DOI: 10.13382/j.jemi.2017.09.020

# 分布式存储系统中基于 Pyramid 码的局部性修复编码*

王 静[1] 张 崇[1] 梁 伟[2] 刘向阳[3]

(1. 长安大学信息工程学院 西安 710064;2. 湖南科技大学计算机科学与工程学院 湘潭 411201;
3. 西安通信学院 西安 710106)

**摘 要**:为了提高分布式存储系统的存储可靠性以及故障节点的修复效率,提出一种基于 Pyramid 码的局部性修复编码方案。该编码方案采用 Pyramid 码的最小可实现编码结构,划分局部修复组,确保较低的修复局部性并实现故障节点的快速修复。性能分析表明,基于 Pyramid 码的局部性修复编码可实现存储系统中多个故障节点的快速修复,具有较低的修复局部性,且相对于三副本复制策略以及简单再生码,基于 Pyramid 码的局部性修复编码在存储开销和修复带宽开销方面的性能更优。

**关键词**: 分布式存储系统;Pyramid 码;再生码;局部性修复编码

**中图分类号**: TP911.2 文献标识码: A **国家标准学科分类代码**: 520.5030

# Locally repairable codes based on Pyramid codes in distributed storage systems

Wang Jing[1] Zhang Chong[1] Liang Wei[2] Liu Xiangyang[3]

(1. School of Information Engineering, Chang'an University, Xi'an 710064, China;
2. School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China;
3. Xi'an Communication College, Xi'an 710106, China)

**Abstract**: In order to improve the reliability of distributed storage systems and repair efficiency of failed nodes, locally repairable codes based on Pyramid codes are proposed in this paper. Specifically, adopting the minimum achievable encoding structure of Pyramid codes, the proposed locally repairable codes divide the nodes of distributed storage systems into multiple local repair groups, to achieve lower repair locality and rapid repair of failed nodes. The performance analysis shows that, the proposed locally repairable codes based on Pyramid codes can achieve rapid repair of multiple failed nodes in distributed storage systems, has lower repair locality. Moreover, compared with three copy replication strategies and simple regenerating codes, the proposed locally repairable codes based on Pyramid codes have advantages in the performances of storage overhead and repair bandwidth overhead.

**Keywords**: distributed storage systems; Pyramid codes; regenerating codes; locally repairable codes

## 0 引 言

当前信息数据呈现出爆炸性增长,分布式存储系统因其廉价性和高扩展性等特点,得到了广泛的应用,很大程度上缓解了存储压力。为了确保数据存储的可靠性和可用性,目前很多分布式存储系统依然采取复制策略来保证存储的可靠性[1],比如 Hadoop 分布式文件系统(distributed file system, HDFS)[2],OpenStack 平台的 SWIFT[3]和微软的 Azure 服务平台[4]等等,复制策略需要存储大量副本数据以确保系统较高的可靠性,存储代价过高。

纠删码能够有效地提高存储效率,其中的最大距离可分(maximum distance separable, MDS)码,在确保系统

收稿日期:2017-05 Received Date: 2017-05

* 基金项目:国家自然科学基金(61640006, 61572188)、陕西省自然科学基金(2016JQ6011)西安市科技计划项目(2017088CG/RC051(CADX002))、中央高校基本科研业务费专项资金(310850160317)资助项目

可靠性的同时,能有效提高存储效率,已广泛应用到分布式存储系统中[5]。然而,纠删码在修复故障节点时需要下载整个文件大小的数据量,与复制策略相比,修复带宽开销过大。针对复制策略和纠删码存在的局限性,Dimakis 等人[6]提出了再生码的概念,能显著降低分布式存储系统中故障节点的修复带宽开销。通过对带宽-存储开销进行分析,Dimakis 等人[7]进一步提出了最小存储再生(minimum storage regenerating, MSR)码和最小带宽再生(minimum bandwidth regenerating, MBR)码。

再生码在故障节点修复时,只关注修复带宽开销和存储开销,没有考虑磁盘 I/O 开销,磁盘 I/O 开销是分布式存储系统数据修复的又一性能瓶颈[8]。磁盘 I/O 开销与故障节点修复过程中连接的存活节点数目成正比,修复过程中连接的存活节点数量越少,磁盘 I/O 开销越少。局部性修复编码(locally repairable codes, LRC)通过减少修复过程中连接的节点数,进一步减少修复过程中的磁盘 I/O 开销,具有较好的修复局部性[9-10],但却无法获得如 MSR 码以及 MBR 码的较低修复带宽开销。Papailiopoulos 等人[11]将可容多错的 RS 纠删码与简单的异或运算相结合,构造了一类简单局部性修复编码——简单再生码,通过连接少量节点快速修复单节点故障。

分布式存储系统中两个或多个节点同时故障的概率很大,已有的局部性修复编码在修复一个节点故障时具有很好的修复局部性[12-13],但是在修复两个或多个故障节点时需要连接多个存活节点,磁盘 I/O 开销较高。为此,本文考虑将 Pyramid 码[14]和局部性修复编码结合,提出一种新的编码方式——基于 Pyramid 码的局部性修复编码。采用 Pyramid 码的最小可实现编码结构,划分局部修复组,确保较低的修复局部性并实现故障节点的快速修复。理论分析表明,基于 Pyramid 码的局部性修复编码可实现存储系统中多个故障节点同时快速修复,具有较低的修复局部性,且相对于三副本复制策略以及简单再生码,基于 Pyramid 码的局部性修复编码在存储开销和修复带宽开销方面的性能更优。

## 1 再生码和局部性修复编码

### 1.1 再生码

在具有 $n$ 个节点的分布式存储系统中,需要存储大小为 B 的文件,每个节点存储的数据量为 $\alpha$。采用再生码对分布式存储系统中的一个故障节点进行修复,需要连接任意 $d \geqslant k$ 个存活节点,并从每个节点下载 $\beta \leqslant \alpha$ 大小的数据量,修复带宽开销 $\gamma = d\beta$。再生码中的参数 $n$、$k$、$d$、$\alpha$、$\beta$ 必须满足:

$$B \leqslant \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\} \tag{1}$$

当式(1)中等号成立时,此时的再生码为最优编码。根据式(1),可以得到存储-带宽开销曲线[6-7],以及 MSR 码对应的 MSR 点,满足:

$$\alpha_{\mathrm{MSR}} = \frac{B}{k},\ \gamma_{\mathrm{MSR}} = \frac{Bd}{k(d-k+1)} \tag{2}$$

MBR 码对应的 MBR 点,满足:

$$\alpha_{\mathrm{MBR}} = \frac{2Bd}{2kd-k^2+k},\gamma_{\mathrm{MBR}} = \frac{2Bd}{2kd-k^2+k} \tag{3}$$

### 1.2 局部性修复编码

随着需要存储的数据量地增加,分布式存储系统中节点故障概率增大,因此提高分布式存储系统可靠性的关键在于提高故障节点的修复效率。故障节点修复效率由修复过程中参与修复的节点数决定,为此,Gopalan 等人[15]于 2012 年将局部性修复的概念首次引入分布式存储系统,同时给出了 $(r,d_{\min})$ 编码最小距离的上界:

$$d_{\min} \leqslant n-k-\left\lceil \frac{k}{r} \right\rceil + 2 \tag{4}$$

**定义 1**[15] 若分布式存储系统所采用编码的最小距离为 $d_{\min}$,修复系统中的任何一个故障节点只需要连接 $r$ 个存活节点,称这个分布式存储系统所采用的编码具有局部性 $r$,并将这种形式的编码称为 $(r,d_{\min})$ 编码。

对于一个采用 $(r,d_{\min})$ 编码的分布式存储系统,如果系统中存在一个节点发生故障,需要连接该故障节点所在局部修复组内的 $r$ 个数据实现故障节点修复,如果是两个或多个节点同时故障则需要连接 $k$ 个节点数据信息进行修复,这种故障节点修复方式称为局部性修复。

## 2 基于 Pyramid 码的局部性修复编码

为了确保存储系统故障节点修复时获得较低磁盘 I/O 开销的同时,实现节点存储开销和修复带宽开销之间的均衡,考虑将 Pyramid 码引入局部性修复编码中。Pyramid 码在编码过程中,将所有数据块进行局部编码生成局部校验块,即冗余编码块全部是由原文件中部分数据块编码生成的局部校验块,将存储数据块的节点记为数据节点,存储校验块的节点记为校验节点。

如图 1 所示,Pyramid 码的最小编码结构中,包含 4 个数据块 $d1$、$d2$、$d3$、$d4$,存储在 $2\times2$ 的二维阵列节点中。对该二维阵列节点中每行每列分别进行局部编码生成 4 个局部校验块 $c1$、$c2$、$c3$、$c4$。采用 Pyramid 码,故障节点可以通过所在行或者列对应的局部校验块进行修复,具有更好的鲁棒性。

用矩阵表示图 1 中的 Pyramid 码。假设原始数据块 $\boldsymbol{D} = [d_1 \quad d_2 \quad d_3 \quad d_4]$,进行 Pyramid 码编码后的编码块记为 $\boldsymbol{C} = [d_1 \quad d_2 \quad d_3 \quad d_4 \quad c_1 \quad c_2 \quad c_3 \quad c_4]$,则 $\boldsymbol{C} = \boldsymbol{D}\cdot\boldsymbol{G}$,生成矩阵:

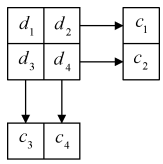



图 1　Pyramid 码最小编码结构

Fig. 1　The minimum encoding structure of Pyramid codes

$$\boldsymbol{G} = [\boldsymbol{I} \mid \boldsymbol{P}] = \begin{bmatrix} 1 & 0 & 0 & 0 & g_{11} & 0 & g_{13} & 0 \\ 0 & 1 & 0 & 0 & g_{21} & g_{22} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & g_{33} & g_{34} \\ 0 & 0 & 0 & 1 & 0 & g_{42} & 0 & g_{44} \end{bmatrix}$$

式中:变量 $g_{ij}$ 表示 Pyramid 码的编码系数。由生成矩阵可知局部校验块 $c_1 = g_{11}d_1 + g_{21}d_2$ , $c_2 = g_{22}d_2 + g_{42}d_4$ , $c_3 = g_{13}d_1 + g_{33}d_3$ , $c_4 = g_{34}d_3 + g_{43}d_4$。

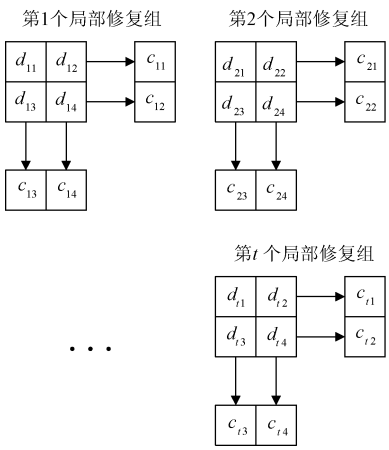



图 2　基于 Pyramid 码的局部性修复编码结构 1

Fig. 2　The first locally repairable codes structure based on Pyramid codes

根据 Pyramid 码编码原理,为实现故障节点的快速修复,首先对分布式存储系统中的存储节点进行分组,以最小修复组 8 个节点为标准进行分组。假设在分布式存储系统中存在 $n$ 个存储节点,存在以下两种情况。

1）存储节点数 $n$ 可以被 8 整除时,将整个分布式存储系统分为 $t = \frac{n}{8}$ 个局部修复组,在每个局部修复组内采用 Pyramid 码的最小编码结构。图 2 所示为对应的基于 Pyramid 码的局部性修复编码结构,将原文件分为 $4t$ 个数据块,分别存储在 $t$ 个局部修复组内,在每个局部修复组内通过生成矩阵 $\boldsymbol{G}$ 分别生成 4 个局部校验块。

根据图 1 可知,Pyramid 码中如果数据节点发生故障,则可以通过连接该故障节点所在的行或者列中的其它两个节点进行快速修复;如果是校验节点发生故障,只需连接该故障节点所在行(或者列)的两个数据节点实现快速修复。无论是数据节点还是校验节点发生故障,都可以实现局部性为 2 的快速修复。

2）存储节点数 $n$ 不能够被 8 整除时,假设 $n$ 除以 8 的整数部分为 $t$ ,余数部分为 $\lambda$ ( $\lambda$ 取整数且 $1 \leqslant \lambda \leqslant 7$ ),在局部修复组内分别采用 Pyramid 码和 MDS 码进行编码,图 3 所示为对应的编码结构。将原文件分为 $4t + \varepsilon$ 个数据块分别存储在 $t$ 个局部修复组内,前 $t - 1$ 个局部修复组中每个局部修复组分别存储 4 个数据块并采用 Pyramid 码的最小编码结构进行 Pyramid 编码。第 $t$ 个局部修复组中存储 $\varepsilon + 4$ 个数据块,并对其采用 $(\lambda + 8, \varepsilon + 4)$ MDS 码进行编码。

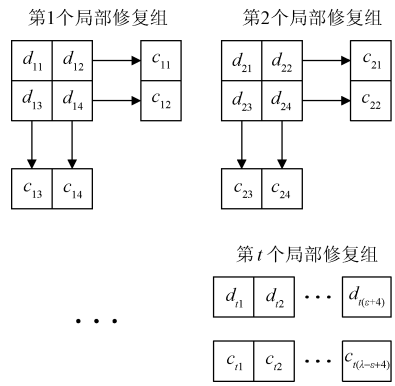



图 3　基于 Pyramid 码的局部性修复编码结构 2

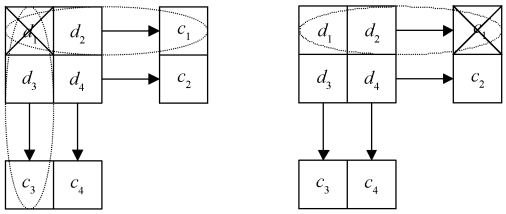
Fig. 3　The second locally repairable codes structure based on Pyramid codes

具体地,$t$ 个局部修复组中每个局部修复组具有 8 个存储节点,其中前 $t - 1$ 个局部修复组采用 Pyramid 码的最小编码结构进行编码,第 $t$ 个局部修复组采用 $(\lambda + 8, \varepsilon + 4)$ MDS 码编码。

## 3　故障节点快速修复算法

考虑到分布式存储系统被分成 $t$ 个局部修复组,多个故障节点将分别处于不同的局部修复组内,且每个局部修复组内 3 个及多个节点同时发生故障的概率很低,这里只考虑局部修复组内一个节点故障或者两个节点同时发生故障的情况。首先,讨论局部修复组内一个节点发生故障的情况,图 4(a)所示是存储数据块 $d1$ 的数据节点发生故障,可以通过该故障节点所在行的存活节点中的数据块 $d2$ 和局部校验块 $c1$ 进行修复,也可以选择该故障节点所在的列存活节点中的数据块 $d3$ 和局部校验块 $c3$ 来修复。存储数据块 $d2$ 、$d3$ 、$d4$ 的数据节点故障时也可以通过类似的方法实现快速修复。图 4(b)所示为存储校验块 $c1$ 的校验节点发生故障,通过所在行存活节点中的数据块 $d1$ 和 $d2$ 实现快速修复。

对于局部修复组内两个节点同时发生故障,存在以



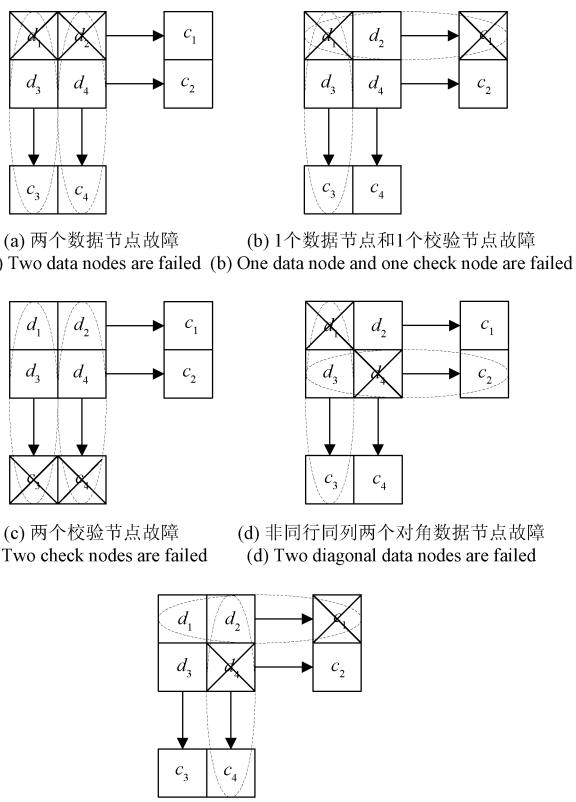
(a) 故障节点是数据节点
(a) The fault node is the data node

(b) 故障节点是校验节点
(b) The fault node is the check node

图 4　局部修复组内一个故障节点修复

Fig. 4　Repair of a failed node in one local repair group

下 4 种情况:1) 两个数据节点同时故障,根据 Pyramid 码在节点修复过程中的独立性,只需分别连接故障节点所在行(或者列)对应的存活节点进行修复,图 5(a)所示为对存储数据块 $d1$ 和 $d2$ 的数据节点故障进行修复;2) 1 个数据节点和 1 个校验节点同时故障,则只需连接 3 个存活节点即可,如图 5(b)所示, $d1$ 和 $c1$ 同时发生故障,首先需要连接 $d3$ 和 $c3$ 来修复 $d1$ ,然后利用存活节点 $d1$ 和 $d2$ 对校验节点 $c1$ 进行修复;3) 两个同行(或者同列)校验节点同时故障,则需要连接对应行(或者列)的 4 个数据节点进行修复,如图 5(c)所示,存储 $c3$ 和 $c4$ 的两个校验节点同时故障,需要连接 $d1$ 、$d2$ 、$d3$ 和 $d4$ 即可实现故障节点修复;4) 若不是同行(或者同列)的两个存储节点同时发生故障,则只需要连接局部修复组内 3 个存活节点即可,如图 5(d)和(e)所示,图 5(d)中存储数据块 $d1$ 和 $d4$ 的两个数据节点发生故障,则只需要连接存储 $d3$ 、$c2$ 和 $c3$ 的 3 个存活节点即可修复,图 5(e)中存储 $c1$ 的校验节点和存储 $d4$ 的数据节点同时发生故障,只需连接存储 $d1$ 、$d2$ 和 $c4$ 的 3 个存活节点即可修复。



(a) 两个数据节点故障
(a) Two data nodes are failed

(b) 1个数据节点和1个校验节点故障
(b) One data node and one check node are failed

(c) 两个校验节点故障
(c) Two check nodes are failed

(d) 非同行同列两个对角数据节点故障
(d) Two diagonal data nodes are failed

(e) 非同行同列中一个数据节点和一个校验节点故障
(e) One data node and one check node not in the same line and the same column are failed

图 5　局部修复组内两个故障节点修复

Fig. 5　Repair of two failed nodes in one local repair group

如果故障节点位于 MDS 码所在局部修复组内,则根据 MDS 码故障节点修复的性质,修复单个或者两个故障节点时,只需连接局部修复组内 $\varepsilon + 4$ 个存活节点,并从每个存活节点下载 $\beta = B/(4t + \varepsilon)$ 大小的数据量来进行修复。进一步地, $(\lambda + 8,\ \varepsilon + 4)$ MDS 码所在局部修复组最多可以同时修复 $\lfloor(\lambda - \varepsilon + 4)/2\rfloor$ 个故障节点。

## 4　性能分析

首先分析基于 Pyramid 码的局部性修复编码的存储开销,并与最常用的三副本复制方式以及最典型的局部性修复编码——简单再生码[11]进行对比;进一步,理论分析基于 Pyramid 码的局部性修复编码的修复带宽开销以及修复局部性,并与三副本复制方式以及局部性修复编码进行对比。

### 4.1　存储开销

假设原文件大小为 $B$ ,采用三副本复制策略,存储一位有效数据的代价为 3,则节点存储开销为 $\alpha = 3B/k$ 。在 $(n,k,f)$ 简单再生码中,每个局部修复组内采用 $(n,k)$ 删余码,这里 $f$ 表示原文件 $B$ 所分的子文件数目。在 $(n,k,f)$ 简单再生码中,当原文件大小为 $B$ 时,需要将原文件平均分成 $f$ 个子文件,并且对这 $f$ 个大小为 $B/f$ 子文件分别进行 $(n,k)$ RS 编码,并且将具有相同下标的编码块进行简单的异或运算生成 $n$ 个校验块,并以下标循环的方式依次将生成的编码块存储在这 $n$ 个节点中。可以看出,采用 $(n,k,f)$ 简单再生码,每个节点存储 $f+1$ 个编码块。且从前面分析已经得到,大小为 $B/f$ 的子文件将存储在 $k$ 个存储节点中,且通过 $(n,k)$ RS 编码,将生成的冗余数据存储在另外 $n - k$ 个节点上。则每个存储节点上存储的编码块大小等于子文件的大小 $B/f$ 除以 $k$ ,即 $B/kf$ 。经过 $(n,k,f)$ 简单再生码,每个节点存储 $f + 1$ 个编码块,则此时每个节点存储开销就是 $(f + 1)B/fk$ 。取 $f = 3$ ,则节点存储开销为 $\alpha = 4B/3k$ 。

在基于 Pyramid 码的局部性修复编码方案中,数据节点数目:

$$k = \begin{cases} 4t, & n\%8 = 0 \\ 4t + \varepsilon, & n\%8 \neq 0 \end{cases} \tag{5}$$

根据原文件大小为 $B$ 可得两种情况下节点存储开销均为 $\alpha = B/k$ 。

假设文件大小 $B = 1\ 000$ Mb ,图 6 所示为节点存储开销随着数据块存储节点 $k$ 的变化曲线。由图 6 可知,三副本复制策略、简单再生码以及基于 Pyramid 码的局部性修复编码的节点存储开销都随着数据节点个数 $k$ 的增加而逐渐减小,且本文提出的基于 Pyramid 码的局部性修复编码具有最小的节点存储开销。当分布式存储系统中存在 $k = 30$ 个数据节点时,基于 Pyramid 码的局部性修复编码的节点存储开销为 33.33 Mb,远小于三副本复制策略 100 Mb 的节点存储开销,同时也小于简单再生码 44.44 Mb 的节点存储开销。

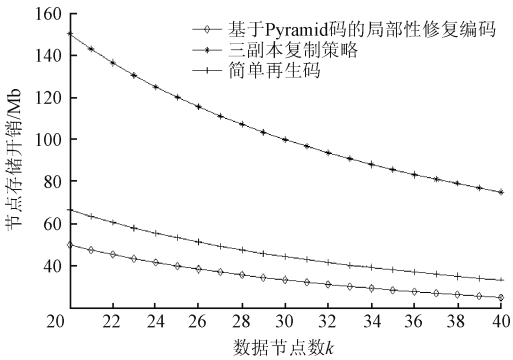



图 6　节点存储开销与数据节点个数之间的关系

Fig. 6　Relationship of storage overhead vs. the number of data nodes

### 4.2　带宽开销和修复局部性

为了更好地分析基于 Pyramid 码的局部性修复编码的修复带宽开销和修复局部性,考虑到节点故障的概率以及 MDS 编码在故障节点修复时的性质,假设所有故障节点都位于 Pyramid 码所在局部修复组内。

**表 1　几种编码方案的故障节点修复性能比较**

**Table 1　Comparison of the performances of several encoding schemes in repairing failed nodes**

| | | 三副本复制策略 | 简单再生码 | 基于 Pyramid 码的局部性修复编码 |
|---|---|---|---|---|
| 节点存储开销 | | $3B/k$ | $(f+1)B/fk$ | $B/k$ |
| 修复带宽开销 | 单节点故障 | $3B/k$ | $(f+1)B/k$ | $2B/k$ |
| | 两节点故障 | $6B/k$ | $B$ | $3B/k$ 或 $4B/k$ |
| 修复局部性 | 单节点故障 | 1 | $f$ | 2 |
| | 两节点故障 | 2 | $k$ | 3 或 4 |

表 1 给出了分布式存储系统采用三副本复制策略、局部性修复编码中的简单再生码以及基于 Pyramid 码的局部性修复编码时的节点存储开销、修复带宽开销和修复局部性。考虑到修复带宽开销等于修复局部性乘以节点存储开销,文件大小取 $B = 1\ 000$ Mb 固定。在单节点故障时,三副本复制策略只需要访问存储了故障节点数据的单一存活节点,修复局部性为 1,修复带宽开销为 $3B/k$ ;简单再生码需要访问 $f$ 个存活节点,节点存储开销为 $(f+1)B/fk$ ,则修复带宽开销为 $\alpha = (f+1)B/k$ ;提出的基于 Pyramid 码的局部性修复编码在单节点故障时需要访问局部修复组内的 2 个存活节点,且节点存储开销为 $B/k$ ,则修复带宽开销为 $2B/k$ 。同样地方式可以得到两节点故障时,三副本修复策略、局部性修复编码以及基于 Pyramid 码的局部性修复编码的修复带宽开销和修复局部性。

同样取 $f = 3$ ,图 7 所示为单节点故障时,几种编码方案的修复带宽开销曲线。容易看出,随着数据节点个数 $k$ 的增加,基于 Pyramid 码的局部性修复编码、三副本复制策略以及简单再生码的修复带宽开销降低。取相同的数据节点个数 $k$ ,本文提出的基于 Pyramid 码的局部性修复编码获得了最小的修复带宽开销。同样假定分布式存储系统存在 $k = 30$ 个数据节点,基于 Pyramid 码的局部性修复编码的修复带宽开销 66.67 Mb,远小于简单再生码 133.33 Mb 的修复带宽开销,同时也小于三副本复制策略 100 Mb 的修复带宽开销。

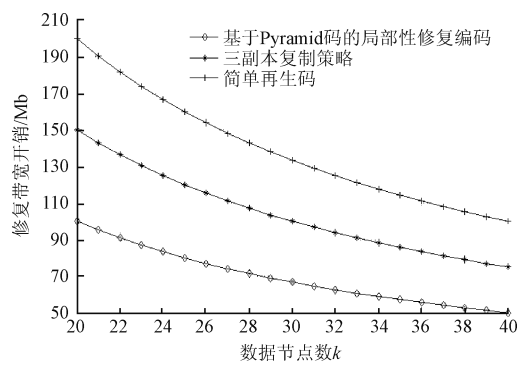



图 7　修复带宽开销对比

Fig. 7　Comparison of repair bandwidth overhead

修复局部性也就是故障节点修复时的磁盘读取开销。当单节点发生故障时,三副本复制策略只需要从存储了故障节点数据的一个存活节点中下载数据块,其磁盘读取开销为 1,修复局部性也为 1;局部性修复编码中的简单再生码,在修复一个故障节点时总共需要从 $f$ 个节点下载数据块和校验块,因此其磁盘读取开销为 $f$ ,修复局部性为 $f$ ( 跟前文保持一致此处取 $f = 3$ );基于 Pyramid 码的局部性修复编码中发生单节点故障时,只需在局部修复组内连接两个存活节点即可,因此其修复局部性为 2。由此可以看出单节点故障时,基于 Pyramid 码的局部性修复编码的修复局部性优于简单再生码的修复

局部性,三副本复制策略具有最小的修复局部性。

当两个节点同时发生故障时,简单再生码需要连接 $k$ 个节点才能实现故障节点修复,其修复局部性为 $k$。根据图 5 中基于 Pyramid 码的局部性修复编码的局部修复组内两个故障节点修复过程示意图,基于 Pyramid 码的局部性修复编码修复两个故障节点只需要连接 3 个或者 4 个存活节点,进行 3 次或者 4 次磁盘访问,修复局部性为 3 或者 4。可以看出在修复两个故障节点时,基于 Pyramid 码的局部性修复编码的修复局部性远优于简单再生码的修复局部性 $k$,接近三副本复制策略的修复局部性 2。

## 5 结 论

现有的局部性修复编码只有在修复一个故障节点时具有很好的修复局部性,在修复两个或多个故障节点时需要连接多个存活节点,修复局部性较高,且没有考虑节点的存储开销。本文介绍了基于 Pyramid 码的局部性修复编码的构造方法,采用 Pyramid 码的最小可实现编码结构,划分局部修复组,确保该编码具有较低的修复局部性并能实现故障节点的快速修复。理论分析和性能仿真表明,基于 Pyramid 码的局部性修复编码可实现分布式存储系统中多个故障节点的同时快速修复;相对于三副本复制策略以及简单再生码,基于 Pyramid 码的局部性修复编码具有最优的存储开销和修复带宽开销性能,且其修复局部性接近于三副本复制策略的修复局部性。

## 参考文献


[1] LIU Y, VLASSOV V. Replication in distributed storage systems: State of the art, possible directions, and open issues [C]. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, 2013: 225-232.

[2] BORTHAKUR D. The hadoop distributed file system: Architecture and design [J]. Hadoop Project Website, 2007, 11(11): 1-10.

[3] ARNOLD J. Openstack Swift: Using. Administering, and Developing for Swift Object Storage [M]. Sebastopol: O'Reilly Media, Inc., 2014.

[4] CALDER B, WANG J, OGUS A, et al. Windows azure storage: A highly available cloud storage service with strong consistency [C]. 23rd ACM Symposium on Operating Systems Principles, 2011: 143-157.

[5] LI J, LI B. Erasure coding for cloud storage systems: A survey [J]. Tsinghua Science and Technology, 2013, 18(3): 259-272.

[6] DIMAKIS A G, GODFREY P B, WAINWRIGHT M J, et al. Network coding for distributed storage systems [C]. 26th IEEE International Conference on Computer Communications (INFOCOM 2007), 2007: 2000-2008.

[7] 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.

[8] ZHOU T, LI H, ZHU B, et al. STORE: Data recovery with approximate minimum network bandwidth and disk I/O in distributed storage systems [C]. IEEE International Conference on Big Data, 2014: 33-38.

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

[10] HUANG P, YAAKOBI E, UCHIKAWA H, et al. Linear locally repairable codes with availability [C]. IEEE International Symposium on Information Theory (ISIT), 2015: 1871-1875.

[11] PAPAILIOPOULOS D S, LUO J Q, DIMAKIS A G, et al. Simple regenerating codes: network coding for cloud storage [C]. 2012 Proceedings IEEE INFOCOM, 2012: 2801-2805.

[12] SHAHABINEJAD M, KHABBAZIAN M, ARDAKANI M. An efficient binary locally repairable code for hadoop distributed file system [J]. IEEE Communications Letters, 2014, 18(8): 1287-1290.

[13] GOPARAJU S, CALDERBANK R. Binary cyclic codes that are locally repairable [C]. IEEE International Symposium on Information Theory (ISIT), 2014: 676-680.

[14] HUANG C, CHEN M, LI J. Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage system [C]. 6th IEEE International Symposium on Network Computing and Applications (NCA 2007), 2007: 79-86.

[15] GOPALAN P, HUANG C, SIMITCI H, et al. On the locality of codeword symbols [J]. IEEE Transactions on Information Theory, 2012, 58(11): 6925-6934.


## 作者简介


**王静**,2009 年于西安电子科技大学获得博士学位,现为长安大学副教授,主要研究方向为分布式存储、再生码和局部性修复编码。

E-mail:jingwang@chd.edu.cn

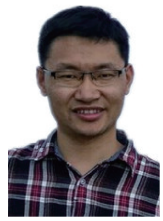
**Wang Jing** received Ph. D. from Xidian University in 2009. And now, she is an associate professor in Chang'an University. Her main research interests include distributed storage, regenerating codes and local repairable codes.

**张崇**，长安大学硕士研究生，主要研究方向为分布式存储和再生码。
E-mail：zhangchong365@126.com

**Zhang Chong** is M. Sc. candidate in Chang'an University. His main research interest includes distributed storage and regenerating codes.

**梁伟**，湖南科技大学副教授，主要研究方向为网络编码、实时嵌入式系统。
E-mail：wliang@hnust.edu.cn

**Liang Wei** is an associate professor in Hunan University of Science and Technology. His main research interests include network encoding, and realtime embedded system.



**刘向阳**，2010 年于西安电子科技大学获得博士学位，现为西安通信学院讲师，主要研究方向为分布式存储、信号检测。
E-mail：xiangyangliu@mail.xidian.edu.cn

**Liu Xiangyang** received Ph. D. from Xidian University in 2010. And now he is a lecturer in Xi'an Communications College. His main research interests include distributed storage, and signal detection.

# 是德科技发布业内首款支持信令连接的 5G 射频设计验证测试工具套件，加速新一代 5G 终端设备的开发

新工具套件让开发人员得以快速验证 5G 射频要求并进行深入分析

**新闻要点：**

- 使用丰富多样的射频测量套件执行传导测试和空中（OTA）测试，对 5G 设备进行验证
- 支持验证 5G 终端片上阵列天线的波束赋形和波束控制
- 通过灵活开发测试用例、快速执行测试活动以及深入的结果分析，验证 5G 波形和参数

是德科技公司（NYSE：KEYS）近日宣布推出行业内首个网络模拟器解决方案 5G RF DVT 工具套件。新工具套件能经济高效地从 6 GHz 到扩展到毫米波，以及从 Pre-5G 标准扩展到新空口（5G NR）。

5G RF DVT 工具套件为是德科技 5G 网络模拟器解决方案（NES）产品组合中的最新成员，它以是德科技率先推向市场的 UXM 5G 无线测试平台（2017 年 5 月发布）为基础设计而成。该工具套件旨在确保从早期原型设计到验收和制造的全过程中，测量具有良好的可追溯性。

是德科技副总裁兼无线设备和运营商事业部总经理 Kailash Narayanan 表示："随时进行 5G 信令连接下的射频测试对于验证设备性能至关重要，但由于毫米波频率和波束赋形的原因，测试具有极大的挑战性。是德科技率先推出相应的解决方案，使业界开发 5G 设备项目的进程显著加快。"

5G RF DVT 工具套件使用是德科技的测试自动化平台（TAP），允许设计工程师轻松创建并定制具有最高参数化程度的射频和无线资源管理（RRM）测试用例。

**更多信息**

关于 5G 协议研发工具套件的更多信息，请参见 www.keysight.com/find/5G-Protocol。浏览产品图片，请访问 www.keysight.com/find/5G-Protocol-images。观看是德科技 5G 网络仿真解决方案的视频演示，请访问 YouTube。

**是德科技 5G**

现代化的工具是推动 5G 通信技术发展的必要保证，它们可以帮助工程师轻松探索新的信号、场景和拓扑技术。是德科技 5G 解决方案能够随着标准的发展即时提供更深入的洞察力。在设计和测试方面，是德科技支持业界领先厂商在全新技术和现有技术方面持续创新，将设计创意变成实际产品。有关是德科技 5G 设计、测试和测量解决方案的详细信息，请访问 www.keysight.com/find/5G。