

随着信息化以及云计算的兴起,海量信息被数字化,使得数据存储量呈现爆炸式的增长,存储系统的规模也日益扩大。为了提供安全可靠高效的存储服务,高性能的存储容灾技术成为当前存储领域中的研究热点,其衡量标准主要体现在容灾扩展性能,计算效率以及存储空间利用率等方面。

当前,存储容灾技术主要分为多副本备份技术和纠删码技术。多副本备份技术由于其不超过50%的存储空间利用率造成了极大的空间浪费,已经很不适用于大规模存储容灾系统,纠删码存储容灾技术成为当前研究的主流。基于廉价磁盘冗余(RAID)阵列的 EVENODD 码、X 码、RDP 码、GRID 码等阵列码技术^[1-4],具有较高的计算效率和存储空间利用率,非常适用于小规模存储容灾系统,但是编码参数受素数的限制,存储结构和容灾能力不易扩展,这阻碍了其在大规模存储容灾系统中的应用。基于有限域运算的 CRS 码、LRC 码、SD 码、Stair 码等^[5-9]有限域编码技术的容灾能力在有限域范围内得到了一定的扩展,并具有较高的存储空间利用率,但是大规模有限域上高昂的计算复杂度阻碍了有限域编码技术的推广。一些通信方面的 LD-PC 码、Tornado 码、LT 码等^[10-11]编码技术被引入到了存储容灾系统中,它们具有灵活的容灾扩展和高效的计算性能,但是高概率成功译码需求带来较高的存储冗余,导致较低存储空间利用率,阻碍了其在存储容灾系统中的应用。

针对大数据时代高效、易扩展存储系统的需求,本文提出了稀疏随机纠删码(SREC),这是一种概率性纠删码存储容灾技术,其编码参数不受素数或有限域大小限制可灵活设置,具有较高的容灾扩展性能;采用等行重稀疏随机矩阵作为校验矩阵,具有较高的计算效率;在较大码长下能够达到近似的 MDS 性质,从而具有较高的存储空间利用率,非常适用于当前大规模存储容灾系统。

1 系统模型

当前的大规模存储系统可抽象为由无数个存储节点构成的整体,每个节点均相互独立,而待存文件则可抽象为一个数据块,存储系统模型如图1所示。

该系统由 n 个存储节点构成,每个存储节点可以是一个磁盘或 RAID 阵列,数据文件 A 首先被分为 k 个大小相同的数据块,随后通过编码产生 m 个冗余数据块,最后将这 $k+m$ 个数据块存放在 n 个存储节点上,从而实现任意 t 个存储节点的失效,可利用剩余的 $n-t$ 个节点高概率恢复出原来 t 个节点的数据。

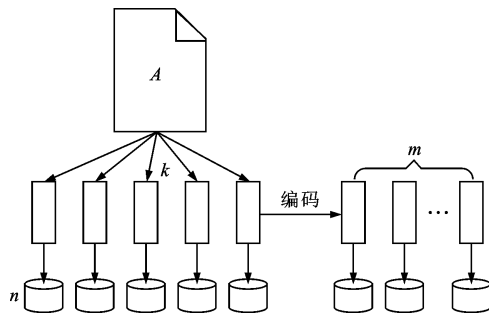


图1 存储系统模型

稀疏随机纠删码的编码参数不再受到素数或有限域大小的限制,因此其容灾能力也可以自由设置,从而达到更高的容灾性能,而存储系统中随机冗余的节点数 δ 是保证稀疏随机纠删码能够高概率译码的一个关键因素。此外,由于采用特殊稀疏随机矩阵作为校验矩阵,稀疏随机纠删码在译码效率上有了较大的提升,这为频繁发生的译码恢复提供了效率保障。

本文所提稀疏随机纠删码利用特殊稀疏随机矩阵的性质,其定义和相关性质如下。

定义 有限域 $GF(2)$ 上的矩阵 $\mathbf{R}_{t \times m}$, $m > t > 0$, 若其中的各个元素取值相互独立,且满足分布规律

$$Q\{e_{ij} = q\} = \begin{cases} p, & q = 1 \\ 1 - p, & q = 0 \end{cases}, p \in (0, 1)$$

则称其为随机矩阵。特别地,当 $p \in ((\lfloor \ln m \rfloor + x)/m, 0.5)$ 时,称其为稀疏随机矩阵。

稀疏随机纠删码采用特殊的稀疏随机矩阵作为校验矩阵,其特殊性在于行重固定且为奇数。 $\lfloor \ln m \rfloor$ 为偶数时,行重为 $\lfloor \ln m \rfloor + 1$, $\lfloor \ln m \rfloor$ 为奇数时,行重为 $\lfloor \ln m \rfloor + 2$,从而矩阵中每个元素为 1 的概率为 $p = d/m$,具有如下性质。

引理1 稀疏随机矩阵 $\mathbf{R}_{t \times m}$ ($m > t$) 行满秩的概率为

$P(t, m) = \prod_{i=1}^t (1 - p^{\delta+i}) > 1 - \frac{1}{1-p} p^{\delta+1}$, 其中 $p \in ((lbm+x)/m, 0.5)$, $\delta = m - t$ 。

证明: 记稀疏随机矩阵 $\mathbf{R}_{t \times m}$ ($m > t$) 行满秩的概率为 $P(t, m)$, 并记 $\delta = m - t$, 逐行增加矩阵 $\mathbf{R}_{t \times m}$ 的秩, 可得

$P(t, m) = \prod_{i=1}^t (1 - (1-p)^{\delta+i})$ 。令 $\epsilon = 1 - p$, 从而 $0 < \epsilon < 1$, 当 $i \rightarrow \infty$ 时, $\epsilon^i \rightarrow 0$, 对其取对数得 $\ln P = \ln(1 - \epsilon^{\delta+1}) + \ln(1 - \epsilon^{\delta+2}) + \dots + \ln(1 - \epsilon^m)$, 根据 $\ln(1-x)$ 在 $x=0$ 处的泰勒展开, 展开各项得

$$\ln P = -\epsilon^{\delta+1} \frac{1-\epsilon^k}{1-\epsilon} - \frac{1}{2} \epsilon^{2(\delta+1)} \frac{1-\epsilon^{2k}}{1-\epsilon^2} - \dots \quad (1)$$

由 $0 < \epsilon < 1$, 式(1)可化简为

$$\ln P > -\frac{\epsilon^{\delta+1}}{1-\epsilon} - \frac{\epsilon^{2(\delta+1)}}{1-\epsilon} - \frac{\epsilon^{3(\delta+1)}}{1-\epsilon} - \dots > -\frac{1}{1-\epsilon} \frac{\epsilon^{\delta+1}}{1-\epsilon^{\delta+1}}$$

当 δ 较大时, 可得

$$P > e^{-\frac{1}{1-\epsilon} \frac{\epsilon^{\delta+1}}{1-\epsilon^{\delta+1}}} \approx 1 - \frac{1}{p} (1-p)^{\delta+1}。$$

对于稀疏随机矩阵 $\mathbf{R}_{t \times m}$, 当 $p = (lbm+x)/m$ 时, $P \sim \frac{\lambda^c}{c!} e^{-\lambda}$, 其中 $\lambda = ae^{-x}$, $a = t/m^{[12]}$ 。由此可知, 稀疏随机矩阵行满秩概率符合泊松分布, 且其概率与矩阵规模、稀疏程度有关。根据引理 1, 可将矩阵规模对秩的影响细化为 $\delta = m - t$, 从而明确了矩阵规模对秩的影响, 同时也为稀疏随机纠删码的近似 MDS 性质提供了理论保障。

引理 2 等行重稀疏随机矩阵 $\mathbf{R}_{t \times m}$ ($m > t$) 具有较高的行满秩概率 $P \approx 1 - \frac{1}{p} (1-p)^{\delta+1}$ 。

2 稀疏随机纠删码

本节将详细介绍稀疏随机纠删码 (SREC) 的相关构造及编译码算法。

2.1 SREC 码基本构造思想

对于线性分组码, 通常利用生成矩阵 $\mathbf{G}$ 和信息矢量 α 产生码字 β , 可表示为 $\mathbf{G} \times \alpha = \beta$, 同时利用校验矩阵 $\mathbf{H}$ 实现失效数据的恢复, 可表示为 $\beta^T \times \mathbf{H} = \theta$ 。由此可知, 生成矩阵、校验矩阵是编码技术的核心, 决定了其构造方式和译码性能。作为线性分组码的一种, SREC 码同样可以利用上述编译码方式实现编译码。由于矩阵 $\mathbf{G}$ 、 $\mathbf{H}$ 互为零空间, 则 $\mathbf{H}^T \times \mathbf{G} = \theta$ 或者 $\mathbf{G}^T \times \mathbf{H} = \theta$ 。鉴于实际存储系统中译码恢复过程频发, 为了提升译码速率, 本文采用等行重稀疏随机矩阵构造校验矩阵, 同时为便于数据的存

取和恢复, 将稀疏随机纠删码设计为系统码结构, 也即生成矩阵 $\mathbf{G}_{n \times k}$ 为上单位阵。稀疏随机纠删码的校验矩阵和生成矩阵的构建如下。

算法 1

输入: 码长 n , 信息位长度 k , 行重 d

输出: 矩阵 $\mathbf{H}$ 、 $\mathbf{G}$

1. 计算 $m = n - k$;
2. 构造二进制稀疏随机矩阵 $\mathbf{H}_1 = \mathbf{R}_{k \times m}$, 且使之行重为 d ;
3. 构造二进制满秩稀疏随机矩阵 $\mathbf{H}_2 = \mathbf{R}_{m \times m}$, 且使之行重为 d ;
4. 设置校验矩阵 $\mathbf{H}_{n \times m} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix}$;
5. 由 $\mathbf{H}^T \times \mathbf{G} = \mathbf{0}$, 设置生成矩阵 $\mathbf{G}_{n \times k} = \begin{bmatrix} \mathbf{I}_{k \times k} \\ [\mathbf{H}_2^T]^{-1} \mathbf{H}_1^T \end{bmatrix}$;
6. 返回校验矩阵 $\mathbf{H}$ 和生成矩阵 $\mathbf{G}$ 。

2.2 编码算法

稀疏随机纠删码的编码参数可以根据实际存储系统的规模及容灾需求灵活设置, 因此可以实现各种不同存储环境中不同容灾能力下的数据容灾存储。稀疏随机纠删码的编码算法如下。

算法 2

输入: 存储系统中存储节点个数 n , 容灾能力 f , 随机冗余 δ

输出: 编码冗余数据

1. 计算容灾节点数 $t = nf$, 数据节点数 $k = n - t - \delta$;
2. 冗余节点数 $m = n - k$, 校验矩阵行重 $d = \lfloor lbm \rfloor$, 若 d 为偶数, 则 $d = d + 1$, 否则 $d = d + 2$;
3. 根据编码参数, 利用算法 1 构造生成矩阵 $\mathbf{G}$;
4. 将原始文件数据平均分为 k 个大小相同的块, 不足的位补 0;
5. 在 n 个存储节点中依次选取 k 个节点存放原始数据分块;
6. 由 $\mathbf{G}_{n \times k} \times \alpha_{k \times 1} = \beta_{n \times 1}$ 产生编码数据 $\beta_{n \times 1}$;
7. 将编码数据 $\beta_{n \times 1}$ 中最后 m 个数据块依次存储在剩余的 m 个冗余存储节点中;
8. 编码结束。

综上所述, 稀疏随机纠删码的编码流程主要分为编码参数的确定、生成矩阵的构造以及冗余数据的产生 3 个部分。编码速率主要体现在编码冗余的产生过程中, 而这主要由生成矩阵中冗余数据对应行组成的子矩阵中 1 的个数决定。

2.3 译码算法

当存储系统中由于存储节点损毁或失效等原因

导致数据丢失时,可以根据损毁程度进行失效数据的译码恢复。定义存储系统的失效率 s 为存储系统中失效存储节点数占整个存储系统中存储节点数的比例。稀疏随机纠错码利用校验矩阵进行译码恢复,译码流程可以按照如下算法进行。

算法 3

输入:校验矩阵 $\mathbf{H}$,容灾能力 f ,剩余存储节点中的数据

输出:失效存储节点的数据

1. 统计存储容灾系统中失效存储节点的数目;
2. 计算该存储容灾系统的失效率 s ;
3. if $s > f$ 则失效节点数目过多,超过容灾能力 f ,无法恢复失效数据,重构失败,返回无法译码;
4. end if;
5. 设置失效存储节点为未知;
6. 由 $\mathbf{H}_{n \times (n-k)}^T \times \boldsymbol{\beta}_{n \times 1} = \mathbf{0}$, 构建译码方程组;
7. 如果方程组无解,则返回无法译码;
8. 方程组的解即为失效存储节点的数据,并将其依次放入失效存储节点中;
9. 恢复成功,译码结束。

综上可知,稀疏随机纠错码的译码过程主要在于译码方程组的求解,而译码过程主要是求解失效节点对应校验矩阵行构成的子矩阵所构造方程组解的过程,失效数据的值即为方程组的解。为了简化计算,可采用广义矩阵的逆矩阵进行译码恢复。若要译码成功,则表示方程组有唯一解,方程组有唯一解的充要条件是系数矩阵 $\mathbf{A}_{t \times m}$ 行满秩。

由引理 1、2 可知,等行重稀疏随机矩阵行满秩的概率 $P \approx 1 - \frac{1}{p}(1-p)^{\delta+1}$ 。稀疏随机纠错码的译码成功率取决于译码子矩阵 $\mathbf{A}_{t \times m}$ 行满秩的概率,当设置合适的 p, δ 后,稀疏随机纠错码可具有较高的译码成功率。

3 性能分析及相关实验结果

本节主要从扩展性、可靠性、高效性 3 个方面讨论稀疏随机纠错码的性能及相关实验结果。

3.1 扩展性对比

对于传统阵列码方案,其数据和冗余的放置都是固定的,编码参数受到了素数的限制,不能自由扩展,从而其容灾能力也受到了限制。对于基于有限域的编码方案,其编码参数和容灾能力虽在有限域范围内可以实现自由设置,具备一定的灵活性和扩展性,但是大规模有限域上高昂的计算复杂度,导致

其运算效率在较大规模存储系统中比较低,严重阻碍了其在存储系统中的应用。而对于概率性译码方案,其编码参数和容灾能力均可实现自由扩展,但是概率性译码的特点导致其需要非常多的编码冗余。稀疏随机纠错码作为一类概率性译码方案,其编码参数不受素数或有限域大小的限制,可自由设置,从而其容灾能力也可根据容灾需求灵活设置,因此具有较高的容灾扩展性。

3.2 可靠性分析

存储容灾系统的可靠性主要体现在容灾能力上和数据的容灾恢复上,即存储系统的译码成功率上,根据稀疏随机纠错码的译码算法,可知译码成功率取决于等行重稀疏随机子矩阵行满秩的概率,因此可通过实验分析等行重稀疏随机矩阵行满秩的概率来增强对稀疏随机纠错码可靠性的认识。

首先讨论 $t \times m$ 等行重稀疏随机矩阵 $\mathbf{R}_{t \times m}$ 行重对行满秩概率的影响。选取 $m = \{32, 64, 128, 256, 512\}$, $t = m - \delta$, 随机冗余 $\delta = 20$, 行重 $d = 1 \sim 15$ (间隔 1), 测试对应稀疏随机矩阵 $\mathbf{R}_{t \times m}$ 行满秩的情况, 结果如图 2 所示。由图 2 可知, 随着行重的增加, 等行重稀疏随机矩阵行满秩的概率不断增加, 当 $d \geq \lg m$ 时, 相应等行重稀疏随机矩阵行满秩的概率几乎等于 1, 同时稀疏随机纠错码的译码成功率也趋于 1, 从而保证了存储容灾系统译码恢复的高可靠性。因此, 为了即保持较高的译码成功率, 又保持较高的译码速率, 即较低的稀疏度, 一般选取略大于 $\lg m$ 的行重。

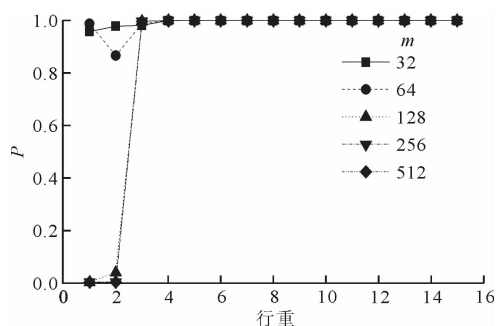
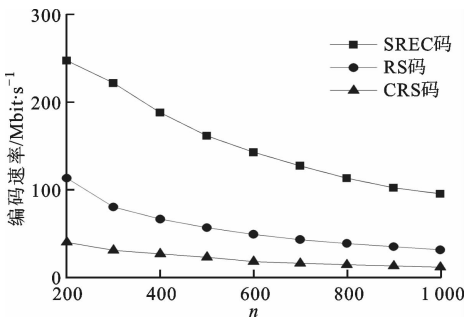


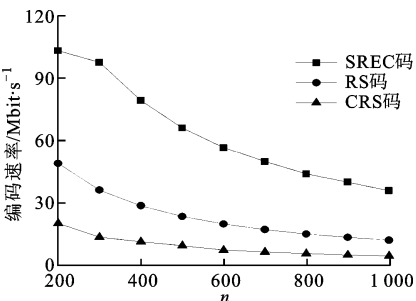
图 2 行重对矩阵行满秩的影响

此外,行重 d 的奇偶性对等行重稀疏随机方阵 $\mathbf{R}_{t \times t}$ 的满秩概率有着重要的影响。选取 $t = \{32, 64, 128, 256, 512\}$, 行重 $d = 1 \sim 15$ (间隔 1) 构造等行重稀疏随机方阵 $\mathbf{R}_{t \times t}$, 编译码速率比较如图 3 所示。由图 3 可知, 稀疏随机方阵 $\mathbf{R}_{t \times t}$ 在行重 d 为奇数且 $d \geq \lg m$ 时, $\mathbf{R}_{t \times t}$ 满秩的概率较高, 约为 0.289, 而当行重 d 为偶数时, 稀疏随机方阵 $\mathbf{R}_{t \times t}$ 满秩的概率则

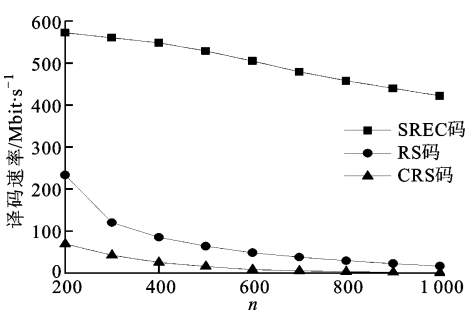
几乎为 0,因此设置稀疏随机纠删码校验矩阵的行重为奇数。



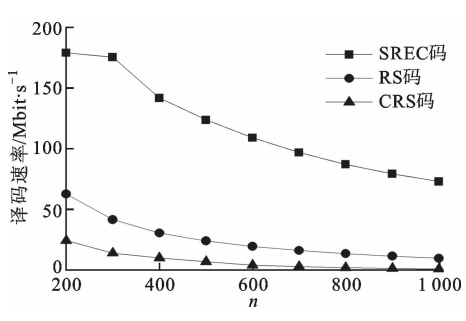
(a)容灾 10%情况下编码速率比较



(b)容灾 30%情况下编码速率比较



(c)容灾 10%情况下译码速率比较



(d)容灾 30%情况下译码速率比较

图 3 编译码速率比较

3.3 高效性分析

稀疏随机纠删码的高效性主要体现在编译码性

能、存储空间利用率、恢复带宽和更新效率上。本节实验测试所用计算机 CPU 为 Intel(R) Core(TM) i3,内存为 4.0 GB,64 位 Ubuntu 15.10 操作系统。

选取存储系统的容灾能力 $f = \{10\%, 30\%\}$, $\delta = 20$,构造码长 $n = 200 \sim 1\,000$ (间隔 100)下相应的编译码方案,并分别与经典 RS 码、CRS 码方案进行编译码效率的实验,比较结果如图 3 所示。由图 3 可知,稀疏随机纠删码的编码速率是相应 RS、CRS 方案的 2 倍以上,随着规模的不断增加,编码速率均呈现下降趋势,但稀疏随机纠删码仍然保持了编码速率的相对优势。此外,CRS 方案在本质上也使用了 XOR 运算,但是其生成矩阵是在有限域 $GF(2^w)$ 上构造的,从而相同参数下的生成矩阵,CRS 的规模是稀疏随机纠删码的 w^2 倍,因此需要的运算更多,编码效率较低。同样的,稀疏随机纠删码的译码速率是相应的 RS 和 CRS 方案的 2 倍以上,且随着规模的不断增加,译码速率均呈现下降趋势,但稀疏随机纠删码仍然保持了译码速率的相对优势。

存储空间利用率是大规模存储系统中一个很重要的衡量标准,存储空间利用率可利用原始数据量占整个存储区域的比例进行描述,恢复 t 个数据块所需编码块的数量即为额外增加的存储空间量。对于稀疏随机纠删码,高概率恢复 t 个数据块需要 $t + \delta$ 个编码块,当 $\delta = 20$ 时译码成功率很高;对于 RS 或 CRS 码,由于具有 MDS 性质,恢复 t 个数据块只需要 t 个编码块;对于 Tornado、LT 码,高概率恢复 t 个数据块需要 $1.05t \sim 1.1t$ 个编码块,同时都要求较大的 $t^{[13]}$ 。稀疏随机纠删码随着 t 的增长,恢复 t 个数据块所需的编码块数量将越来越接近 t ,也即越来越接近 MDS 性质,因此具有较高的存储空间利用率。

在大规模数据存储容灾系统中,恢复带宽和更新效率也是需要考量的两个因素,恢复带宽是在译码恢复过程中所需传输的编码块数量,而更新效率是单个数据块的修改所关联的冗余数据块数量。

在恢复带宽方面,对于 k 个数据块的译码恢复,通常具有 MDS 性质的 RAID 阵列码方案和基于有限域的方案,一般需要 k 个编码块即可实现失效数据的译码恢复,而对于概率性译码方案则需要平均 $1.05k \sim 1.1k$ 的编码块来实现译码恢复,从而恢复带宽为 $1.05k \sim 1.1k$ 。稀疏随机纠删码尽管也属于概率性译码,但它却只需要 $k + \delta$ 个编码块进行译码恢复,当 k 较大时, δ 所占的比重非常小,可近似看作 k ,从而具有近似 MDS 性质。

在更新效率上,RAID 阵列码方案可达到最优值 2,基于有限域的方案通常为 m 。对于稀疏随机纠删码,其更新效率取决于编码矩阵 G 中的子矩阵 D ,但由于 D 为两个稀疏随机矩阵的乘积,可粗略地认为其更新效率为 $O(m)$ 。

4 结 论

本文依据稀疏随机矩阵高概率行满秩的特性,提出了一种稀疏随机纠删码的编译码方案,实现了大规模存储系统中对高效存储容灾技术设计需求,其优势主要体现在以下方面。

(1)容灾扩展性。稀疏随机纠删码具有编码参数不受素数或有限域大小限制,容灾能力可随规模灵活扩展的特点。

(2)译码可靠性。稀疏随机纠删码在设置合适的随机冗余的情况下,可达到较高的译码成功率。

(3)高效性。稀疏随机纠删码与基于有限域运算的存储容灾技术相比,采用 XOR 运算,避免了有限域上的计算开销,具有较高的编译码速率,而与概率性存储容灾技术相比,在较大码长下又具有近似 MDS 性质,可实现较高的存储空间利用率。

因此,SREC 码非常适用于大规模存储容灾系统,具有良好的研究价值和广阔的应用前景。

参考文献:

[1] BLAUM M, BRADY J, BRUCK J, et al. EVEN-ODD: an efficient scheme for tolerating double disk failures in RAID architectures [J]. IEEE Transactions on Computers, 1995, 44(2): 192-202.

[2] XU L, BRUCK J. X-code: MDS array codes with optimal encoding [J]. IEEE Transactions on Information Theory, 1999, 45(1): 272-276.

[3] CORBETT P, ENGLISH B, GOEL A, et al. Row-diagonal parity for double disk failure correction [EB/OL]. [2016-10-26]. https://www.usenix.org/legacy/publications/library/proceedings/fast04/tech/corbett/corbett_html/.

[4] LI M, SHU J, ZHENG W. GRID codes: Strip-based erasure codes with high fault tolerance for storage systems [J]. ACM Transactions on Storage, 2009, 4(4): NO. 15.

[5] PLANK J S, XU L. Optimizing Cauchy Reed-Solomon codes for fault-tolerant network storage applications [C]//Proceedings of the Network Computing and Applications. Piscataway, NJ, USA: IEEE, 2006: 173-180.

[7] TAMO I, BARG A. A family of optimal locally recoverable codes [J]. IEEE Transactions on Information Theory, 2013, 60(8): 4661-4676.

[8] PLANK J S, BLAUM M. Sector-disk (SD) erasure codes for mixed failure modes in RAID systems [J]. ACM Transactions on Storage, 2014, 10(1): 4-20.

[9] BLAUM M, PLANK J S, SCHWARTZ M, et al. Partial MDS (PMDS) and sector-disk (SD) codes that tolerate the erasure of two random sectors [C]//Proceedings of the 2014 IEEE International Symposium on Information Theory. Piscataway, NJ, USA: IEEE, 2014: 1792-1796.

[10] LI M, LEE P P. STAIR codes: a general family of erasure codes for tolerating device and sector failures in practical storage systems [EB/OL]. [2016-10-26]. https://www.usenix.org/system/files/conference/fast14/fast14-paper_li-mingqiang.pdf.

[11] LUBY M G, MITZENMACHER M, SHOKROLLAHI M A, et al. Practical loss-resilient codes [C]//Proceedings of the 29th Annual ACM Symposium on Theory of Computing. New York, USA: ACM, 1997: 150-159.

[12] KOLCHIN V F. Random graphs [M]. Cambridge, UK: Cambridge University Press, 1999: 16-29.

[13] LUBY M. LT codes [C]//Proceedings of the 43rd Symposium on Foundations of Computer Science. Piscataway, NJ, USA: IEEE, 2002: 271-280.

(编辑 赵炜)