计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220580
60(10): 2418−2439, 2023

# 低跨云数据中心修复流量的纠删码的快速构造方法

包　涵　王意洁
（并行与分布计算全国重点实验室（国防科技大学）　长沙　410073）
（国防科技大学计算机学院　长沙　410073）
（hanb@nudt.edu.cn）

# A Fast Construction Method of the Erasure Code with Small Cross-Cloud Data Center Repair Traffic

Bao Han and Wang Yijie
（*National key Lab of Parallel and Distributed computing* (*National University of Defense Technology*), *Changsha* 410073）
（*College of Computer Science and Technology*, *National University of Defense Technology*, *Changsha* 410073）

**Abstract**　Compared with cross-cloud data center replication, cross-cloud data center erasure code is more reliable and space-efficiency. However, existing cross-cloud data center erasure codes cannot achieve low cross-cloud data center repair traffic, high encoding parameters adaptability, and high erasure code construction efficiency at the same time, so they are rarely used in production. We propose a fast construction method of the erasure code with small cross-cloud data center repair traffic, called FMEL, which can obtain the erasure code with small cross-cloud data center repair traffic quickly under different encoding parameters. Specifically, FMEL converts erasure code repair group distribution schemes and the corresponding encoding parameters into fixed-length feature vectors, and verifies whether the erasure code repair group distribution schemes match the encoding parameter by classifying corresponding feature vectors with a support vector machine—a feature vector positively indicates that the corresponding erasure code repair group distribution scheme passes the verification. Then, FMEL uses a parallel search algorithm to pick the erasure code repair group distribution scheme with the smallest cross-cloud data center repair traffic from all distribution schemes passing the verification, and converts it into the generator matrix of the erasure code with small cross-cloud data center repair traffic. Experiments in a cross-cloud data center environment show that FMEL can construct the optimal code that can achieve the lower bound of cross-cloud data center repair traffic under most encoding parameters. Meanwhile, FMEL's erasure code construction time is 89% less than the existing work's optimal code construction time. Compared with several popular erasure codes, the erasure code constructed by FMEL can reduce the cross-cloud data center repair traffic by from 42.9% to 56.0%.

**Key words**　cross-cloud data center storage; erasure code; disaster-tolerance; fault-tolerance; repair traffic

**摘　要**　近年来,云数据中心故障频发,因而各大机构纷纷采用跨云数据中心多副本技术对数据进行容灾存储.与跨云数据中心多副本技术相比,跨云数据中心纠删码技术可靠性更高、冗余度更低.但是,现有跨

**收稿日期**: 2022-06-16; **修回日期**: 2022-09-16
**基金项目**: 国家重点研发计划项目(2016YFB1000101); 国家自然科学基金项目(61379052); 国家教育部科研创新基金项目(2018A02002); 湖南省自然科学杰出青年基金项目(14JJ1026)
This work was supported by the National Key Research and Development Program of China(2016YFB1000101), the National Natural Science Foundation of China(61379052), the Science Foundation of Ministry of Education of China(2018A02002), and the Natural Science Foundation for Distinguished Young Scholars of Hunan Province(14JJ1026).
**通信作者**: 王意洁(wangyijie@nudt.edu.cn)

云数据中心纠删码技术无法同时满足低跨云数据中心修复流量、高编码参数适应性和高纠删码构造效率，因而尚未在生产系统中得到普遍应用. 提出一种低跨云数据中心修复流量的纠删码的快速构造方法 (fast construction method of the erasure code with small cross-cloud data center repair traffic, FMEL), 该方法可在不同编码参数下快速构造具有低跨云数据中心修复流量的纠删码. 具体而言，FMEL 首先将纠删码修复组分布方案及用户指定的编码参数转换为定长特征向量，并基于支持向量机对各特征向量进行快速分类以检验其对应纠删码修复组分布方案和编码参数的匹配性——某特征向量属于正类表示其对应纠删码修复组分布方案与编码参数相匹配. 而后，FMEL 用一种并行搜索算法从所有通过检验的纠删码修复组分布方案中选出平均跨云数据中心修复流量较小的一个方案，并用一种试错算法将其转换为具有低跨云数据中心修复流量的纠删码的生成矩阵. 跨云数据中心环境中的实验表明，与现有的可在不同编码参数下构造出能达到平均跨云数据中心修复流量下限的最优码的工作相比，FMEL 可将纠删码构造用时缩短 89%，且在大部分编码参数下，二者构造的纠删码的跨云数据中心修复流量相同. 此外，与其他几类常用纠删码相比，FMEL 构造的纠删码可将跨云数据中心修复流量降低 42.9%~56.0%.

**关键词** 跨云数据中心存储；纠删码；容灾；容错；修复流量

**中图法分类号** TP302.8

近年来，云数据中心故障频发，常导致其中数据长时间不可访问[1-6]. 同时，云际计算技术的逐渐成熟使得部署跨云数据中心存储系统更加便捷[7]. 因此，各机构纷纷采用跨云数据中心多副本技术来实现重要数据的容灾存储[8].

相较于跨云数据中心多副本技术，跨云数据中心纠删码技术具有更高的可靠性和更低的冗余度[9-12]. 然而，跨云数据中心纠删码技术要在生产系统中得到普遍运用，还需满足 3 方面要求：

1）低跨云数据中心修复流量. 纠删码技术在跨云数据中心存储系统中修复故障节点中的数据时，需跨云数据中心传输大量数据，即纠删码技术的跨云数据中心修复流量较大. 由于云数据中心间带宽往往远低于云数据中心内带宽，所以纠删码技术在跨云数据中心存储系统中修复数据的时间较长[13-15]. 因此，亟需降低纠删码的跨云数据中心修复流量.

2）高编码参数适应性. 在生产中，用户通常对存储节点数 $n$、冗余度 $n/k$、容错度 $d$ 和容灾度 $D$ 等纠删码编码参数具有多样化需求. 因此，有必要提高跨云数据中心纠删码的编码参数适应性. 编码参数为 $(n, k, d, D)$ 的纠删码将每 $k$ 个数据块编码为 $n$ 个编码块，并将其分别存储在位于多个云数据中心的 $n$ 个存储节点，且当任意 $d-1$ 个存储节点或任意 $D$ 个云数据中心故障时，其中存储的编码块可由其他编码块修复.

3）高纠删码构造效率. 因为纠删码编解码数据的过程与存储系统的读写过程深度耦合，所以开发和部署存储系统前需先完成纠删码的构造，因而用户通常对纠删码构造用时较为敏感. 因此，有必要提高跨云数据中心纠删码的构造效率.

现有工作提出的跨云数据中心纠删码[13-18] 虽然能在一定程度上降低跨云数据中心修复流量，但普遍存在编码参数适应性较低的问题，无法在满足用户对编码参数的多样化需求的同时有效降低跨云数据中心修复流量. 此外，有工作提出了面向跨云数据中心存储的自适应纠删码 ACIoT[19]，可求得不同编码参数 $(n,k,d)$ 下的跨云数据中心修复流量下限，并构造能达到该下限的纠删码，即最优码. 但是，ACIoT 需要消耗较长时间来检验纠删码修复组分布方案与编码参数 $(n,k,d)$ 的匹配性，故其纠删码构造效率较低. 纠删码修复组分布方案 $E$ 与指定编码参数 $\boldsymbol{P}$ 相匹配是指存在一个编码参数为 $\boldsymbol{P}$ 的纠删码的修复组分布方案为 $E$.

综上，现有工作无法兼顾编码参数适应性、纠删码跨云数据中心修复流量和纠删码构造效率. 本文提出一种低跨云数据中心修复流量的纠删码的快速构造方法（fast construction method of the erasure code with small cross-cloud data center repair traffic）FMEL，可在不同的编码参数下快速构造出具有较低跨云数据中心修复流量的纠删码. FMEL 的主要思想为：

首先，FMEL 将纠删码修复组分布方案及相应的编码参数 $(n,k,d,D)$ 转换为定长特征向量，并将检验纠删码修复组分布方案与指定编码参数匹配性的问题转换为定长特征向量的二分类问题. 其中，特征向量属于正类表示纠删码修复组分布方案与相应编码参数相匹配. 然后，FMEL 采用具有较高分类效率的支持向量机（support vector machine, SVM）来对各个特

征向量进行分类以实现其所对应纠删码修复组分布方案的快速检验. 在检验的同时，FMEL 不断采集新的训练集对 SVM 分类器进行增量更新，从而不断提高其分类准确率. 随后，FMEL 采用一种并行搜索方法来快速地从所有可通过 SVM 检验的纠删码修复组分布方案中选出跨云数据中心修复流量最小的一个方案. 最后，FMEL 采用一种基于试错的纠删码修复组分布方案转换方法将搜索到的纠删码修复组分布方案转换为具有低跨云数据中心修复流量的纠删码的生成矩阵.

在跨云数据中心环境中的实验表明，FMEL 在大部分编码参数下可构造出能达到平均跨云数据中心修复流量下限的最优码，且其构造纠删码的时间仅为现有工作构造最优码时间的 11%.

## 1 相关工作

### 1.1 单云数据中心纠删码

现有工作提出了 2 类低修复流量纠删码——再生码和局部性码. 再生码通过降低新生节点从各提供者节点里的编码块中读取的数据量来减少修复流量，局部性码通过降低各个编码块的提供者节点数量来减少修复流量. 与再生码相比，局部性码更容易实现且灵活性更高，因而被广泛地应用于 Amazon AWS[20]，Microsoft WAS[21]，Facebook HDFS-RAID[22] 等生产系统中. 特别地，Shahabinejad 等人[23] 提出了一种可达到平均修复流量下限的纠删码 ACAL.

虽然现有单云数据中心纠删码能够降低平均修复流量，但是跨云数据中心修复流量并不与修复流量正相关. 因此，这些纠删码在跨云数据中心环境下不能充分降低跨云数据中心修复流量.

### 1.2 跨云数据中心纠删码

Yu 等人[13] 提出了一种跨域容错纠删码 DFC，其基本思想是在传统的 RS 码的基础上引入局部校验块，首先使用 RS 码将 $k$ 个数据块编码为 $w$ 个编码块并在 $N$ 个云数据中心中各存储 $w/N(w/N \leqslant w-k)$ 个数据块. 由于 RS 码可以保证在这 $w$ 个编码块中的任意 $w-k$ 个编码块失效时重构原始数据，所以在任意一个云数据中心故障时，仍可以通过其他云数据中心里的 $w-w/N(w-w/N \geqslant k)$ 个编码块恢复出原始数据. 然后，DFC 为每个机架存放的编码块生成局部校验块，使得在任意一个编码块失效时，可以使用机架内的编码块和局部校验块对其进行修复. 因此，DFC 的跨云数据中心修复流量较小. 但是，DFC 对编码参数具有严格的限制，要求 $n-k-d+1 \geqslant N$.

Caneleo 等人[14] 提出了一种多倍异或码 MXOR，其基本思想是将数据块分为 2 行 $k/2$ 列，而后通过异或计算分别求得各行各列的局部校验块，然后将所有编码块和局部校验块按行分散到各云数据中心. 当单个编码块失效时，MXOR 可通过对云数据中心内部的其他编码块和局部校验块进行异或计算对其进行修复，因此 MXOR 的跨云数据中心修复流量较小. 但是，MXOR 对编码参数具有严格的限制，要求 $n/k \geqslant 2.4$ 且 $d \leqslant 4$.

Chen 等人[15-16] 和 Hu 等人[17] 分别提出 FMSR 码和 DRC 码均能有效降低跨云数据中心修复流量. 然而，FMSR 和 DRC 均对编码参数有严格的限制，FMSR 要求 $n-k=2$，而 DRC 码要求 $n, k, N$（云数据中心数）至少满足 2 个条件中的 1 个：1）$n=3z, k=2z, N=3$（$z$ 为正整数）；2）$N=n/(n-k)$.

虽然上述纠删码均能在一定程度上降低跨云数据中心修复流量，但它们普遍存在编码参数适应性较差的问题，无法在满足用户对编码参数的多样化需求的同时有效降低跨云数据中心修复流量. 为此，有工作[19] 提出了面向跨云数据中心存储的自适应纠删码 ACIoT，可求得不同编码参数 $(n,k,d)$ 下的最小跨云数据中心修复流量码，即最优码. 具体而言，ACIoT 首先定义了纠删码的修复组分布方案，该方案决定了纠删码的跨云数据中心修复流量. 然后，ACIoT 可枚举指定编码参数 $(n, k)$ 下的纠删码修复组分布方案，并检验它们是否与指定编码参数 $(n,k,d)$ 相匹配. 最后，ACIoT 可从所有与编码参数 $(n,k,d)$ 相匹配的纠删码修复组分布方案中找出具有最小跨云数据中心修复流量的 1 个并将其转换为最优码生成矩阵. 然而，ACIoT 忽略了容灾度对跨云数据中心修复流量下限的影响. 此外，因 ACIoT 需要消耗较长时间来检验纠删码修复组分布方案与编码参数 $(n,k,d)$ 的匹配性，故其纠删码构造用时较长.

综上，现有的跨云数据中心纠删码无法同时满足低跨云数据中心修复流量、高编码参数适应性和高纠删码构造效率，这严重限制了其在生产系统中的运用.

除以上聚焦于数据修复的工作外，Saeed 等人[24] 还考虑到 RS 码等纠删码技术在读取数据时具备多种读取方案，可以通过访问不同云数据中心的节点来重构用户数据. 因此，他们提出了距离优先读取策略和负载均衡优先读取策略，分别用于对读取过程的网络开销和各存储节点的负载进行优化. 同时，他

们对用户读取数据时的网络开销和各节点的负载进行了综合建模，得到了一个综合考虑 2 方面因素的开销模型，并基于此模型提出了跨数据中心纠删码数据读取节点选择算法 Sandooq，能够有效降低数据读取的综合开销. 此外，我们在之前的工作中提出了一种跨云数据中心纠删码增量写入方法[25]和一种基于生成矩阵变换的跨云数据中心纠删码写入方法[26]，可分别通过提高编码并行度和降低跨云数据中心写入流量来提高写入效率.

## 2 重要定义及定理

**定义 1.**生成矩阵. 跨云数据中心纠删码技术将用户数据划分为若干数据块，并将每 $k$ 个数据块（记为 $x_t, t\in[1,k]$）编码为 $n$ 个编码块（记为 $y_i, i\in[1,n]$），这 $n$ 个编码块被称为一个条带，编码过程如式（1）所示，其中 $\boldsymbol{G}$ 为纠删码的生成矩阵. 生成矩阵 $\boldsymbol{G}$ 的秩必须为 $k$，否则 $\boldsymbol{G}^{\mathrm{T}}(z_1,\cdots,z_k)^{\mathrm{T}}=(y_1,\cdots y_n)^{\mathrm{T}}$ 没有唯一解，即无法将一个条带中的 $n$ 个编码块解码为原始数据块.

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \boldsymbol{G}^T \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} = \begin{pmatrix} x_1 g_{11}+\cdots+x_k g_{k1} \\ \vdots \\ x_1 g_{1n}+\cdots+x_k g_{kn} \end{pmatrix}. \quad (1)$$

**定义 2.** 校验矩阵. 如果 $\boldsymbol{h}_{1^*}^{\mathrm{T}}, \boldsymbol{h}_{2^*}^{\mathrm{T}}, \cdots, \boldsymbol{h}_{(n-k)^*}^{\mathrm{T}}$ 为 $\boldsymbol{G}[z_1,\cdots,z_k]^{\mathrm{T}}=\boldsymbol{0}$ 的基础解系，那么 $\boldsymbol{H}=[\boldsymbol{h}_{1^*}^{\mathrm{T}}, \cdots, \boldsymbol{h}_{(n-k)^*}^{\mathrm{T}}]^{\mathrm{T}}$ 为对应于生成矩阵 $\boldsymbol{G}$ 的校验矩阵. 因为 $\boldsymbol{G}$ 的秩为 $k$，所以 $\boldsymbol{H}$ 的秩为 $n-k$.

**定义 3.** 修复组. 由生成矩阵 $\boldsymbol{G}$ 与校验矩阵 $\boldsymbol{H}$ 的定义有，$\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$，故 $(y_1,\cdots,y_n)\boldsymbol{H}^{\mathrm{T}}=(x_1,\cdots,x_k)\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$. 因此，每个编码块都可以通过对其他若干个编码块进行线性计算重构. 如果编码块 $y_i$ 可以通过对 $y_{i,1}, y_{i,2}, \cdots, y_{i,r}$ 进行线性计算重构，那么 $\{y_i, y_{i,1}, y_{i,2}, \cdots, y_{i,r}\}$ 为 $y_i$ 的一个修复组（用于修复 $y_i$ 的存储节点被称为新生节点，存储 $y_{i,1}, y_{i,2}\cdots, y_{i,r}$ 的节点被称为提供者节点）. 一个编码块可能有多个修复组. 此外，由于编码块之间的线性关系由生成矩阵 $\boldsymbol{G}$ 或校验矩阵 $\boldsymbol{H}$ 决定，所以编码块的修复组也是由生成矩阵 $\boldsymbol{G}$ 或校验矩阵 $\boldsymbol{H}$ 决定的.

**定义 4.** 编码块分布方案. 一个条带中的编码块所在的云数据中心的编号组成的集合为该条带的编码块分布方案 $R=\{z_1,z_2,\cdots,z_n\}$，其中 $z_i$ 表示该条带中第 $i$ 个编码块位于第 $z_i$ 个云数据中心.

**定义 5.** 编码块修复组分布方案. 设 $T_{i,w}$ 为编码块 $y_i$ 的第 $w$ 个修复组，如果 $T_{i,w}$ 中的编码块分布于 $t$ 个编号分别为 $z_1,z_2,\cdots,z_t$ 的云数据中心中（这 $t$ 个云数据中心分别记为 $DC_{z_1}, DC_{z_2},\cdots,DC_{z_t}$），那么令 $R_{i,w}=\{z_1,z_2,\cdots,z_t\}$. 设编码块 $y_i$ 共有 $Q_i$ 个修复组，记为 $T_{i,1}, T_{i,2},\cdots,T_{i,Q_i}$，且 $\left|R_{i,q}\right|=\min\limits_{w=1}^{Q_i}\left(\left|R_{i,w}\right|\right)$（$q\in[1,Q_i]$），那么 $R_{i,q}$ 为编码块 $y_i$ 的修复组分布方案. 其中，$\left|R_{i,w}\right|$ 表示 $R_{i,w}$ 中含有的云数据中心编号个数.

**定义 6.** 纠删码修复组分布方案.如果 $C_1,C_2,\cdots,C_n$ 分别为编码块 $y_1,y_2,\cdots,y_n$ 的修复组分布方案，那么 $\{C_1,C_2,\cdots,C_n\}$ 为条带 $\{y_1,y_2,\cdots,y_n\}$ 的修复组分布方案. 通常而言，纠删码的不同编码条带中的编码块在各个云数据中心间的分布相同. 在此情况下，纠删码的各个条带的修复组分布方案相同，因而条带的修复组分布方案也被称为纠删码修复组分布方案.

为了降低跨云数据中心修复流量，在基于纠删码技术的跨云数据中心存储系统修复编码块时，含有提供者节点的云数据中心通常会先将其中的提供者节点的编码块合并为一个和各个编码块大小相同中间块，然后再将中间块发往新生节点. 此外，为了保持系统的容灾度和负载均衡性不变，失效编码块的新生节点通常和该失效编码块位于同一云数据中心. 在此情况下，如果编码块 $y_i$ 的修复组分布方案为 $C_i$ 且编码块大小为 $m$，那么在修复 $y_i$ 时需要向其新生节点发送中间块的云数据中心（含新生节点所在云数据中心）的个数为 $C_i$ 中含有的云数据中心编号个数 $|C_i|$，因而修复 $y_i$ 的最小跨云数据中心传输量为 $m(|C_i|-1)$. 所以，一个纠删码的修复组分布方案为 $E$ 的纠删码的编码条带的平均跨云数据中心流量为 $m\sum\limits_{C\in E}(|C|-1)/n$，其中，$C$ 为 $E$ 中的编码块修复组分布方案，$n$ 为一个编码条带中的编码块数. 因此，纠删码的平均跨云数据中心修复流量由其修复组分布方案决定.

**定义 7.** 纠删码 Tanner 图[23]. 若一个编码参数为 $(n,k,d,D)$ 的纠删码 $C_{\mathrm{O}}$ 的校验矩阵为 $\boldsymbol{H}$，那么该纠删码的 Tanner 图 $\mathcal{T}$ 为满足 3 个条件的二分图：1)$\mathcal{T}$ 的一个端点集包含 $n$ 个变量端点（variable node, VN）；2)$\mathcal{T}$ 的另一个端点集包含 $n-k$ 个校验端点（check node, CN）；3）当且仅当 $\boldsymbol{H}$ 的第 $i$ 行、第 $j$ 列的元素不为 0，$\mathcal{T}$ 的第 $j$ 个校验端点覆盖其第 $i$ 个变量端点，即 $\mathcal{T}$ 中有一条以第 $j$ 个校验端点和第 $i$ 个变量端点为端点的边. 图 1 举例说明了纠删码的 Tanner 图与纠删码校验矩阵 $\boldsymbol{H}$ 间的关系.

令 $\boldsymbol{H}=\left(h_{ji}\right)_{(n-k)\times n}$ 为纠删码 $C_{\mathrm{O}}$ 的校验矩阵，$\{y_1, y_2,\cdots,y_n\}$ 为 $C_{\mathrm{O}}$ 的任意一个条带、$\mathcal{T}$ 为 $C_{\mathrm{O}}$ 的 Tanner 图. 因为 $(y_1,\cdots,y_n)\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$，所以 $h_{j1}y_1+h_{j2}y_2+\cdots+h_{jn}y_n=0$. 由定义 7 有，如果 $CN_j$ 仅覆盖 $VN_{l_1}, VN_{l_2},\cdots,VN_{l_r}$（对于任

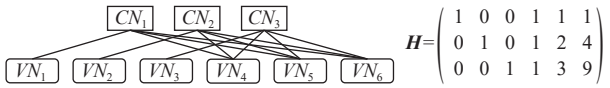


Fig. 1 Erasure code's Tanner graph

图 1 纠删码 Tanner 图

意$t\in[1,r]$有 $l_t\in[1,n]$), 那么 $h_{j1},h_{j2},\cdots,h_{jn}$中仅有 $h_{jl_1}$, $h_{jl_2},\cdots,h_{jl_r}$不为 0. 因此, $h_{jl_1}y_{l_1}+h_{jl_2}y_{l_2}+\cdots+h_{jl_r}y_{l_r}=0$, 其中 $h_{jl_1},h_{jl_2},\cdots,h_{jl_r}$均不为 0, 即 $y_{l_1},y_{l_2},\cdots,y_{l_r}$为一个修复组. 因此, 纠删码 Tanner 图的每个变量端点对应于 1 个编码块($VN_i$对应于$y_i$)且每个校验端点对应 1 个修复组.

图 2 举例说明了纠删码 Tanner 图与编码块和修复组之间的关系. 纠删码的一个编码条带中的 6 个编码块 $y_1,y_2,\cdots,y_6$ 分别对应着该纠删码 Tanner 图的 6 个变量端点 $VN_1,VN_2,\cdots,VN_6$. 覆盖纠删码 Tanner 图中的第 1,4,5,6 个变量端点 $VN_1,VN_4,VN_5,VN_6$ 的校验端点 $CN_1$ 对应的修复组为 $\{y_1,y_4,y_5,y_6\}$, 该修复组内各编码块之间的线性关系为 $y_1+y_4+y_5+y_6=0$. 同理, 校验端点 $CN_2$ 对应的修复组为 $\{y_2,y_4,y_5,y_6\}$, 该修复组内各编码块的线性关系为 $y_2+y_4+2y_5+4y_6=0$; 校验端点 $CN_3$ 对应的修复组为 $\{y_3,y_4,y_5,y_6\}$, 该修复组内各编码块的线性关系为 $y_3+y_4+3y_5+9y_6=0$.

本文涉及的常用符号及其含义如表 1 所示:

**定理 1.** 假设$\boldsymbol{H}_{(n-k)\times n}$为对应于纠删码 Tanner 图$\mathcal{T}$的一个校验矩阵、$\boldsymbol{H}_1$为 $\boldsymbol{H}$ 的任意一个 $n-k$ 行 $c$ 列($c\leqslant n-k$)的子矩阵. 如果$\mathcal{T}$的最大匹配数小于 $c$, 那么$\boldsymbol{H}_1$的秩小于 $c$, 即$\boldsymbol{H}_1$不可能列满秩.

证明. 假设: $\boldsymbol{H}_1=((h_{1l_1},\cdots,h_{(n-k)l_1})^{\mathrm{T}},\cdots,(h_{1l_c},\cdots,h_{(n-k)l_c})^{\mathrm{T}})$; $\boldsymbol{H}_2$ 为 $\boldsymbol{H}_1$ 的任意一个 $c$ 行 $c$ 列的子矩阵(不妨设为$((h_{1l_1},\cdots,h_{cl_1})^{\mathrm{T}},\cdots,(h_{1l_c},\cdots,h_{cl_c})^{\mathrm{T}})$); $\mathcal{T}'$为$\mathcal{T}$的一个子图, 其边集为$\{CN_1,CN_2,\cdots,CN_{n-k},VN_{l_1},VN_{l_2},\cdots,VN_{l_c}\}$; $W=\{\{l_{u_1},l_{u_2},\cdots,l_{u_c}\}|u_1,u_2,\cdots,u_c\in\{1,2,\cdots,c\}$, 当$a\neq b$时一定有$u_a\neq u_b\}$; $D_{\mathrm{E}}=\{L|L=\{h_{1l_{u_1}},h_{2l_{u_2}},\cdots,h_{cl_{u_c}}\};\{l_{u_1},l_{u_2},\cdots,l_{u_c}\}\in W\}$.

如果存在一个$L\in D_{\mathrm{E}}$不含有零元素(不妨设$L=\{h_{1l_1},h_{2l_2},\cdots,h_{cl_c}\}$), 那么$\mathcal{T}'$的最大匹配数为 $c$(对应的最大匹配为$\{\langle CN_1,VN_{l_1}\rangle,\langle CN_2,VN_{l_2}\rangle,\cdots,\langle CN_c,VN_{l_c}\rangle\}$), 这与假设($\mathcal{T}$的最大匹配数小于 $c$)相悖. 因此, 每个$L\in D_{\mathrm{E}}$至少包含一个零元素. 因此, $|\boldsymbol{H}_2|=\sum_{\{l_{u_1},l_{u_2},\cdots,l_{u_c}\}\in W}(-1)^{A(l_{u_1},l_{u_2},\cdots,l_{u_c})}(h_{1l_{u_1}},h_{2l_{u_2}},\cdots,h_{cl_{u_c}})=0$, 其中$A(l_{u_1},l_{u_2},\cdots,l_{u_c})$为$l_{u_1},l_{u_2},\cdots,l_{u_c}$的逆序数. 因此, $\boldsymbol{H}_1$的任意 $c$ 行 $c$ 列的子矩阵都不满秩. 所以, $\boldsymbol{H}_1$的秩小于 $c$, 即$\boldsymbol{H}_1$不是列满秩的. 证毕.

**定理 2.** 假设$\boldsymbol{H}_{(n-k)\times n}$为纠删码 $C_{\mathrm{O}}$ 的校验矩阵. 当且仅当由 $\boldsymbol{H}$ 的第 $z_1\sim z_c$列组成的矩阵$\boldsymbol{H}_1$的秩为 $c$, $C_{\mathrm{O}}$能够在其各条带中的第$l_1\sim l_c$个编码块同时失效时修复这些失效编码块.

证明. 从纠删码的校验方程$(y_1,\cdots,y_n)\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$中得到的以$y_{l_1},y_{l_2},\cdots,y_{l_c}$为未知数的方程组如式(2)所示:

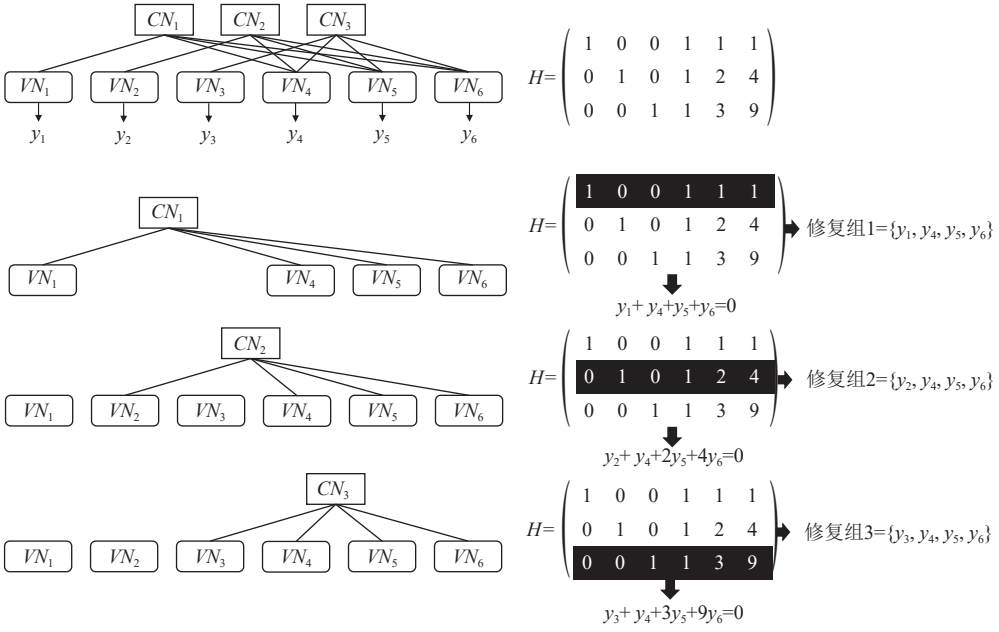


Fig. 2 The relationship between the erasure code's Tanner graph, coded blocks, and the repair groups

图 2 纠删码 Tanner 图与编码块和修复组之间的关系

**Table 1 Notations and Presentations of the Parameters**

**表 1 参数符号及其含义**

| 符号 | 含义 |
|---|---|
| $N$ | 数据中心总数 |
| $z_s$ | 数据中心编号 |
| $DC_{zs}$ | 数据中心 |
| $n$ | 编码条带中的编码块数 |
| $k$ | 原始条带中的数据块数 |
| $d$ | 容灾度 |
| $D$ | 容错度 |
| $C_{\mathrm{O}}$ | 纠删码 |
| $y_i$ | 编码块 |
| $\mathrm{x}_j$ | 数据块 |
| $\boldsymbol{H}$ | 校验矩阵 |
| $\boldsymbol{h}_{j*}, \boldsymbol{h}_{*i}$ | 校验矩阵的行向量、列向量 |
| $h_j i$ | 校验矩阵中的元素 |
| $\boldsymbol{G}$ | 生成矩阵 |
| $\boldsymbol{g}_{j*}, \boldsymbol{g}_{i*}$ | 生成矩阵的行向量、列向量 |
| $g_{ji}$ | 生成矩阵中的元素 |
| $\mathcal{T}$ | 纠删码 Tanner 图 |
| $CN_j$ | 纠删码 Tanner 图中的校验端点 |
| $VN_i$ | 纠删码 Tanner 图中的变量端点 |
| $T$ | 修复组 |
| $R$ | 编码块分布方案 |
| $C$ | 编码块修复组分布方案 |
| $E$ | 纠删码修复组分布方案 |
| $m$ | 编码块的大小 |
| $\boldsymbol{P}$ | 编码参数 |
| $P_{\mathrm{o}}$ | 抽样概率 |

$$\boldsymbol{H}_1\begin{pmatrix} y_{l_1} \\ \vdots \\ y_{l_c} \end{pmatrix} = \begin{pmatrix} -\left(h_{1l_{c+1}}y_{l_{c+1}} + h_{1l_{c+2}}y_{l_{c+2}} + \cdots + h_{1l_n}y_{l_n}\right) \\ \vdots \\ -\left(h_{(n-k)l_{c+1}}y_{l_{c+1}} + h_{(n-k)l_{c+2}}y_{l_{c+2}} + \cdots + h_{(n-k)l_n}y_{l_n}\right) \end{pmatrix}. \tag{2}$$

因为式（2）中的最大线性无关方程数等于$\boldsymbol{H}_1$的秩，所以，当且仅当$\boldsymbol{H}_1$的秩为$c$时，式（2）有唯一解. 因此，当且仅当$\boldsymbol{H}_1$的秩为$c$时，$y_{l_1}, y_{l_2}, \cdots, y_{l_c}$能够被同条带中的其他编码块修复. 证毕.

**定理 3.** 假设纠删码的编码块分布方案$R$已确定，即任意条带中的$n$个编码块被分配给$N$个云数据中心，亦即纠删码 Tanner 图$\mathcal{T}$的$n$个变量端点和相应的校验矩阵$\boldsymbol{H}$的$n$个列被分配给了$N$个云数据中心（根据定义 7，纠删码 Tanner 图$\mathcal{T}$的$n$个变量端点分别对应于校验矩阵$\boldsymbol{H}$的$n$个列和任意条带中的$n$个编码块）. 在此假设下，$\mathcal{T}$匹配于编码参数（$n,k,d,D$）的一个充要条件是：$\mathcal{T}$有$n-k$个校验端点、$n$个变量端点（子条件 1）；$\mathcal{T}$的任意$\gamma$个校验端点至少覆盖$\gamma+k$个变量端点，其中，$\gamma \in [J, n-k]$且$J=n-k-d+2$（子条件 2）；$\mathcal{T}$的最大匹配数不小于$n-k$（子条件 3）；任意含有$B(B \leqslant n-k)$个变量端点的$D$个云数据中心中的任意$\beta(\beta \in [1, B])$个变量端点至少覆盖个校验端点（子条件 4）.

证明. 1）必要性证明.

① 子条件 1 的必要性.

根据定义 7，若$\mathcal{T}$匹配于编码参数（$n,k,d,D$），那么其对应的纠删码的生成矩阵一定有$k$行$n$列. 根据纠删码 Tanner 图与纠删码的生成矩阵的关系，$\mathcal{T}$一定有$n-k$个校验端点、$n$个变量端点，即$\mathcal{T}$一定满足子条件 1.

② 子条件 2 的必要性.

如果在$\mathcal{T}$中存在$\gamma$个校验端点（不妨设为$CN_1, CN_2, \cdots, CN_\gamma$）仅覆盖了$w$个变量端点（不妨设为$VN_1, VN_2, \cdots, VN_w$）且$w \leqslant \gamma+k-1$，那么根据定义 7，对应于$\mathcal{T}$的校验矩阵如式（3）所示. 其中，$\boldsymbol{0}_{\gamma\times(n-w)}$为全零矩阵.

$$\boldsymbol{H} = \begin{pmatrix} \boldsymbol{A}_{(\gamma\times w)} & \boldsymbol{0}_{\gamma\times(n-w)} \\ \boldsymbol{B}_{(n-k-\gamma)\times w} & \boldsymbol{C}_{(n-k-\gamma)\times(n-w)} \end{pmatrix}. \tag{3}$$

（i）如果$n-w \leqslant d-1$，从纠删码的校验方程$(y_1, \cdots, y_n)\boldsymbol{H}^{\mathrm{T}}=0$中只能得到以$y_{w+1}, y_{w+2}, \cdots, y_n$为未知数的线性方程组：

$$\begin{cases} h_{(\gamma+1)(w+1)}y_{w+1} + h_{(\gamma+1)(w+2)}y_{w+2} + \cdots + h_{(\gamma+1)n}y_n = \\ -\left(h_{(\gamma+1)1}y_1 + h_{(\gamma+1)1}y_2 + \cdots + h_{(\gamma+1)w}y_w\right), \\ \vdots \\ h_{(n-k)(w+1)}y_{w+1} + h_{(n-k)(w+2)}y_{w+2} + \cdots + h_{(n-k)n}y_n = \\ -\left(h_{(n-k)1}y_1 + h_{(n-k)2}y_2 + \cdots + h_{(n-k)w}y_w\right). \end{cases} \tag{4}$$

因为$n-w \leqslant d-1$，所以$n-w \geqslant n-(\gamma+k-1)=n-k-\gamma+1$. 所以，式（4）中的具有$n-w$个未知数和$n-k-\gamma$个方程的方程组无唯一解. 因此，$y_{w+1}, y_{w+2}, \cdots, y_n$不能由同条带中的其他编码块修复. 因此，$\mathcal{T}$不匹配于编码参数$d$.

（ii）如果$n-w > d-1$，从纠删码的校验方程$[y_1, \cdots, y_n]\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$中只能得到以$y_{n-d+2}, y_{n-d+3}, \cdots, y_n$为未知数的线性方程组：

$$\begin{cases} h_{(\gamma+1)(n-d+2)}y_{n-d+2} + h_{(\gamma+1)(n-d+3)}y_{n-d+3} + \cdots + h_{(\gamma+1)n}y_n = \\ -\left(h_{(\gamma+1)1}y_1 + h_{(\gamma+1)2}y_2 + \cdots + h_{(\gamma+1)(n-d+1)}y_{n-d+1}\right), \\ \vdots \\ h_{(n-k)(n-d+2)}y_{n-d+2} + h_{(n-k)(n-d+3)}y_{n-d+3} + \cdots + h_{(n-k)n}y_n = \\ -\left(h_{(n-k)1}y_1 + h_{(n-k)2}y_2 + \cdots + h_{(n-k)(n-d+1)}y_{n-d+1}\right). \end{cases} \tag{5}$$

因为$\gamma\in[J,n-k]$且$J=n-k-d+2$，所以$n-k-\gamma\leqslant n-k-(n-k-d+2)=d-2$. 所以，式（5）中的具有$d-1$个未知数和$n-k-\gamma$个方程的方程组无唯一解. 因此，$y_{n-d+2},y_{n-d+3},\cdots,y_n$不能由同条带的其他编码块修复. 因此，$\mathcal{T}$不匹配于编码参数$d$.

综上，如果$\mathcal{T}$匹配于编码参数$(n,k,d,D)$，$\mathcal{T}$的任意$\gamma$个校验端点至少覆盖$\gamma+k$个变量端点，其中$\gamma\in[J,n-k]$且$J=n-k-d+2$.

③ 子条件3的必要性证明.

假设$\mathcal{T}$的最大匹配数为$b<n-k$且$\boldsymbol{H}$(含有$n-k$行、$n$列）为任意一个对应于$\mathcal{T}$的校验矩阵，那么由定理1有，$\boldsymbol{H}$的每个$n-k$行、$n_k$列的子矩阵均不满秩. 因此，$\boldsymbol{H}$的秩不为$n-k$. 因此，根据校验矩阵的定义，$\boldsymbol{H}$不可能是一个$(n,k,d,D)$纠删码的校验矩阵. 因此，$\mathcal{T}$不匹配于$(n,k,d,D)$. 因此，若$\mathcal{T}$匹配于编码参数$(n,k,d,D)$，$\mathcal{T}$的最大匹配数不小于$n-k$.

④ 子条件4的必要性证明.

如果存在$D$个云数据中心（不妨设为$DC_{z_1},DC_{z_2},\cdots,DC_{z_D}$），其中的$\beta$个变量端点（不妨设为$VN_1,VN_2,\cdots,VN_\beta$）只覆盖了$c\leqslant\beta-1$个校验端点（不妨设为$CN_1,CN_2,\cdots,CN_c$），那么根据定义7，对应于$\mathcal{T}$的校验矩阵$\boldsymbol{H}$中只有$c$个行中的第1~$\beta$列中存在非零元素. 因此，从纠删码的校验方程$(y_1,\cdots,y_n)\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$中只能得到以$y_1,y_2,\cdots,y_\beta$为变量的线性方程组：

$$\begin{cases} h_{11}y_1+h_{12}y_2+\cdots+h_{1\beta}y_\beta= \\ -(h_{1(\beta+1)}y_{\beta+1}+h_{1(\beta+2)}y_{\beta+2}+\cdots+h_{1n}y_n), \\ \quad\vdots \\ h_{c1}y_1+h_{c2}y_2+\cdots+h_{c\beta}y_\beta= \\ -\left(h_{c(\beta+1)}y_{\beta+1}+h_{c(\beta+2)}y_{\beta+2}+\cdots+h_{cn}y_n\right). \end{cases} \tag{6}$$

因为式（6）中的$\beta$元方程组的最大线性无关方程数为$c(c<\beta)$，所以该方程组没有唯一解. 因此，对应于$\mathcal{T}$的纠删码的任意编码条带$\{y_1,y_2,\cdots,y_n\}$中的$y_1,y_2,\cdots,y_\beta$不能被同条带的其他编码块修复($\mathcal{T}$对应的纠删码不能容忍$DC_{z_1},DC_{z_2},\cdots,DC_{z_D}$同时失效). 因此，$\mathcal{T}$不匹配于编码参数$(n,k,d,D)$.

综上，若$\mathcal{T}$匹配于编码参数$(n,k,d,D)$，任意含有$B(B\leqslant n-k)$个变量端点的$D$个云数据中心中的任意$\beta(\beta\in[1,B])$个变量端点至少覆盖$\beta$个校验端点.

2）充分性证明.

证明. 假设纠删码Tanner图$\mathcal{T}$符合定理3中的充要条件，那么可按5个步骤构造出一个纠删码Tanner图为$\mathcal{T}$的匹配于编码参数$(n,k,d,D)$的纠删码的生成矩阵，即$\mathcal{T}$匹配于编码参数$(n,k,d,D)$.

① 求得$\mathcal{T}$的最大匹配. 因为$\mathcal{T}$满足定理3中的充要条件的子条件3，所以其最大匹配包含$n-k$个变量端点. 如果$\mathcal{T}$的最大匹配包含$VN_{l_1},VN_{l_2},\cdots,VN_{l_{n-k}}$，那么令$\boldsymbol{H}_1$为由$\boldsymbol{H}$的第$l_1,l_2,\cdots,l_{n-k}$列组成的矩阵. 由于$\boldsymbol{H}_1$的行数和列数均为$n-k$，所以其为方阵. 然后，我们将$\boldsymbol{H}_1$加入集合$LS$.

② 求得$\boldsymbol{H}$的所有的包含$n-k$行、$d-1$列的子矩阵的集合$SM_1$以及$\boldsymbol{H}$的所有的包含其对应于任意$D$个云数据中心的列的子矩阵组成的集合$SM_2$.

③ 根据纠删码Tanner图的定义，$SM_1$和$SM_2$中每个矩阵都对应于$\mathcal{T}$的一个子图. 因此，我们求得对应于$SM_1$和$SM_2$中每个矩阵所对应的$\mathcal{T}$的子图的最大匹配. 而后，对于$SM_1$和$SM_2$中的每个矩阵$\boldsymbol{Z}$，如果其对应的$\mathcal{T}$的子图的最大匹配包含$CN_{l_1},CN_{l_2},\cdots,CN_{l_c}$，则将其第$l_1,l_2,\cdots,l_c$行组成的子矩阵加入集合$LS$.

可以证明，如果$SM_1$或$SM_2$中的某个矩阵有$q$列，那么其对应的$\mathcal{T}$的子图的最大匹配一定为$q$（即$LS$中的矩阵均为方阵），证明过程为：

首先，证明如果$SM_1$中的某个矩阵有$q$列，那么它的最大匹配数一定为$q$. 由于$SM_1$的矩阵的列数恒为$d-1$，因此只需要证明$SM_1$中的矩阵对应的$\mathcal{T}$的子图的最大匹配数恒为$d-1$.

（i）令$\partial=n-k-\gamma+1$(因为$\gamma\in[J,n-k]$且$J=n-k-d+2$，所以$\partial\in[1,d-1]$)，可以证明$\mathcal{T}$中任意$\partial$个变量端点至少覆盖$\partial$个校验端点：如果存在$\partial$个变量端点仅仅覆盖$u\leqslant\partial-1$个校验端点，那么剩下的$n-k-u\geqslant\gamma$个校验端点最多覆盖$n-\partial=k+\gamma-1$个变量端点，这与定理3中的充要条件的子条件2（任意$\gamma$个校验端点至少覆盖$\gamma+k$个变量端点）相矛盾. 因此，任意$\partial$个变量端点至少覆盖$\partial$个校验端点.

（ii）根据Hall定理[27]，在$\mathcal{T}$的任意含有$d-1$个变量端点、$n-k$个校验端点（$d-1<n-k$）的子图存在完全匹配的充要条件是该子图中任意$\partial$个变量端点至少覆盖$\partial$个校验端点（$\partial\in[1,d-1]$). 由（i）有，$\mathcal{T}$中任意$\partial$个变量端点至少覆盖$\partial$个校验，所以$\mathcal{T}$的任意含有$d-1$个变量端点、$n-k$个校验端点（$d-1<n-k$）的子图存在完全匹配，即$\mathcal{T}$中任意$d-1$个变量端点可以一对一地覆盖$d-1$个校验端点.

（iii）假设$\boldsymbol{Z}$为$SM_1$中的任意矩阵且$\boldsymbol{Z}$包含$\boldsymbol{H}$的第$l_1,l_2,\cdots,l_{d-1}$列. 由（ii）有，$VN_{l_1},VN_{l_2},\cdots,VN_{l_{d-1}}$一定一对一地覆盖$d-1$个校验端点（不妨设为$CN_1,CN_2,\cdots,CN_{d-1}$). 因此，$\{\langle CN_1,VN_{l_1}\rangle,\langle CN_2,VN_{l_2}\rangle,\cdots,\langle CN_{d-1},VN_{l_{d-1}}\rangle\}$为对应于$\boldsymbol{Z}$的$\mathcal{T}$的子图的最大匹配. 因此$SM_1$中任意矩阵对应的$\mathcal{T}$的子图的最大匹配数为

$d$−1.

然后，证明如果 $SM_2$ 中的某个矩阵有 $B$ 列，那么它的最大匹配数一定为 $B$.

（i）不妨假设 $\mathcal{T}$ 中对应于任意 $D$ 个云数据中心的任意 $B$ 个变量端点为 $VN_1,VN_2,\cdots,VN_B$. 根据 Hall 定理[27]，$\mathcal{T}$ 的由其 $VN_1,VN_2,\cdots,VN_B$ 和 $n-k$ 个校验端点组成的子图存在完全匹配的充要条件是该子图中任意 $\beta$ 个变量端点至少覆盖 $\beta$ 个校验端点（$\beta\in[1,B]$）. 因为 $\mathcal{T}$ 满足定理 3 中的充要条件的子条件 4，即任意含有 $B(B\leqslant nk)$ 个变量端点的 $D$ 个云数据中心中的任意 $\beta(\beta\in[1,B])$ 个变量端点至少覆盖 $\beta$ 个校验端点，所以 $\mathcal{T}$ 的由其 $VN_1,\cdots,VN_B$ 和 $n-k$ 个校验端点组成的子图存在完全匹配，即 $\mathcal{T}$ 中对应于任意 $D$ 个云数据中心的任意 $B$ 个变量端点一定可以一对一地覆盖 $B$ 个校验端点.

（ii）假设 $\boldsymbol{Z}$ 为 $SM_2$ 中的任意矩阵且 $\boldsymbol{Z}$ 包含了 $\boldsymbol{H}$ 的第 $l_1,l_2,\cdots,l_B$ 列. 由（i）有，$VN_{l_1},VN_{l_2},\cdots,VN_{l_B}$ 一定一对一地覆盖 $d$−1 个校验端点（不妨设为 $CN_1,CN_2,\cdots,CN_{l_B}$）. 因此，$\{\langle CN_1,VN_{l_1}\rangle,\langle CN_2,VN_{l_2}\rangle,\cdots,\langle CN_B,VN_{l_B}\rangle\}$ 为对应于 $\boldsymbol{Z}$ 的 $\mathcal{T}$ 的子图的最大匹配. 因此 $SM_2$ 中任意矩阵对应的 $\mathcal{T}$ 的子图的最大匹配的边数为 $B$.

④ 根据以上推导，$LS$ 中的矩阵均为方阵，且它们对应的 $\mathcal{T}$ 的子图的最大匹配等于它们的秩. 对任意 $\boldsymbol{Z}_t\in LS$，不妨设对应于 $\boldsymbol{Z}_t$ 的 $\mathcal{T}$ 的子图的最大匹配为 $M=\{\langle CN_{l_{t,1}},VN_{s_{t,1}}\rangle,\langle CN_{l_{t,2}},VN_{s_{t,2}}\rangle,\cdots,\langle CN_{l_{t,b_t}},VN_{s_{t,b_t}}\rangle\}$，那么可以通过对 $\boldsymbol{Z}_t$ 进行初等变换得到矩阵：

$$\boldsymbol{Z}_t'=\begin{pmatrix} h_{l_{t,1}s_{t,1}} & \cdots & h_{l_{t,1}s_{t,b_t}} \\ \vdots & & \vdots \\ h_{l_{t,b_t}s_{t,1}} & \cdots & h_{l_{t,b}s_{t,b_t}} \end{pmatrix}. \tag{7}$$

其中，$h_{l_{t,1}s_{t,1}},h_{l_{t,2}s_{t,2}},\cdots,h_{l_{t,b_t}s_{t,b_t}}$ 不为 0.

因此，对 $\boldsymbol{Z}_t'$ 进行初等行变换可得矩阵：

$$\boldsymbol{Z}_t''=\begin{pmatrix} h_{l_{t,1}s_{t,1}}-A_{l_{t,1}s_{t,1}} & \cdots & \cdots & \cdots \\ 0 & h_{l_{t,2}s_{t,2}}-A_{l_{t,2}s_{t,2}} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & h_{l_{t,b_t}s_{t,b_t}}-A_{l_{t,b_t}s_{t,b_t}} \end{pmatrix}. \tag{8}$$

其中，$A_{l_as_a}$ 为不包含 $h_{l_as_a}$ 的线性多项式（$a\in[1,b]$）. 因为 $A_{l_{t,a}s_{t,a}}$ 为不包含 $h_{l_{t,a}s_{t,a}}$ 的线性多项式（$t\in\{1,|LS|\}$，$a\in\{1,2,\cdots,b_t\}$），$\prod_{t\in[1,|LS|]}\prod_{a\in\{1,2,\cdots,b_t\}}\left(h_{l_{t,a}s_{t,a}}-A_{l_{t,a}s_{t,a}}\right)\neq 0$ 一定有解. 该不等式的解可以使所有的 $\boldsymbol{Z}_t''$ 的行列式均不为 0，即所有的 $\boldsymbol{Z}_t''$ 满秩. 这意味着：（i）$LS$ 中的 $\boldsymbol{H}_1$ 满秩，即 $\boldsymbol{H}$ 满秩；（ii）$SM_1$ 和 $SM_2$ 中的矩阵满秩. 由定理 2 可知，此时 $\boldsymbol{H}$ 对应的纠删码 $C_{\mathrm{O}}$ 能够容忍任意 $d$−1 个编码块失效或任意 $D$ 个云数据中心失效（即 $\boldsymbol{H}$ 为一个匹配于（$n,k,d,D$）的校验矩阵）.

⑤ 由于生成矩阵 $\boldsymbol{G}$ 和校验矩阵 $\boldsymbol{H}$ 满足 $\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{0}$，所以在得到匹配于编码参数（$n,k,d,D$）的校验矩阵 $\boldsymbol{H}$ 后即可求得相应的匹配于编码参数的（$n,k,d,D$）的生成矩阵 $\boldsymbol{G}$. 证毕.

**定理 4.** 纠删码修复组分布方案 $E$ 匹配于编码参数（$n,k,d,D$）的一个必要条件为：设 $ST$ 为 $E$ 中所有编码块修复组分布方案的无重复集合，对于任意 $D$ 个共含有 $B$ 个编码块的云数据中心（不妨设这些云数据中心的编号为 $z_1,z_2,\cdots,z_D$，即设这些云数据中心为 $DC_{z_1},DC_{z_2},\cdots,DC_{z_D}$），$ST$ 中仅含有不多于 $n-k-B$ 个不包含这些云数据中心的编号集合 $\{z_1,z_2,\cdots,z_D\}$ 中任意元素的编码块修复组分布方案.

证明. 根据定义 6，如果 $ST$ 中含有超过 $n-k-B$ 个不包含 $\{z_1,z_2,\cdots,z_D\}$ 中任意元素的编码块修复组分布方案，那么相应的纠删码有超过 $n-k-B$ 个不含有 $DC_{z_1},DC_{z_2},\cdots,DC_{z_D}$ 中编码块的修复组. 因此，根据定义 7，在任何对应于 $E$ 的纠删码 Tanner 图（不妨设为 $\mathcal{T}$）中，一定有超过 $n-k-B$ 个校验端点没有覆盖对应于 $DC_{z_1},DC_{z_2},\cdots,DC_{z_D}$ 中编码块的变量端点. 因此，在 $\mathcal{T}$ 中，对应于 $DC_{z_1},DC_{z_2},\cdots,DC_{z_D}$ 中编码块的变量端点（不妨设为 $VN_{l_1},VN_{l_2},\cdots,VN_{l_B}$）覆盖的校验端点少于 $B$ 个. 因此，根据定义 7，任意对应于 $\mathcal{T}$ 的校验矩阵 $\boldsymbol{H}$ 中的第 $l_1,l_2,\cdots,l_B$ 列组成的矩阵 $\boldsymbol{H}_1$ 最多只有 $B$−1 个非零行. 因此，$\boldsymbol{H}_1$ 的秩小于 $B$. 因此，根据定理 2，校验矩阵为 $\boldsymbol{H}$ 的纠删码不能容忍任意编码条带中的第 $l_1,l_2,\cdots,l_B$ 个编码块失效，即不能容忍 $DC_{z_1},DC_{z_2},\cdots,DC_{z_D}$ 同时失效. 因此，纠删码修复组分布方案 $E$ 不匹配于编码参数（$n,k,d,D$）. 证毕.

**定理 5.** 所有匹配于指定编码参数（$n,k,d,D$）的纠删码修复组分布方案的平均跨数据中心修复流量的最小值为编码参数（$n,k,d,D$）下的纠删码的平均跨数据中心修复流量下限.

证明. 令所有匹配于指定编码参数（$n,k,d,D$）的纠删码修复组分布方案的平均跨数据中心修复流量的最小值为 $T$. 假设存在一个满足编码参数（$n,k,d,D$）且平均跨数据中心修复流量小于 $T$ 的纠删码（不妨设为 $C_{\mathrm{O}}$），那么 $C_{\mathrm{O}}$ 的修复组分布方案 $E_{C_{\mathrm{O}}}$ 的平均跨数据中心修复流量小于 $T$. 因为所有匹配于指定编码参数（$n,k,d,D$）的纠删码修复组分布方案的平均跨数据中心修复流量的最小值为 $T$，所以 $E_{C_{\mathrm{O}}}$ 不匹配于编码参数（$n,k,d,D$），因而不存在一个满足编码参数

$(n,k,d,D)$的纠删码的修复组分布方案为$E_{C_0}$，这与$C_0$的修复组分布方案为$E_{C_0}$相矛盾. 因此，不存在一个匹配于编码参数$(n,k,d,D)$且平均跨数据中心修复流量小于$T$的纠删码，即$T$为编码参数$(n,k,d,D)$下的纠删码的平均跨数据中心修复流量下限. 证毕.

## 3 低跨云数据中心修复流量的纠删码的快速构造方法 FMEL

本节提出了一种低跨云数据中心修复流量的纠删码的快速构造方法 FMEL(如图 3 所示)，可在不同编码参数下快速求得具有低跨云数据中心修复流量的纠删码.

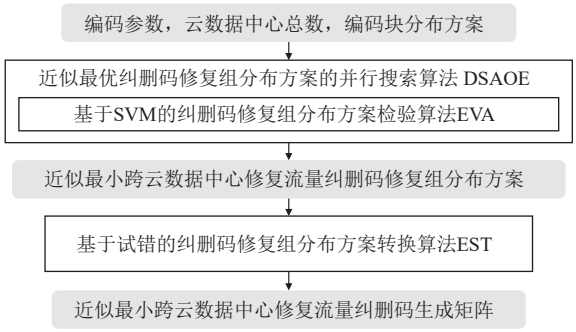


Fig. 3 The structure of FMEL

图 3 FMEL 的结构

具体而言，基于第 2 节推导出的相关定理，本节首先得出纠删码修复组分布方案匹配指定编码参数$(n,k,d,D)$的条件，并以此为依据提出了一种基于 SVM 的纠删码修复组分布方案检验算法(erasure code repair group distribution scheme verification algorithm based on SVM)EVA，可对纠删码修复组分布方案与编码参数匹配性进行快速检验.

然后，本节提出了一种具有近似最小跨云数据中心修复流量的纠删码修复组分布方案的并行搜索算法(distributed search algorithm of the erasure code repair group distribution scheme with the approximate minimum cross-cloud data center repair traffic)DSAOE，可从所有通过 EVA 检验的纠删码修复组分布方案中选出平均跨云数据中心修复流量较小的一个纠删码修复组分布方案.

最后，本节提出一种纠删码修复组分布方案转换算法(erasure code repair group distribution scheme transformation algorithm)EST，可将 DSAOE 搜索到的修复组分布方案转换为具有低跨云数据中心修复流量的纠删码的生成矩阵.

### 3.1 纠删码修复组分布方案匹配于指定($n,k,d,D$)的条件

1)纠删码修复组分布方案匹配于指定$(n,k,d,D)$的充要条件

纠删码修复组分布方案是由纠删码的编码块的修复组和位置决定. 如果将一个纠删码 Tanner 图的各个变量端点分配给不同的云数据中心，那么纠删码 Tanner 图可以同时反映相应纠删码的编码块的修复组及位置. 因此，每个纠删码 Tanner 图对应一个纠删码修复组分布方案. 因为多个纠删码 Tanner 图可能对应同一个纠删码修复组分布方案，所以每个纠删码修复组分布方案通常对应多个纠删码 Tanner 图. 纠删码 Tanner 图$\mathcal{T}$匹配于编码参数$(n,k,d,D)$是指存在一个 Tanner 图为$\mathcal{T}$的纠删码的编码参数为$(n,k,d,D)$. 因此，若一个纠删码修复组分布方案所对应的纠删码 Tanner 图中，有不少于一个纠删码 Tanner 图匹配于编码参数$(n,k,d,D)$，则该方案匹配于编码参数$(n,k,d,D)$，反之则该方案不匹配于编码参数$(n,k,d,D)$. 即纠删码修复组分布方案匹配于编码参数$(n,k,d,D)$的充要条件是至少存在一个与之对应的 Tanner 图满足定理 3 中的子条件 1~4.

2)纠删码修复组分布方案匹配于指定$(n,k,d,D)$的必要条件

纠删码修复组分布方案匹配于编码参数$(n,k,d,D)$的一个必要条件如定理 4 所示.

### 3.2 EVA 算法

基于 3.1 节得出的纠删码修复组分布方案匹配指定编码参数$(n,k,d,D)$的条件，首先，EVA 把纠删码修复组分布方案和对应的编码参数转换为便于分类算法高效处理的定长特征向量，转换过程能够保留判断纠删码修复组分布方案是否匹配于指定编码参数所需的全部信息，因而能够保证分类的准确性. 此外，EVA 使用一个 SVM 来对定长特征向量进行分类以达到快速检验纠删码修复组分布方案是否匹配于对应的编码参数的目的. 在分类过程中，EVA 可以持续收集更多带标签定长特征向量，并使用这些定长特征向量对 SVM 分类器进行增量更新，从而达到不断提高分类准确率的目的. EVA 使用 3.1 节推导出的纠删码修复组分布方案匹配于指定编码参数$(n,k,d,D)$的条件为定长特征向量打标签，以得到 SVM 分类器的初始训练数据集和增量更新数据集.

#### 3.2.1 特征向量构造

基于 SVM 的 EVA 对纠删码修复组分布方案进行转换：

首先，由定义 6 有，如果可用云数据中心的数目为 $N$，那么编码块的修复组分布方案 $C$ 一共有 $2^N-1$ 个可能的取值. 因此，EVA 将把这 $2^N-1$ 个可能的取值一一映射为 $1,2,\cdots,2^N-1$，映射函数如式（9）所示.

$$map(C)=\sum_{i\in C}2^i-1. \tag{9}$$

然后，对于每个云数据中心，EVA 将统计出纠删码修复组分布方案中对应于该云数据中心中的编码块的修复组分布方案的映射值中 $1,2,\cdots,2^N-1$ 出现的次数，从而得到了一个长度为 $N(2^N-1)$ 的定长特征向量 $\boldsymbol{X}'$，该向量中的第 $a\times b$ 个元素的值为 $c$ 表示第 $a$ 个云数据中心中有 $c$ 个编码块的修复组分布方案的映射值为 $b$. 例如，如图 4 所示，如果纠删码修复组分布方案为{{1},{1,2},{1,2},{2,3},{1,2,3},{1,2,3}}、云数据中心总数为 3、编码条带中的第 1, 2 个编码块位于第 1 个云数据中心、编码条带中的第 3, 4 个编码块位于第 2 个云数据中心、编码条带中的第 5, 6 个编码块位于第 3 个云数据中心，那么其中各个编码块修复组分布方案的映射值分别为 1,3,3,6,7,7，因而相应的 $\boldsymbol{X}'$=(1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2).

因为 $\boldsymbol{X}'$中含有各个云数据中心中的对应于不同修复组分布方案的编码块数目，所以使用 $\boldsymbol{X}'$可以恢复出各个云数据中心中的编码块的修复组分布方案的无序集合. 例如，从 $\boldsymbol{X}'$=(1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2) 的第 3×7 个元素为 2 可以推导出第 3 个云数据中心中有 2 个编码块的修复组分布方案的映射值为 7，即它们的修复组分布方案为{1,2,3}. 因此，和原始的纠删码修复组分布方案相比，$\boldsymbol{X}'$中仅仅丢失了各个编码块在各云数据中心内的顺序信息. 因为各个编码块在各云数据中心内的顺序并不影响纠删码的冗余度、容错度和容灾度，因此纠删码修复组分布方案转换过程没有丢失判断一个其是否匹配于编码参数（$n,k,d,D$）所需的有效信息.

在将纠删码修复组分布方案转换为一个数值型的定长特征向量 $\boldsymbol{X}'$后，EVA 将 $n, k, d, D$ 追加到 $\boldsymbol{X}'$即可得到用于表示纠删码修复组分布方案及其对应编码参数（$n, k, d, D$）的特征向量 $\boldsymbol{X}$，其长度恒为 $N(2^N-1)+4$.

3.2.2 基于 SVM 的特征向量分类

由于匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案之间以及不匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案之间均有一定的相似性，所以匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案对应的特征向量之间以及不匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案对应的特征向量之间存在一定的相似性. 因此，可以采用有监督学习算法[28-29]来对纠删码修复组分布方案对应的特征向量进行分类：使用一些已知匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案对应的特征向量和已知不匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案对应的特征向量作为训练集来训练一个分类器. 分类器会判断新的纠删码修复组分布方案对应的特征向量是否和已知匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案对应的特征向量相似，如果相似则判断其匹配于编码参数（$n,k,d,D$），反之则判断其不匹配于编码参数（$n,k,d,D$）. 由于 SVM 具有分类速度快以及可对线性不可分的数据进行分类等优点[30-32]，所以可以基于 SVM 对纠删码修复组分布方案对应的特征向量进行分类，以达到快速检验纠删码修复组分布方案的目的.

EVA 的工作流程如算法 1 所示.

**算法 1.** EVA 算法.

输入：编码参数（$n,k,d,D$），编码块分布方案 $R$，纠删码修复组分布方案 $E$，云数据中心数 $N$；

输出：检验结果 *result*.

① if(系统初始化)

② $TE$←随机抽取纠删码修复组分布方案的集合{$n,N,R$}；

③ for($TE$ 中的每个元素 $e$)；

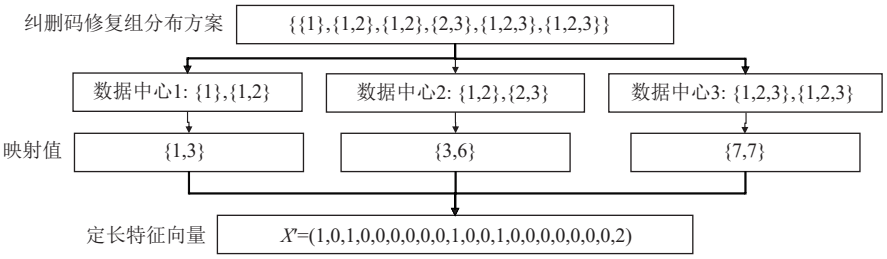


Fig. 4 Illustration of feature vectors construction

图 4 特征向量构造示意图

④ *vector*←特征向量转换(*e*);
⑤ *label*←枚举检验 Tanner 图(*n,k,d,D,R,e*);
⑥ 将〈*vector*,*label*〉加入初始训练集;
⑦ end for
⑧ end if
⑨ *svm*←使用初始训练集训练分类器;
⑩ *result.isPass*←false;
⑪ *vector* ←特征向量转换(*E,n,k,d,D,N,R*);
⑫ $p_1$←分类器分类(*svm*,*vector*);
⑬ $p_2$←判断 *E* 是否满足必要条件(*n,k,d,D,R*);
⑭ if($p_1$==true 且 $p_2$==true)
⑮ *T*←枚举纠删码 Tanner 图(*E*);
⑯ for(每一个 *T* 中的纠删码 Tanner 图 *t*)
⑰ if(*t* 满足充要条件的子条件 2(*n,k,d,D,R*))
⑱ *result.isPass*←true;
⑲ *result.value*=〈*E,t*〉;
⑳ break;
㉑ end if
㉒ if(*result.isPass*==false)
㉓ 将〈*vector*,false〉加入新训练集;
㉔ end if
㉕ 更新误分率;
㉖ end for
㉗ end if
㉘ if($p_1$==false)
㉙ *p*←0~1 间的随机数;
㉚ if(*p* 小于误分率)
㉛ *T*←枚举纠删码 Tanner 图(*E*);
㉜ for(每一个 *T* 中的纠删码 Tanner 图 *t*)
㉝ if(*t* 满足充要条件的子条件 2(*n,k,d,D,ND*))
㉞ *result.isPass*←true;
㉟ *result.value*=〈*E,t*〉;
㊱ 将〈*vector*,true〉加入新训练集;
㊲ break
㊳ end if
㊴ end for
㊵ 更新误分率;
㊶ end if
㊷ end if
㊸ if($p_1$==true 且 $p_2$==false)
㊹ 将〈*vector*,false〉加入新训练集
㊺ end if
㊻ if(新训练集大小大于阈值)
㊼ 使用新训练集更新分类器(*svm*);
㊽ end if
㊾ return *result*.

1)初始训练集获取

EVA按4步获取用于构造初始SVM分类器的训练集:

① 随机抽取部分编码参数下的部分纠删码修复组分布方案(不妨设纠删码修复组分布方案的集合为 *TE*);

② 将 *TE* 中的纠删码修复组分布方案及其对应的编码参数转换为定长特征向量;

③ 对于 *TE* 中的任意一个纠删码修复组分布方案 *E*,枚举其对应的纠删码 Tanner 图,并使用纠删码 Tanner 图匹配于指定编码参数(*n,k,d,D*)的充要条件来检验这些纠删码 Tanner 图是否存在 1 个匹配(*n,k,d,D*),从而判断 *E* 是否匹配于(*n,k,d,D*),即得到 *E* 对应定长特征向量的类别标签;

④ 将所有打上类别标签的特征向量组合为SVM分类器的初始训练集(算法 1 的行①~⑧),由于当云数据中心数 *N* 固定时,所有纠删码修复组分布方案的特征向量长度相同,所以同一套部署环境下只需要准备一份初始训练集.

2)分类器增量更新

SVM的分类器的本质是一个能将高维空间一分为二的决策超平面,而其分类特征向量的本质则是判断被投射到高位空间后的特征向量位于决策超平面的哪一侧.在SVM中,能够对决策超平面(分类器)的构造造成影响的特征向量为有效特征向量.当初始训练集中的有效特征向量的数目有限时,首次训练出的分类器很难反映真实的数据分布情况,使得其无法对特征向量进行准确分类.针对这个问题,基于SVM的EVA将在训练分类器时筛选出训练集中的有效特征向量,并在分类的同时持续收集新的已知确切类别的特征向量.然后,EVA将挑选部分新收集的特征向量与旧训练集中的有效特征向量组成新的训练集,并使用新的训练集训练出一个新的分类器(算法 1 的行 ㊻~㊽).因为新的训练集同时包含新收集的特征向量中的有效特征向量和旧训练集中有效特征向量,所以新的分类器的分类准确性更高.

增量学习的关键是如何从旧训练集中挑选出有效特征向量以及如何获取新的有效特征向量:

① 旧的训练集的有效特征向量.EVA会在训练分类器的同时获取一组支持向量,这些支持向量决定了决策超平面.也就是说,这些支持向量即为旧训练集中的有效特征向量.因此,EVA直接将这些支持

向量加入到新的训练集中.

② 新收集的特征向量中的有效特征向量. 如果分类器对一个特征向量进行了误分类，那么意味着现有的分类器（决策超平面）缺少该特征向量的信息. 因此，如果将该特征向量加入到新的训练集中，新的决策超平面将与旧的决策超平面有所不同. 所以，新收集的特征向量中被旧分类器错误分类的部分即为有效特征向量. 因此，EVA 将这些支持向量加入到新的训练集中.

具体而言，EVA 搜集新的有效特征向量以及检验纠删码修复组分布方案的具体过程如图 5 所示.

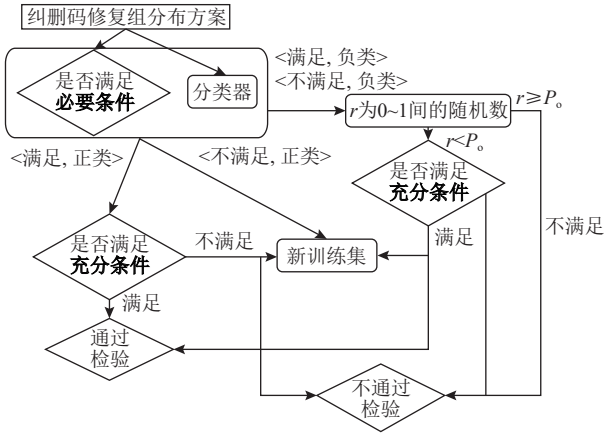


Fig. 5 Verification process of erasure code repair group distribution scheme

图 5 纠删码修复组分布方案检验过程

首先，EVA 分别用纠删码修复组分布方案匹配于指定编码参数的必要条件和分类器来检验输入的纠删码修复组分布方案.

如果分类器和纠删码修复组分布方案匹配于指定编码参数的必要条件的检验结果均为某个纠删码修复组分布方案 $E$ 匹配于编码参数 $(n,k,d,D)$，EVA 则使用纠删码 Tanner 图匹配于指定编码参数的充要条件来检验 $E$ 的纠删码 Tanner 图. 如果存在一个 $E$ 的纠删码 Tanner 图匹配于编码参数 $(n,k,d,D)$，则可确定 $E$ 匹配于编码参数 $(n,k,d,D)$. 反之，如果不存在任何一个 $E$ 的纠删码 Tanner 图匹配于编码参数 $(n,k,d,D)$，则可确定 $E$ 不匹配于编码参数 $(n,k,d,D)$ 且分类器对 $E$ 对应的的特征向量进行了错误分类，此时 $E$ 对应的特征向量将被加入新的训练集（算法 1 的行⑭~㉗）.

如果分类器的分类结果是某个纠删码修复组分布方案 $E$ 不匹配于编码参数 $(n,k,d,D)$，EVA 将在较小的概率 $P_o$ 下使用纠删码 Tanner 图匹配于指定编码参数的充要条件对 $E$ 进行检验（以检验分类器的分类结果）——因为分类器误分类频率越小，检验它的分类结果的必要性越小，所以 $P_o$ 始终等于分类器的当前误分率. 如果存在一个 $E$ 的纠删码 Tanner 图符合其匹配于指定编码参数的充要条件，可以确定 $E$ 匹配于编码参数 $(n,k,d,D)$ 且分类器对 $E$ 进行了错误分类，此时 $E$ 对应的特征向量将被加入新的训练集. 如果 EVA 没有使用纠删码 Tanner 图匹配于指定编码参数的充要条件对 $E$ 进行检验或者不存在一个对应于 $E$ 的纠删码 Tanner 图符合其匹配于指定编码参数的充要条件，那么 EVA 将 $E$ 视为不通过检验的纠删码修复组分布方案（算法 1 的行㉘~㉜）.

如果分类器的分类结果是某个纠删码修复组分布方案 $E$ 匹配于编码参数 $(n,k,d,D)$ 但 $E$ 不符合其匹配于指定编码参数的必要条件，那么可以确定 $E$ 不匹配于编码参数 $(n,k,d,D)$ 且分类器对其进行了误分类，此时 $E$ 对应的特征向量将被加入新的训练集（算法 1 的行㉝~㉟）.

### 3.3 近似最优纠删码修复组分布方案搜索算法

3.2 节提出 EVA 可快速检验纠删码修复组分布方案是否匹配于指定编码参数 $(n,k,d,D)$. DSAOE 可从所有能通过 EVA 检验的纠删码修复组分布方案中挑选出具有最小平均跨云数据中心修复流量的一个纠删码修复组分布方案.

根据定义 6，当云数据中心数目 $N$、编码参数 $n$ 以及条带分布方案确定时，相应的所有纠删码修复组分布方案均可被枚举出来. 此外，每个纠删码修复组分布方案对应一个平均跨云数据中心修复流量.

因此，DSAOE 的主要思想是：当云数据中心数 $N$, 编码块数 $n$ 和编码块分布方案被指定后，枚举出所有与它们相对应的纠删码修复组分布方案，并将这些纠删码修复组分布方案按它们对应的平均跨云数据中心修复流量递增的顺序排序. 然后，对纠删码修复组分布方案进行分组并采用多个计算节点对各组纠删码修复组分布方案进行并行检验——各个计算节点使用 EVA 对其进行检验. 各个计算节点中第一个通过 EVA 检验的纠删码修复组分布方案即为局部最小跨云数据中心修复流量纠删码修复组分布方案. 各个计算节点的局部最小跨云数据中心修复流量纠删码修复组分布方案中具有最小平均跨云数据中心修复流量的一个纠删码修复组分布方案即为 DSAOE 的输出.

DSAOE 使用 1 个协调者节点和多个工作节点来并行完成最优纠删码修复组分布方案的搜索，其具体分工为：

1)协调者节点

协调者节点的工作流程如算法 2 所示.

**算法 2.** DSAOE 的协调节点的工作流程.

输入:编码参数($n,k,d,D$),云数据中心总数 $N$,编码块分布方案 $R$, *Worker* 总数 $W$;

输出:全局最优解 $GO$(包括近似最小跨云数据中心修复流量修复组分布方案及其对应的最优纠删码 Tanner 图和平均跨云数据中心修复流量).

① 已完成工作的 Worker 数←0;
② $ES$←枚举纠删码修复组分布方案($n,k,d,D,R$);
③ *subSets*←对纠删码修复组分布方案随机分组($ES,W$);
④ 将 *subSets* 中的各个组分别分发到各个 *Worker*;
⑤ while true do
⑥ if 接收到局部最优解 $LO$
⑦ $GO$←$LO$;
⑧ 将 $GO$ 分发到各个 *Worker*;
⑨ 完成工作的 *Worker* 数←完成工作的 *Worker* 数+1;
⑩ end if
⑪ if 完成工作的 *Worker* 数==*Worker* 总数
⑫ break;
⑬ end if
⑭ end while
⑮ return $GO$.

当协调者节点接收到编码参数($n,k,d,D$)、云数据中心总数 $N$ 以及条带的编码块分布方案 $R$ 后,协调者节点将枚举所有可能的纠删码修复组分布方案并计算出这些纠删码修复组分布方案的平均跨云数据中心修复流量(算法 2 的行②)——根据定义 6,纠删码的修复组分布方案 $E$ 的平均跨云数据中心流量为 $m\sum_{C\in E}(|C|-1)/n$,($C$ 为 $E$ 中的编码块修复组分布方案、$n$ 为一个编码条带中的编码块数),所以协调者节点计算纠删码修复组分布方案的平均跨云数据中心修复流量的开销较低.

然后,协调者节点将这些纠删码修复组分布方案随机分成若干子集并分发到若干工作节点进行检验(算法 2 的行③④).工作节点检验这些子集时,协调者节点负责维护临时全局近似最小跨云数据中心修复流量纠删码修复组分布方案及其对应的临时全局近似最小平均跨云数据中心修复流量(算法 2 的行⑤~⑮).

2)工作节点

工作节点的工作流程如算法 3 所示.

**算法 3.** DSAOE 的工作节点的工作流程.

输入:编码参数($n,k,d,D$),编码块分布方案 $R$,纠删码修复组分布方案组 *subSet*,云数据中心数 $N$,分类器 *svm*;

① *subSet*←将 *subSet* 按平均跨云数据中心修复流量的升序排序;
② for 每一个 $i$,1⩽$i$⩽*subSet* 中的纠删码修复组分布方案数
③ if 接收到了新的 $GO$
④ $GO$←新的 $GO$;
⑤ end if
⑥ if *subSet*[$i$] 的平均跨云数据中心修复流量小于 $GO$ 的平均跨云数据中心修复流量
⑦ *result*←EVA($n,k,d,D,R$,*subSet*[$i$],$N$,*svm*);
⑧ if *result.isPass*==true
⑨ $LO$ 的平均跨云数据中心修复流量←*subSet*[$i$] 的平均跨云数据中心修复流量;
⑩ $LO$ 的纠删码修复组分布方案← *subSet*[$i$];
⑪ $LO$ 的纠删码 Tanner 图← *result.value.get Tanner*();
⑫ 将 $LO$ 传送到协调者;
⑬ 结束程序;
⑭ end if
⑮ end if
⑯ end for

在接收到协调者节点发来的纠删码修复组分布方案的集合后,各个工作节点首先将这些纠删码修复组分布方案按平均跨云数据中心修复流量递增的顺序进行排列(算法 3 的行①).然后,工作节点依次读取这些纠删码修复组分布方案.如果一个纠删码修复组分布方案的平均跨云数据中心修复流量小于临时全局近似最小平均跨云数据中心修复流量,那么工作节点将使用 EVA 对其进行检验.第 1 个通过 EVA 的检验的即为局部最低跨云数据中心修复流量纠删码修复组分布方案,其对应的平均跨云数据中心修复流量即为局部近似最小平均跨云数据中心修复流量.一旦一个工作节点得到了局部最低跨云数据中心修复流量纠删码修复组分布方案和局部近似最小平均跨云数据中心修复流量,它便立即停止后续纠删码修复组分布方案的检验,并分别用它的局部最低跨云数据中心修复流量纠删码修复组分布方案和局部近似最小平均跨云数据中心修复流量来更新协调者节点中的临时全局最低跨云数据中心修复

流量纠删码修复组分布方案和临时全局近似最小平均跨云数据中心修复流量（算法 3 的行②~⑭）.

当所有的工作者节点均停止检验纠删码修复组分布方案时，协调者节点中的临时全局近似最小跨云数据中心修复流量纠删码修复组分布方案和临时全局近似最小平均跨云数据中心修复流量即为 DSAOE 得到的近似最小跨云数据中心修复流量纠删码修复组分布方案和平均跨云数据中心修复流量近似下限.

由于在 EVA 的分类器将一个纠删码修复组分布方案分为正类后，EVA 还会使用纠删码 Tanner 图匹配于指定编码参数的充要条件对其进行检验，因此 DSAOE 得到的全局最优纠删码修复组分布方案一定匹配于编码参数（$n,k,d,D$）.

如果 EVA 的误报率为 0, DAOSE 可获得所有通过 EVA 检验的纠删码修复组分布方案中平均跨数据中心修复流量最小的一个. 根据定理 5, 该纠删码修复组分布方案能达到相应编码参数下的平均跨数据中心修复流量下限，即该纠删码修复组分布方案为最优纠删码修复组分布方案. 若 EVA 的误报率 $P>0$, 且一共存在 $Q$ 个最优纠删码修复组分布方案，那么 DAOSE 错过所有最优纠删码修复组分布方案的概率不超过 $P^Q$. 因此，FMEL 可以得到最优码的概率不低于 $1-P^Q$.

此外，因为 DSAOE 使用 EVA 来对大多数纠删码修复组分布方案进行快速检验，所以其效率较高. 另外，DSAOE 的并行度高的特点可进一步保证其搜索效率.

### 3.4 基于试错的纠删码修复组分布方案转换算法

3.3 节提出了 DSAOE 能搜索到近似最小跨云数据中心修复流量纠删码修复组分布方案. 本节提出了一种基于试错的纠删码修复组分布方案转换算法 EST, 用于将近似最小跨云数据中心修复流量纠删码修复组分布方案转换为相应的纠删码生成矩阵.

EST 的工作流程如算法 4 所示.

**算法 4.** EST 算法.

输入：全局最优解 $GO$（包括近似最小跨云数据中心修复流量纠删码修复组分布方案及其对应的最优纠删码 Tanner 图和平均跨数据中心修复流量）；

输出：近似最小跨云数据中心修复流量纠删码的生成矩阵 $\boldsymbol{G}$.

① $\boldsymbol{OT}\leftarrow GO.Tanner$;

② $\boldsymbol{H}\leftarrow$构造准柯西矩阵 $\boldsymbol{OT}$;

③ $SM\leftarrow$获得子矩阵集合 $SM_1$ 和 $SM_2$ 的集合$\{\boldsymbol{OT},\boldsymbol{H}\}$;

④ $ST\leftarrow$获得子图（$SM,\boldsymbol{OT}$）;

⑤ $\boldsymbol{H}_1\leftarrow$获得子矩阵 $\boldsymbol{H}_1$（$\boldsymbol{OT},\boldsymbol{H}$）;

⑥ 将 $\boldsymbol{H}_1$ 添加到集合 $LS$ 中;

⑦ for 每一个 $i$, $1\leqslant i\leqslant ST$ 中子图数

⑧ $maxM\leftarrow$获得 $ST_i$ 的最大匹配;

⑨ $\boldsymbol{Z}\leftarrow$获得对应于最大匹配的子矩阵（$SM_i,ST_i$）;

⑩ 将 $\boldsymbol{Z}$ 添加到集合 $LS$ 中;

⑪ end for

⑫ $NE\leftarrow$获得对角元素（转换为上三角矩阵（$LS$））;

⑬ for（每一个 $i$, $1\leqslant i\leqslant NE$中的元素数）

⑭ if $NE[i]$ 的值==0

⑮ 更新 $\boldsymbol{H}$ 和 $LS$;

⑯ 跳转到行⑩;

⑰ end if

⑱ end for

⑲ $\boldsymbol{G}\leftarrow$获取 $\boldsymbol{H}$ 对应的生成矩阵;

⑳ return $\boldsymbol{G}$.

1）对于指定的纠删码修复组分布方案，DSAOE 只有在找到一个与其相对应的匹配于编码参数（$n,k,d,D$）的纠删码 Tanner 图时才会确认该纠删码修复组分布方案匹配于编码参数（$n,k,d,D$）. 因此，DSAOE 在搜索近似最小跨云数据中心修复流量纠删码修复组分布方案的同时也得到了与之相对应的近似最小跨云数据中心修复流量纠删码 Tanner 图 $\mathcal{T}$, 如算法 4 的行①所示.

2）令 $\boldsymbol{U}$ 为如式（10）所示的柯西矩阵，基于试错的纠删码修复组分布方案转换算法 EST 将构造一个对应于$\mathcal{T}$的校验矩阵 $\boldsymbol{H}$: 如果$\mathcal{T}$的第 $j$ 个校验端点覆盖其第 $i$ 个变量端点，那么 $\boldsymbol{H}$ 的第 $j$ 行 $i$ 列$\left(h_{ji}\right)$的值等于 $\boldsymbol{U}$ 的第 $j$ 行 $i$ 列$\left(u_{ji}\right)$的值. 如果$\mathcal{T}$的第 $j$ 个校验端点不覆盖其第 $i$ 个变量端点，那么$h_{ji}$的值为 0，如算法 4 的行②所示.

$$\boldsymbol{U}_{(n-k)\times n}=\begin{pmatrix}\frac{1}{1+1} & \cdots & \frac{1}{1+n}\\ \vdots & & \vdots\\ \frac{1}{n-k+1} & \cdots & \frac{1}{n-k+n}\end{pmatrix}. \quad (10)$$

3）EST 获取 $\boldsymbol{H}$ 的所有包含 $n-k$ 行 $d-1$ 列的子矩阵的集合 $SM_1$ 以及 $\boldsymbol{H}$ 的所有包含其对应于任意 $D$ 个云数据中心的列的子矩阵的集合 $SM_2$. 根据定义 7, $SM_1$ 和 $SM_2$ 中的任意矩阵均对应于一个$\mathcal{T}$的子图，如算法 4 的行③④所示.

4）EST 获取$\mathcal{T}$的最大匹配. 因为$\mathcal{T}$符合其匹配于指定编码参数的充要条件的子条件 3, 其最大匹配中包含 $n-k$ 个变量端点，不妨设$\mathcal{T}$的最大匹配包含$VN_{z_1}$,

$VN_{z_2},\cdots,VN_{z_{n-k}}$. 随后, EST 求得由 $\boldsymbol{H}$ 的第 $l_1,l_2,\cdots,l_{n-k}$ 列组成的子矩阵 $\boldsymbol{H}_1$. 显然, $\boldsymbol{H}_1$ 是一个方阵 ($n-k$ 行、$n-k$ 列). 同时, EST 将 $\boldsymbol{H}_1$ 加入到集合 $LS$ 中, 如算法 4 的行⑤⑥所示.

5) 基于试错的纠删码修复组分布方案转换算法 EST 获取对应于 $SM_1$ 和 $SM_2$ 中各个矩阵 $\mathcal{T}$ 的各个子图的最大匹配. 对于 $SM_1$ 和 $SM_2$ 中的各个矩阵 $\boldsymbol{Z}$, 如果它对应 $\mathcal{T}$ 的子图的最大匹配中包含的所有校验端点为 $CN_{l_1},CN_{l_2},\cdots,CN_{l_a}$, 则将 $\boldsymbol{Z}$ 的第 $l_1,l_2,\cdots,l_a$ 行组成的子矩阵添加到集合 $LS$ 中. 根据定理 3 中的纠删码 Tanner 图匹配于指定编码参数的充要条件的充分性证明的步骤③, $LS$ 中的矩阵均为方阵, 如算法 4 的行⑦~⑩所示.

6) EST 通过初等行变换将 $LS$ 中的各个矩阵转换为如式 (11) 所示的上三角矩阵 (转换过程见定理 3 中的纠删码 Tanner 图匹配于指定编码参数的充要条件的充分性证明的步骤④), 并获得这些上三角矩阵的对角元素的集合 $NE=\left\{h_{l_{t,a}s_{t,a}}-A_{l_{t,a}s_{t,a}}\middle|t\in[1,|LS|];a\in\{1,2,\cdots,b_t\}\right\}$, 其中, $A_{z_{t,a}s_{t,a}}$ 为不包含 $h_{l_{t,a}s_{t,a}}$ 的线性多项式, 如算法 4 的行⑫所示.

$$\begin{pmatrix} h_{l_{t,1}s_{t,1}}-A_{l_{t,1}s_{t,1}} & \cdots & \cdots & \cdots \\ 0 & h_{l_{t,2}s_{t,2}}-A_{l_{t,2}s_{t,2}} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & h_{l_{t,b_t}s_{t,b_t}}-A_{l_{t,b_t}s_{t,b_t}} \end{pmatrix}. \tag{11}$$

7) EST 考察 $NE$ 中的各个对角元素是否为 0. 如果 $NE$ 中某个元素的值为 0 (不妨设 $h_{l_{o,q}s_{o,q}}-A_{l_{o,q}s_{o,q}}=0$), 则进行操作: 对 $h_{l_{o,q}s_{o,q}}$ 进行加 1 操作; 更新 $\boldsymbol{H}$, $LS$; 重新构造上三角矩阵并更新集合 $NE$; 重新考察 $NE$ 中各个元素的值. 直到 $NE$ 中的各个元素的值均不为 0, 如算法 4 的行⑬~⑱所示, 根据定理 3 中的纠删码 Tanner 图匹配于指定编码参数的充要条件的证明过程, 此时 $LS$ 中的矩阵均满秩. 其中, $LS$ 中的 $\boldsymbol{H}_1$ 满秩意味着 $\boldsymbol{H}$ 满秩. 此外, 由定理 2 可知, $LS$ 中的其他矩阵满秩意味着此时的 $\boldsymbol{H}$ 对应的纠删码 $C_{\mathrm{O}}$ 能够容忍任意 $d-1$ 个编码块失效或任意 $D$ 个云数据中心失效 (即 $\boldsymbol{H}$ 为一个匹配于 $(n,k,d,D)$ 的校验矩阵). 事实上, 在我们的多次实验中, 初始 $NE$ 中的各个元素的值全不为 0. 也就是说, 在大多数情况下, 不需要对 $NE$、$LS$ 和 $\boldsymbol{H}$ 进行更新即可一次性得到最终的校验矩阵 $\boldsymbol{H}$.

8) EST 对 $\boldsymbol{H}$ 执行初等行变换以得到矩 $\boldsymbol{H}'=\boldsymbol{H}\boldsymbol{\Delta}^{-1}=\left[\boldsymbol{I}_{(n-k)\times(n-k)},\boldsymbol{P}_{(n-k)\times k}\right]$ ($\boldsymbol{H}'\boldsymbol{\Delta}=\boldsymbol{H}$). 然后, EST 构造 $\boldsymbol{G}'=[-(\boldsymbol{P}^{\mathrm{T}})_{k\times(n-k)},\boldsymbol{I}_{k\times k}]$ 和 $\boldsymbol{G}=\boldsymbol{G}'\cdot\left(\boldsymbol{\Delta}^{\mathrm{T}}\right)^{-1}$. 显然, $\boldsymbol{G}'(\boldsymbol{H}')^{\mathrm{T}}=\boldsymbol{0}$. 因此 $\boldsymbol{G}\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{G}'\cdot(\boldsymbol{\Delta}^{\mathrm{T}})^{-1}\boldsymbol{\Delta}^{\mathrm{T}}(\boldsymbol{H}')^{\mathrm{T}}=\boldsymbol{G}'\boldsymbol{I}_{n\times n}(\boldsymbol{H}')^{\mathrm{T}}=\boldsymbol{0}$. 因此, $\boldsymbol{G}$ 为对应于 $\boldsymbol{H}$ 的生成矩阵, 即对应于近似最小跨云数据中心修复流量纠删码修复组分布方案的生成矩阵, 如算法 4 的行⑲所示.

### 3.5 FMEL 方法描述

低跨云数据中心修复流量的纠删码的快速构造方法 FMEL 的工作流程如算法 5 所示.

**算法 5.** 算法 FMEL.

输入: 编码参数 $(n,k,d,D)$, 云数据中心总数 $N$, 编码块分布方案 $R$, *Worker* 总数 $W$, 云数据中心数 $N$;

输出: $\boldsymbol{G}$.

① 全局最优解 $GO\leftarrow DSAOE\,(n,k,d,D,N,R,W,N)$;

② $\boldsymbol{G}\leftarrow EST\,(GO)$;

③ return $\boldsymbol{G}$.

首先, FMEL 通过 DSAOE 从所有能通过基于 SVM 的 EVA 的检验的纠删码修复组分布方案中选出具有较小平均跨云数据中心修复流量的一个纠删码修复组分布方案, 如算法 5 的行①所示. 挑选出的纠删码修复组分布方案即为近似最小跨云数据中心修复流量纠删码修复组分布方案. 然后, FMEL 通过 EST 将近似最小跨云数据中心修复流量纠删码修复组分布方案转换为相应编码参数下的近似最小跨云数据中心修复流量纠删码生成矩阵 $\boldsymbol{G}$, 如算法 5 的行②所示.

## 4 实验与结果

### 4.1 方法实现

为了评估 FMEL 构造出的近似最小跨云数据中心修复流量纠删码的实际性能, 我们基于 OpenEC[33] 和 FastDFS[34] 实现了 FMEL 构造出的纠删码. 其中, OpenEC 是一种用于在已有的分布式文件系统中实现不同的纠删码编码方法和修复方法的可定制框架, FastDFS 是一种轻量级的文件系统.

具体而言, 我们首先在原始的 FastDFS 上增加了对 OpenEC 的支持. 然后, 我们在 OpenEC 上实现了 FMEL, 用于构造近似最小跨云数据中心修复流量纠删码生成矩阵. 最后, 基于 OpenEC 的编码方法和修复方法定制功能, 实现了近似最小跨云数据中心修复流量纠删码的数据编码方法和数据修复方法.

### 4.2 实验环境与参数

实验部署于 UCloud[35] 的 6 个云数据中心, 其中 2 个位于北京 (记为 PEK1 和 PEK2)、1 个位于上海 (记为 SHA)、1 个位于洛杉矶 (记为 LOS)、1 个位于

台北（记为 TPE）、1 个位于广州（记为 CAN）. 实验使用了每个云数据中心的 10 个存储节点（云主机），每个节点配备 4 核 Intel Cascadelake 6248R 3.0 GHz 处理器, 8 GB 内存和 1 TB 磁盘, 外网最大带宽为 800 Mbps, 内网最大带宽为 25 Gbps.

为了评估 FMEL 构造出的近似最小跨云数据中心修复流量纠删码的性能，我们将其与典型的纠删码进行了比较：最优码构造方法 ACIoT 构造出的最优码（因为原始的 ACIoT 只能构造出指定编码参数（$n,k,d$）下的最优码，我们对其进行了扩展，使其能够构造出指定编码参数（$n,k,d,D$）下的最优码）、一种能够达到平均局部性下限的纠删码 ACL 码[23]、经典的 RS 码[36]、一种典型的局部性码 Xorbas 码[22] 和一种典型的跨云数据中心纠删码 DFC 码[13].

为了评估 FMEL 构造出的近似最小跨云数据中心修复流量纠删码在不同环境中的性能，我们将 UCloud 的 6 个云数据中心分为了 2 个实验环境. 由于大多数的跨云数据中心存储系统均是部署在 3 个云数据中心[1,14]，所以我们的每个实验环境均包含 3 个云数据中心. 具体而言，实验环境 1 包含 PEK1, PEK2, SHA, 实验环境 2 包含 LOS, TPE, CAN. 由于各个实验环境包含 3 个云数据中心，因而容灾度 $D$ 的合理取值范围为 [1,2], 所以实验中 $D\in[1,2]$. 此外，在实际应用中，为了便于条带管理，单个条带内的编码块数 $n$ 通常较小. 因此，我们在实验中将 $n$ 的范围限制在[6, 11]（常见的生产系统中的 $n$ 均处于该范围内[37-39]）. 另外，实验中单个条带内的数据块数 $k$ 的取值范围为 $[n/3, 2n/3]$, 因为当 $k$ 属于此范围时 $D$ 的取值范围为 [1,2]. 最后，容错度 $d$ 的取值范围为 $[2, n-k+1]$, 即 $d$ 的合理取值范围[1,12-18].

在本文实验中，各个编码条带的编码块被平均分配到各个云数据中心中，以获取较高的容灾度和负载均衡性. 此外，新生节点与其相应的失效编码块位于同一云数据中心，以保持系统的容灾度和负载均衡性. 实验中的编码块大小和 HDFS 一致[35]，为 128 MB.

### 4.3 评价指标

我们用 4 个指标来评价 FMEL 的性能：

1）分类器误报率（false-negative rate, FN）. 假设被 FMEL 中的 SVM 分类器误分类为负类（不满足编参数（$n,k,d,D$）的类）的纠删码修复组分布方案的数目为 $f_1$, EVA 检验过的纠删码修复组分布方案中匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案数为 $f_2$, 那么分类器误报率为 $f_1/f_2$.

2）纠删码构建时间（construction time, CT）. 纠删码构造时间是指 ACIoT 构造最优码的时间或 FMEL 构造近似最小跨云数据中心修复流量纠删码的时间.

3）平均跨云数据中心修复流量 $\overline{T}$. 令某纠删码修复它的一个条带中的 $n$ 个编码块产生的跨云数据中心修复流量分别为 $T_1, T_2, \cdots, T_n$ 且每个编码块的大小为 $m$, 那么该纠删码的平均跨云数据中心修复流量 $\overline{T}=\sum_i T_i/mn$.

4）平均修复用时 $\bar{t}$. 令某纠删码在实验环境 1 中修复它的一个条带中的 $n$ 个编码块所消耗的时间分别为 $t_{1,1}, t_{1,2}, \cdots, t_{1,n}$, 在实验环境 2 中修复它的一个条带中的 $n$ 个编码块所消耗的时间分别为 $t_{2,1}$, $t_{2,2}, \cdots, t_{2,n}$, 每个编码块的大小为 $m$. 因为 $t_{j,i}$ 受到云数据中心的排列顺序的影响，令 $\bar{t}_{j,i}$ 为 $t_{j,i}$ 在不同云数据中心的排列顺序下的平均值. 那么，该纠删码的平均修复用时 $\bar{t}=\left(\sum_i \bar{t}_{1,i}+\sum_i \bar{t}_{2,i}\right)/mn$.

### 4.4 分类器误报率

在 FMEL 中，SVM 分类器将一个纠删码修复组分布方案分为正类后，还会使用纠删码 Tanner 图匹配于指定编码参数的充要条件对其进行检验，因此 FMEL 不会将不匹配于编码参数（$n,k,d,D$）纠删码修复组分布方案错误分为正类. 此外，如果 SVM 分类器将一个匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案错误分为负类，FMEL 可能会错过具有较小平均跨云数据中心修复流量且匹配于编码参数（$n,k,d,D$）的纠删码修复组分布方案. 因此，我们主要关注分类器将纠删码修复组分布方案错误分为负类的概率，即误报率.

我们的测试数据包括所有编码参数下的所有纠删码修复组分布方案. 在分类器初始化时，首先在各组编码参数下随机挑选了 50 个纠删码修复组分布方案并将这些纠删码修复组分布方案转换为对应的特征向量. 然后，使用纠删码修复组分布方案匹配于编码参数（$n,k,d,D$）的充要条件判断这些纠删码修复组分布方案是否满足各自的编码参数，并以此为依据为对应的特征向量打上标签. 因此，我们可以得到这些纠删码修复组分布方案对应的带标签的特征向量，它们组成了分类器的初始训练集（10 次实验的平均初始训练集构造用时为 1 711 s、平均初始分类器构造用时为 192 s）. 然后，我们按编码参数 $n$ 递增的顺序将其他的纠删码修复组分布方案输入到分类器中进行分类. 与此同时，FMEL 不断搜集新的训练集对分类器进行增量更新.

分类器分类每组（$n,k$）对应的所有纠删码修复

组分布方案的误报率如图 6 所示. 因为在分类过程中含有效信息的特征向量被逐渐加入到新的训练集中，并不断更新分类器，因此分类器的误报率随着 $n$ 的增加而逐渐降低.

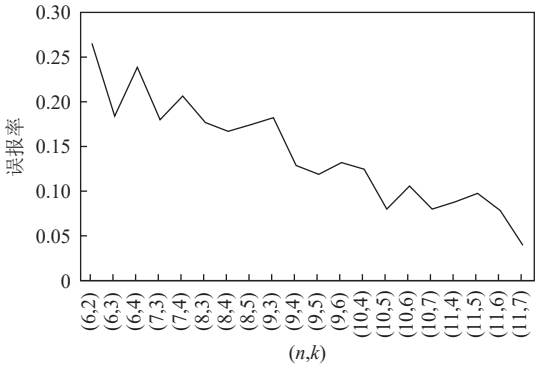


Fig. 6 False-negative rate of SVM's classifier

图 6 SVM 分类器的误报率

因为同一组编码参数下的最优码往往存在多个，所以最优纠删码修复组分布方案也往往存在多个. 假设一共存在 $Q$ 个最优纠删码修复组分布方案且分类器的误报率为 $P$，那么 FMEL 错过所有最优纠删码修复组分布方案的概率不超过 $P^Q$. 因此，FMEL 可以得到最优码的概率不低于 $1-P^Q$. 由图 6 可知，$P$ 总是小于 27% 的. 所以，如果 $Q$>1，FMEL 有 92.7% 的概率得到最优码；如果 $Q$>2，FMEL 则有 98% 的概率得到最优码.

### 4.5 纠删码构造时间

每组 $(n,k)$ 对应多组 $(n,k,d,D)$，FMEL 和 ACIoT 在每组 $(n,k)$ 对应的所有 $(n,k,d,D)$ 下的纠删码构造时间如图 7 所示（FMEL 的工作节点数和 ACIoT 的工作进程数均为 5）. 在构造最优码或近似最小跨云数据中心修复流量纠删码时，ACIoT 和 FMEL 均需要枚举和检验部分纠删码修复组分布方案的纠删码 Tanner 图. 由于编码参数 $(n,k,d,D)$ 下的纠删码 Tanner 图的总数为 $2^{n(n-k)}$，所以，通常而言，$n(n-k)$越大，ACIoT 和 FMEL 需要检验的纠删码 Tanner 图越多. 因此，ACIoT 和 FMEL 的纠删码构造时间均呈现出随着 $n(n-k)$的减少而减少的趋势.

对于大部分的纠删码修复组分布方案，FMEL 仅需要用 SVM 分类器对它们分类即可，无需枚举和检验它们对应的纠删码 Tanner 图. 因此，FMEL 的平均纠删码构造时间仅为 ACIoT 的 11%. 特别地，当 $n(n-k)$ 较小时，总的纠删码构造时间较短. 所以，此时 FMEL 中更新分类器的时间占总的纠删码构造时间的比例较大. 因此，此时 FMEL 的纠删码构造时间比 ACIoT 略长.

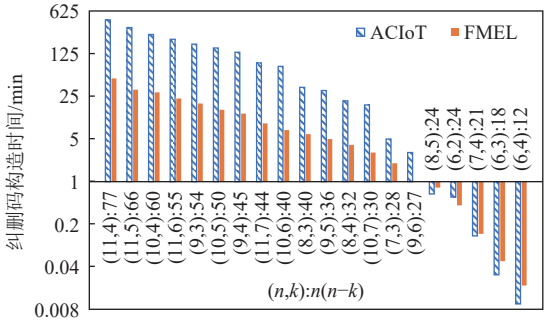


Fig. 7 Erasure code construction time of ACIoT and FMEL

图 7 ACIoT 和 FMEL 的纠删码构造时间

### 4.6 平均跨云数据中心修复流量

因为不同的纠删码适应的编码参数不同，我们首先在除 RS 码外的所有纠删码都适应的编码参数下比较了各个纠删码的平均跨云数据中心修复流量（RS 码使用和其他纠删码相近的编码参数），如图 8 所示. 在这些编码参数下，DFC 码和 ACIoT 均可为每个云数据中心分配一个局部校验块，因此，DFC 和 ACIoT 均能够在不跨云数据中心传输数据的情况下完成编码块的修复，因而它们的平均跨云数据中心修复流量为 0. 大多数情况下，FMEL 的平均跨云数据中心修复流量也为 0，即 FMEL 有较大概率得到最优码.

因为 RS 码在修复任意一个编码块时均需要读取 $k$ 个编码块，而在实验中使用的全部编码参数下各编码条带在同一个云数据中心的编码块数始终小于 $k$（如果要使各编码条带在同一个云数据中心的编码块数始终不小于 $k$，那么冗余度将十分大），所以 RS 码在修复任意一个编码块时均需要跨云数据中心传输数据. 因此，RS 码跨云数据中心修复流量最大.

因为 Xorbas 码和 ACL 码具有较低的修复流量（其中 ACL 码能够达到平均修复流量的下限），所以它们能在一定程度上降低平均跨云数据中心修复流量，进而使得它们的平均跨云数据中心修复流量相较于局部性为 $k$–1 的 RS 码更小. 但是，由于修复流量不与跨云数据中心修复流量严格正相关，所以 ACL 码和 Xorbas 码的平均跨云数据中心修复流量大于 ACIoT，FMEL，DFC 码.

因为 RS 码和 DFC 码对编码参数的限制较为严格，我们在更多的编码参数下比较了其他纠删码. 如图 9 所示，在这些编码参数下，FMEL 的平均跨云数据中心修复流量比 ACL 码和 Xorbas 码分别低了 24.0% 和 34.8%，这是因为 FMEL 能在这些参数下达到平均跨云数据中心修复流量的近似下限.

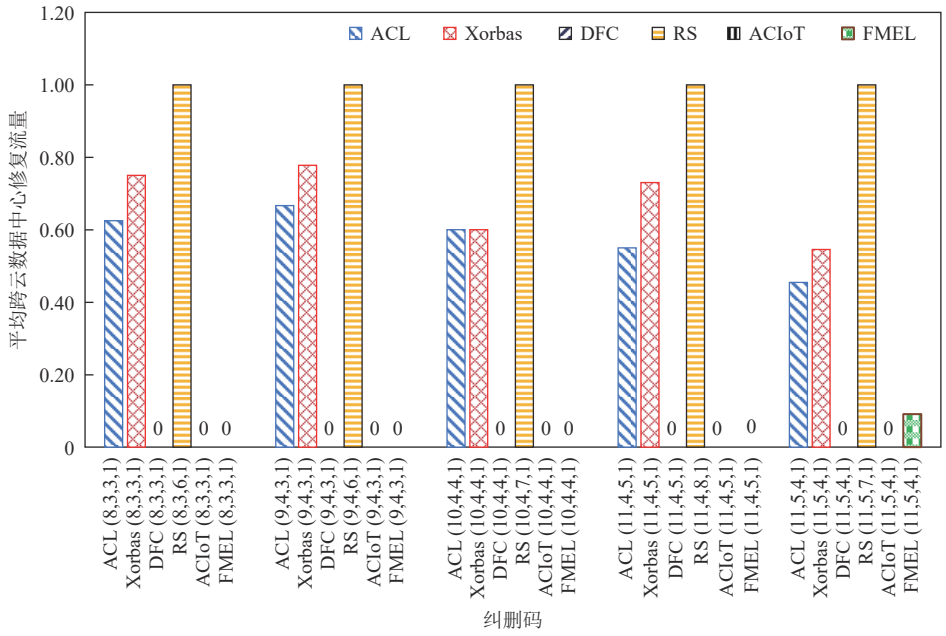


Fig. 8 Average cross-cloud data center repair traffic of ACL, Xorbas, DFC, RS, ACIoT and FMEL

图 8 ACL 码、Xorbas 码、DFC 码、RS 码、ACIoT 和 FMEL 的平均跨云数据中心修复流量

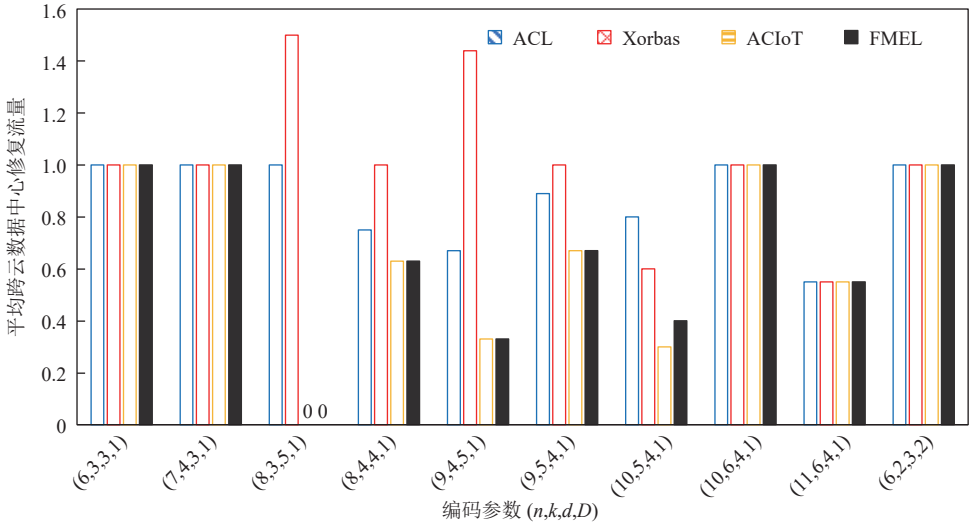


Fig. 9 Average cross-cloud data center repair traffic of ACL, Xorbas, ACIoT and FMEL

图 9 ACL 码、Xorbas 码、ACIoT 和 FMEL 的平均跨云数据中心修复流量

平均而言，FMEL 的平均跨云数据中心修复流量比 ACL 码、Xorbas 码和 RS 码分别低了 42.9%, 51.1%, 56.0%. 此外，FMEL 的平均跨云数据中心修复流量与 DFC 码相近，但 DFC 码对编码参数限制严格.

另外，FMEL 的平均跨云数据中心修复流量是 ACIoT 的 100%~133%，并且在实验采用的 15 组编码参数中的 13 组编码参数下，FMEL 的平均跨云数据中心修复流量与 ACIoT 相同，即 FMEL 构造出最优码的概率较高.

容错度 $d$ 和 FMEL 构造的近似最小跨云数据中心修复流量纠删码的平均跨云数据中心修复流量 $\overline{L}$ 之间的关系如图 10 所示，当 $d$ 小于一个特定值 $d'$后，近似最小跨云数据中心修复流量纠删码的平均跨云数据中心修复流量 $\overline{L}$,即平均跨云数据中心修复流程近似下限. 这是因为当 $d<d'$时，$\overline{L}$的主要影响因素为容灾度 $D$. 基于这一发现，不仅能够得到编码参数 $(n,k,D)$ 下的平均跨云数据中心修复流量的近似下限，还可以得到满足该近似下限时的 $d$ 的上限，即图中的最优 $d$.

因为与最优 $d$ 相对应的编码参数和近似最小跨云数据中心修复流量纠删码能够在冗余度 $(\overline{T})$、容错度 $(d)$、容灾度 $(D)$ 和平均跨云数据中心修复流量

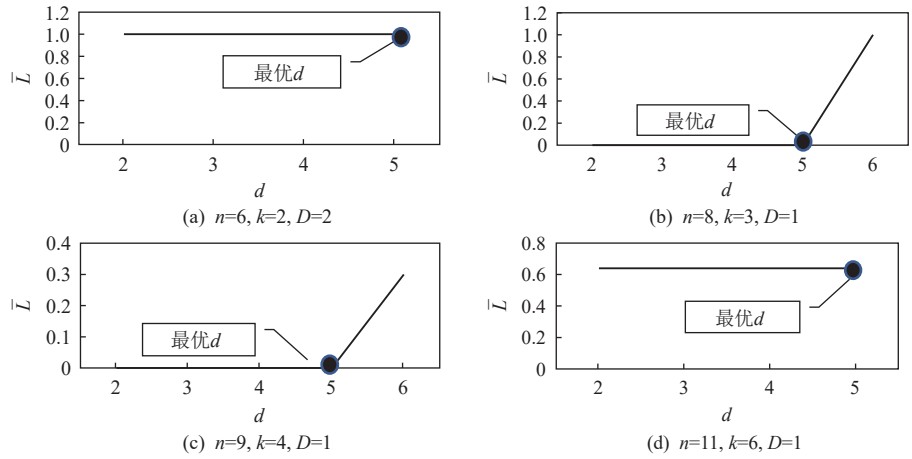


Fig. 10 Relationship between $d$ and average cross-cloud data center repair traffic's approximate lower bound under different $n$, $k$, $D$

图 10 不同 $n$, $k$, $D$ 下 $d$ 与平均跨云数据中心修复流量近似下限的关系

($\overline{L}$) 之间取得较好权衡，因此关于最优 $d$ 的这一发现对于实际应用具有指导意义. 比如说，基于这一发现，我们找到了 2 个能够在冗余度、容错度、容灾度和平均跨数据修复流量之间取得较好权衡的纠删码——FMEL（$n$=6, $k$=2）和 FMEL（$n$=9, $k$=5, 其生成矩阵如式（12）和式（13）所示.

$$\begin{pmatrix} 1 & 0 & -4/5 & -2/3 & -1/2 & -2/7 \\ 0 & 1 & -8/9 & -3/4 & -4/7 & -1/3 \end{pmatrix}, \tag{12}$$

$$\begin{pmatrix} 1 & 0 & -9/11 & 0 & 0 & 0 & -1/2 & 0 & 1/4 \\ 0 & 1 & -9/10 & 0 & 0 & 0 & 0 & -1/2 & 1/3 \\ 0 & 0 & 0 & 1 & 0 & 0 & -2/3 & 0 & 1/21 \\ 0 & 0 & 0 & 0 & 1 & 0 & -3/4 & -2/3 & 35/72 \\ 0 & 0 & 0 & 0 & 0 & 1 & -6/7 & -3/4 & 37/70 \end{pmatrix}. \tag{13}$$

如表 2 所示，FMEL（$n$=6, $k$=2）的 $d$ 值比 3 副本技术高 66.7%，同时，FMEL（$n$=6, $k$=2）的 $D$、$\overline{T}$和$\overline{L}$和 3 副本技术相同. 此外，FMEL（$n$=9, $k$=5）的 $D$ 比 2 副本技术高 33.3%，同时 FMEL（$n$=9, $k$=5）的$\overline{T}$和$\overline{L}$比 2 副本技术低 10% 和 33% 且二者 $D$ 相同.

### 4.7 平均修复用时

因为不同的纠删码适应的编码参数不同，我们首先在除 RS 码外的所有纠删码都适应的编码参数下比较了各个纠删码的平均修复用时（RS 码使用和其他纠删码相近的编码参数），如图 11 所示.

由于 FMEL 和 ACIoT 的平均跨云数据中心修复流量小于对照组中的其他纠删码，其修复数据时跨云数据中心传输数据的时间较短，使得其平均修复用时也比其他纠删码少. 具体地，FMEL 的平均修复用时比 ACL 码、Xorbas 码和 RS 码分别少了 36.9%, 46.1%, 50.6%. 此外，ACIoT 的平均修复用时与 FMEL 相近，但是 ACIoT 的纠删码构造时间更长.

**Table 2 Comparison Between FMEL and Replications**

**表 2 FMEL 与多副本的对比**

| 评估指标 | FMEL（$n$=6, $k$=2） | 3 副本 | FMEL（$n$=9, $k$=5） | 2 副本 |
|---|---|---|---|---|
| $\overline{T}$ | 1 | 1 | 0.67 | 1 |
| $\overline{t}$ /（ms·MB$^{-1}$） | 86.6 | 83.2 | 62.1 | 82.1 |
| $d$ | 5 | 3 | 1 | 1 |
| $D$ | 2 | 2 | 1 | 1 |
| $n/k$ | 3 | 3 | 1.8 | 2 |

因为 RS 码和 DFC 码的编码参数适应性较差，我们在更多的编码参数下对除这 2 种纠删码之外的纠删码进行了对比，如图 12 所示. 在这些编码参数下，FMEL 和 ACIoT 的平均修复用时少于 ACL 码和 Xorbas 码，这是因为基于 FMEL 和 ACIoT 的平均跨云数据中心修复流量较小. 特别地，在编码参数为（8,3,5,1）时，FMEL 和 ACIoT 的平均修复用时远低于其他纠删码，因为此时只有它们能够在不跨云数据中心传输数据的情况下完成数据修复.

此外，如表 2 所示，因为 FMEL（$n$=9, $k$=5）的平均跨云数据中心修复流量低于 2 副本技术，因此其平均修复用时也短于 2 副本技术. 特别地，虽然 FMEL（$n$=6, $k$=2）的平均跨云数据中心修复流量与 3 副本技术相同，但其平均修复用时略长于 3 副本技术，这是因为 FMEL（$n$=6, $k$=2）在修复数据时的计算量大于 3 副本技术.

## 5 总 结

针对现有纠删码构造方法无法兼顾编码参数适

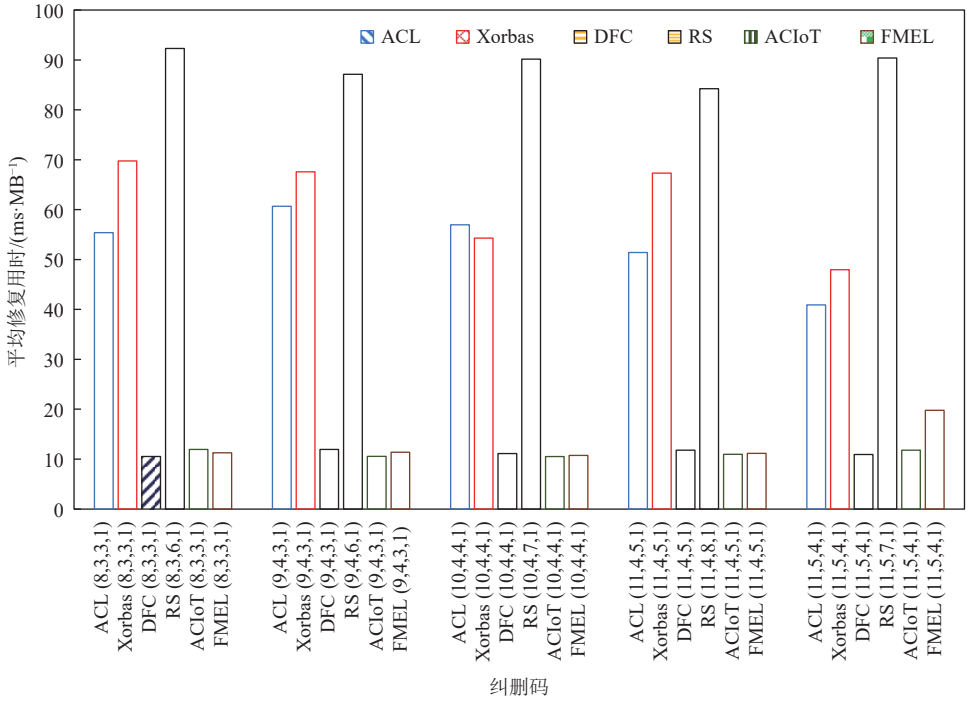


Fig. 11 Average repair time of ACL, Xorbas, DFC, RS, ACIoT and FMEL

图 11 ACL 码、Xorbas 码、DFC 码、RS 码、ACIoT 和 FMEL 的平均修复用时

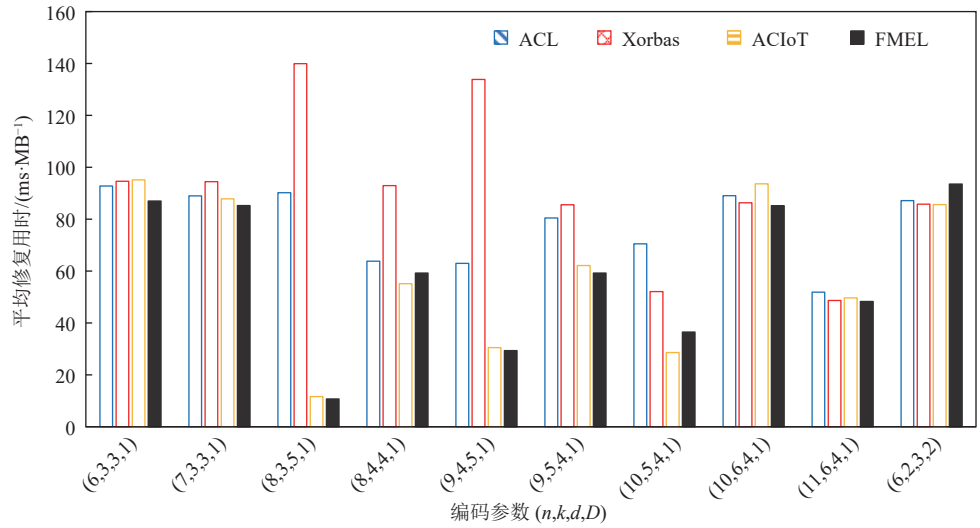


Fig. 12 Average repair time of ACL, Xorbas, ACIoT and FMEL

图 12 ACL 码、Xorbas 码、ACIoT 和 FMEL 的平均修复用时

应性、跨云数据中心修复流量和纠删码构造效率的问题，本文提出了一种低跨云数据中心修复流量的纠删码的快速构造方法 FMEL，可在不同编码参数下快速求得低跨云数据中心修复流量纠删码. 在真实跨云数据中心环境中的实验表明，相较于现有的可构造能达到平均跨云数据中心修复流量下限的最优码的方法，FMEL 构造纠删码的时间少了 89%，且在大部分编码参数下二者构造的纠删码的平均跨云数据中心修复流量相同. 此外，我们利用 FMEL 评估了大量编码参数，并挑选出 2 组编码参数. FMEL 在这 2 组编码参数下构造的纠删码的平均修复流量低于已得到广泛使用的多副本技术，同时其容灾性、容错性和冗余度相较于多副本技术具有明显优势.

**作者贡献声明：**包涵提出了算法思路和实验方案，并负责完成实验和撰写论文；王意洁提出指导意见并修改论文.

## 参 考 文 献


[1] Cheng Yuxia, Yu Xinjie, Chen Wenzhi, et al. A practical cross-datacenter fault-tolerance algorithm in the cloud storage system[J]. Cluster Computing, 2017, 20(2): 1801−1813

[2] SOHU. Amazon AWS confirms the downtime in night [EB/OL]. (2019-06-24)[2019-08-11]. http://www.sohu.com/a/322769512_115060 (in Chinese)
(搜狐. 亚马逊AWS证实晚间宕机[EB/OL]. (2019-06-24)[2019-08-11]. http://www.sohu.com/a/322769512_115060)

[3] SOHU. Unexpected power outage in AWS data center causes over 1TBofdataloss[EB/OL].(2019-09-05)[2021-09-24].https://www.sohu.com/a/338998898_468733 (in Chinese)
(搜狐. AWS 数据中心再出断电事故，丢失数据超过1TB[EB/OL]. (2019-09-05)[2021-09-24]. https://www.sohu.com/a/338998898_468733)

[4] Sina Technology. Cable smashing affects Alipay [EB/OL]. (2015-05-27)[2019-08-11]. http://tech.sina.com.cn/i/2015-05-27//doc-iavxeafs8200893.shtml (in Chinese)
(新浪科技. 光缆挖断影响支付宝[EB/OL]. (2015-05-27)[2019-08-11]. http://tech.sina.com.cn/i/2015-05-27//doc-iavxeafs8200893.shtml)

[5] SOHU. Japan earthquake threatens data centers of several IT giants in Tokyo [EB/OL]. (2019-06-03)[2019-08-11]. http://it.sohu.com/20110311/n279778961.shtml (in Chinese)
(搜狐. 日本地震危及数家IT巨头设在东京的数据中心[EB/OL]. (2019-06-03)[2019-08-11]. http://it.sohu.com/20110311/n279778961.shtml)

[6] Kejixun. Official response to large-scale failure of Amazon China cloud service: Affected by the construction party to cut fiber [EB/OL]. (2019-06-03)[2019-08-11]. http://www.kejixun.com/article/190603/464156.shtml (in Chinese)
(科技迅. 官方回应亚马逊中国云服务大规模故障[EB/OL]. (2019-06-03)[2019-08-11]. http://www.kejixun.com/article/190603/464156.shtml)

[7] Wang Huaimin, Shi Peichang, Zhang Yiyan. JointCloud: A cross-cloud cooperation architecture for integrated Internet service customization[C]// Proc of the 37th IEEE Int Conf on Distributed Computing Systems (ICDCS). Piscataway, NJ: IEEE, 2017: 1846−1855

[8] Zhang Yuchao, Nie Xiaohui, Jiang Junchen, et al. BDS+: An inter-datacenter data replication system with dynamic bandwidth separation[J]. IEEE/ACM Transactions on Networking, 2021, 29(2): 918−934

[9] Zhou Tianli, Tian Chao. Fast erasure coding for data storage[J]. ACM Transactions on Storage, 2020, 16(1): 1−24

[10] Wang Yijie, Li Sikun. Research and performance evaluation of data replication technology in distributed storage systems[J]. International Journal of Computer and Mathematics with Applications, 2006, 51(11): 1625−1632

[11] Wang Yijie, Xu Fangliang, Pei Xiaoqiang. Research on erasure code-based fault-tolerant technology for distributed storage[J]. Chinese Journal of Computers, 2017, 40(1): 236−255 (in Chinese)
(王意洁，许方亮，裴晓强. 分布式存储中的纠删码容错技术研究[J]. 计算机学报, 2017, 40(1): 236−255)

[12] Wang Yijie, Pei Xiaoqiang, Ma Xingkong, et al. TA-Update: An adaptive update scheme with tree-structured transmission in erasure-coded storage systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2017, 29(8): 1893−1906

[13] Yu Xinjie. Cloud storage system with cross datacenters fault tolerance [D]. Hangzhou: Zhejiang University, 2016 (in Chinese)
(俞新杰. 跨数据中心容错的云存储系统[D]. 杭州：浙江大学，2016)

[14] Caneleo P, Mohan L, Parampalli U, et al. On improving recovery performance in erasure code based geo-diverse storage clusters [C]//Proc of the 12th Int Conf on the Design of Reliable Communication Networks. Piscataway, NJ: IEEE, 2016: 123−129

[15] Chen H, Hu Yuchong, Lee P, et al. NCCloud: A network-coding-based storage system in a cloud-of-clouds[J]. IEEE Transactions on Computers, 2013, 63(1): 31−44

[16] Hu Yuchong, Chen H, Lee P, et al. NCCloud: Applying network coding for the storage repair in a cloud-of-clouds [C]//Proc of the 10th USENIX Conf on File and Storage Technologies. Berkeley, CA: USENIX Association, 2012: 21

[17] Hu Yuchong, Lee P, Zhang Xiaoyang. Double regenerating codes for hierarchical data centers [C]//Proc of the IEEE Int Symp on Information Theory (ISIT). Piscataway, NJ: IEEE, 2016: 245-249

[18] Xie Xin, Wu Chentao, Gu Junqing, et al. AZ-Code: An efficient availability zone level erasure code to provide high fault tolerance in cloud storage systems [C]//Proc of the 35th Symp on Mass Storage Systems and Technologies (MSST). Piscataway, NJ: IEEE, 2019: 230−243

[19] Bao Han, Wang Yijie, Xu Fangliang. An adaptive erasure code for JointCloud storage of Internet of things big data[J]. IEEE Internet of Things Journal, 2020, 7(3): 1613−1624

[20] AWS. Cloud storage on AWS[EB/OL]. (2021-09-24)[2021-09-24].https://aws.amazon.com/cn/products/storage/ (in Chinese)
(亚马逊. AWS上的云存储[EB/OL]. (2021-09-24)[2021-09-24].https://aws.amazon.com/cn/products/storage/)

[21] Huang Cheng, Simitci H, Xu Yikang, et al. Erasure coding in windows Azure storage [C]//Proc of the USENIX Annual Technical Conf. Berkeley, CA: USENIX Association, 2012: 2

[22] Sathiamoorthy M, Asteris M, Papailiopoulos D, et al. XORing elephants: Novel erasure codes for big data[J]. VLDB Endowment, 2013, 6(3): 325−336

[23] Shahabinejad M, Khabbazian M, Ardakani M. On the average locality of locally repairable codes[J]. IEEE Transactions on Communications, 2017, 66(7): 2773−2783

[24] Saeed S. Sandooq: Improving the communication cost and service latency for a multi-user erasure-coded geo-distributed cloud

environment [D]. Urbana-Champaign: University of Illinois at Urbana-Champaign, 2016

[25] Xu Fangliang, Wang Yijie, Ma Xingkong. Incremental encoding for erasure-coded cross-datacenters cloud storage[J]. Future Generation Computer Systems, 2018, 87: 527−537

[26] Bao Han, Wang Yijie, Xu Fangliang. A cross-datacenter erasure code writing method based on generator matrix transformation[J]. Journal of Computer Research and Development, 2020, 57(2): 291−305 (in Chinese)
(包涵, 王意洁, 许方亮. 基于生成矩阵变换的跨数据中心纠删码写入方法[J]. 计算机研究与发展, 2020, 57(2): 291−305)

[27] Murashka V. A generalization of Hall's theorem on hypercenter [EB/OL]. (2021-08-16)[2022-07-25].https://arxiv.org/abs/2103.04900v2

[28] Wang Yijie, Li Xiaoyong, Li Xiaoling, et al. A survey of queries over uncertain data[J]. Knowledge & Information Systems, 2013, 37(3): 485−530

[29] Wang Yijie, Ma Xingkong. A general scalable and elastic content-based publish/subscribe service[J]. IEEE Transactions on Parallel & Distributed Systems, 2015, 26(8): 2100−2113

[30] Wang Zhenya, Yao Ligang, Cai Yongwu, et al. Mahalanobis semi-supervised mapping and beetle antennae search based support vector machine for wind turbine rolling bearings fault diagnosis[J]. Renewable Energy, 2020, 155: 1312−1327

[31] Shankar K, Lakshmanaprabu S, Gupta D, et al. Optimal feature-based multi-kernel SVM approach for thyroid disease classification[J]. The Journal of Supercomputing, 2020, 76(28): 1−16

[32] Sherki P, Vala V. A class-incremental classification method based on support vector machine[C/OL]// Proc of the 14th IEEE Int Conf on Semantic Computing (ICSC). Piscataway, NJ: IEEE, 2020: 31−36

[33] Li Xiaolu, Li Runhui, Lee P, et al. OpenEC: Toward unified and configurable erasure coding management in distributed storage systems [C]//Proc of the 17th USENIX Conf on File and Storage Technologies. Berkeley, CA: USENIX Association, 2019: 331−344

[34] Liu Tao, Wu Shaocheng, Li Jin, et al. Blockchain-based trusted sharing of electric energy privacy data[C]// Proc of the Int Conf on Cyberspace Innovation of Advanced Technologies. New York: ACM, 2020: 556−564

[35] UCloud. UCloud's official website [EB/OL]. (2021-09-24)[2021-09-24].https://www.ucloud.cn (in chinese)
(优刻得. 优刻得官网[EB/OL]. (2021-09-24)[2021-09-24].https://www.ucloud.cn)

[36] Gao Zhen, Zhang Lingling, Cheng Yinghao, et al. Design of FPGA-implemented Reed-Solomon erasure code decoders with fault detection and location on user memory[J]. IEEE Transactions on Very Large Scale Integration Systems, 2021, 29(6): 1073−1082

[37] Apache. Apache Hadoop 3.0. 0 [EB/OL]. (2021-09-24)[2021-09-24]. http://hadoop.apache.org/docs/r3.0.0/

[38] Andrew F. Storage architecture and challenges at Google faculty summit 2010[EB/OL]. (2010-06-29)[2019-08-11]. https://www.systutoriaLS.com/3306/storage-architecture-and-challenges/

[39] Samal S. Yahoocos [EB/OL]. (2015-02-03)[2019-08-11]. https://yahooeng.tumblr.com/post/116391291701/yahoo-cloud-object-store-object-storage-at

**Bao Han**, born in 1992. PhD. His main research interests include cloud storage and erasure coding.
**包　涵**, 1992 年生. 博士. 主要研究方向为云存储、纠删码.

**Wang Yijie**, born in 1971. PhD, professor, PhD supervisor. Distinguished member of CCF. Her main research interests include distributed storage, big data analysis, and cloud computing.
**王意洁**, 1971 年生. 博士, 教授, 博士生导师. CCF 杰出会员. 主要研究方向为分布式存储、大数据分析、云计算.