·先进计算与数据处理· 文章编号: 1000-3428(2019)11-0074-07 文献标志码: A 中图分类号: TP391

分布式存储系统中的纠删码容错方法研究

孙 黎[1,2],苏 宇[2],张 弛[2],张 涛[2]

(1.中国科学院大学,北京 100094; 2.中国科学院空间应用工程与技术中心,北京 100094)

摘 要: HRC码是一种具有存储效率高、计算复杂度低等优点的纠删码,但其存在编解码计算开销大、实现较为复杂等不足。通过对HRC码的译码算法进行优化,提出一种新型的纠删码HRCSD。采用内外层分层结构,内部的冗余由HRC码的编码结构组成,外层采用偏移复制策略,将原始信息进行旋转存储,能够实现并行读写。实验结果表明,与三副本技术和S^2-RAID纠删码相比,HRCSD纠删码具有容错性能高、修复开销低等优势,可满足大规模分布式存储系统的容错需求。

关键词: 分布式存储系统; 纠删码; 数据容错; 数据编码; 数据冗余

开放科学(资源服务)标志码(OSID):

中文引用格式: 孙黎,苏宇,张弛,等. 分布式存储系统中的纠删码容错方法研究[J]. 计算机工程,2019,45(11):74-80.

英文引用格式: SUN Li,SU Yu,ZHANG Chi,et al. Research on fault-tolerant method of erasure code for distributed storage system[J]. Computer Engineering,2019,45(11):74-80.

Research on Fault-Tolerant Method of Erasure Code for Distributed Storage System

SUN Li[1,2],SU Yu[2],ZHANG Chi[2],ZHANG Tao[2]

(1. University of Chinese Academy of Sciences,Beijing 100094,China;
2. Technology and Engineering Center for Space Utilization,Chinese Academy of Sciences,Beijing 100094,China)

【Abstract】 HRC code is an erasure code with high storage efficiency and low computation complexity,but it has some shortcomings,such as high computational overhead and complex implementation. So,we propose a new erasure coding type,the HRCSD code based on the optimized decoding algorithm of HRC code. The HRCSD code adopts an inner and outer layering structure. The inner redundancy is composed of the coding structure of the HRC code,while the outer layer adopts the offset copy strategy to rotate the original information and store it for parallel reading and writing. Experimental results show that compared with the three-copy technology and the S^2-RAID erasure code,the HRCSD has higher fault-tolerant performance and lower repair overhead,which can satisfy the fault-tolerant requirements of large-scale distributed storage systems.

【Key words】 distributed storage system; erasure code; data fault-tolerant; data encoding; data redundancy

DOI: 10.19678/j.issn.1000-3428.0052959

0 概述

信息资源的爆炸性增长,使得信息技术从以计算设备为核心的计算时代进入到以存储为核心的存储时代。分布式存储系统是存储技术、计算技术和网络技术的融合,其目标是利用廉价的硬盘构建大规模、高可靠性、可扩展性的存储系统。然而,存储节点的分散性、异构性及易失效性,给数据的可靠性带来诸多难题和挑战。为提高分布式存储系统中数据的可靠性和可用性,大量的研究集中在单一地解决数据的容错或数据的恢复问题上[1],而对于能够提升数据容错性能并降低数据修复开销[2-3]的研究较少。

通常的冗余容错方法包括多副本技术和纠删码技术。多副本容错技术是将原始数据块复制多个相同的副本,把多个不同的副本存储到不同的节点上,当有节点失效时,可以通过复制其他未失效节点上的副本数据对失效的数据块进行恢复。这种方式简单直观,易于实现和部署,并且修复原理简单,修复开销较低[4]。

纠删码技术通过对原始数据进行编码,将得到

基金项目:载人航天重大专项(Y6140511RN)。

作者简介:孙 黎(1988—),女,博士研究生,主研方向为分布式存储系统;苏 宇、张 弛,工程师;张 涛,研究员。

收稿日期:2018-10-22 修回日期:2018-12-12 E-mail:sunli_hello@hotmail.com

的冗余数据一并存储于不同的节点之上,以达到容错的目的。当数据发生故障时,可以通过读取剩余存活数据,使用译码算法对故障节点进行修复。研究人员提出多种不同类型的纠删码容错方法,如纠双错 MDS 阵列码、EVENODD[5]、X-code[6]、RDP 码(Row-Diagonal Parity)[7]、自由码(Liberation Codes)[8]等。文献[9]在 EVENODD 基础上提出了纠多列错的 MDS 阵列码 G-EVENODD。文献[10]扩展了 EVENODD 码,并提出能够纠正任意 3 个节点故障的 STAR 码[10]。2005 年,IBM Almaden 实验室基于 RAID 存储系统先后提出了纠多磁盘故障的 WEAVER 码[11]、HoVer 码[12]和 R5X0 码[13]。

在纠删码的使用上,需要考虑以下问题: 1) 数据冗余度,即在纠删性能一定的情况下,需要产生多少冗余数据才能满足系统的可靠性需求,根据数据冗余可将编码分为 MDS 码和非 MDS 码,MDS 码具有较优的存储效率; 2) 编码复杂度,由于存储系统需要进行一定的计算来产生校验冗余数据,因此产生冗余数据所需要的计算量也是衡量纠删码优劣的一个重要指标,存储系统中纠删码的编码复杂度直接决定文件存储的时间; 3) 译码复杂度,当系统出于某种需要而对文件进行重构时,对数据块的重构实质是一个译码过程,即在该过程中,系统同样需要付出一定的计算来重构出所需文件块。因此,译码复杂度决定着在需要对数据块进行重构时文件的修复速度。

本文在研究 HRC 编码原理的基础上,提出一种分布式存储系统中纠删码容错方法。通过对原始数据进行编码,将得到的冗余数据存储于不同的节点上,以满足大规模分布式存储系统中并行读写的需求。

1　HRC 码的编码过程

HRC(Horizontal Rotary Codes)码是一种编译码复杂度低、结构规整的水平码[14]。HRC 的编码过程如下: 令 p 为素数,编码 HRC 的每一个码字代表一个阵列,令 $c_{ij}(0\leqslant i\leqslant p-1, 0\leqslant j\leqslant p)$ 为编码 HRC 构成的 $p\times(p+1)$ 阵列中元素,在该阵列中,前 $p-1$ 列,即 $0\leqslant j\leqslant p-2$ 代表信息符号,后两列,即 $p-1$ 列和 p 列代表校验符号,并且假定存在一个虚拟的全为 0 的第 0 行,即 $c_{0,j}=0\ (0\leqslant j\leqslant p-2)$。HRC 的编码过程是利用信息位计算出 p 列、$p+1$ 列校验位的过程。首先利用前 $p-1$ 列$(0\leqslant j\leqslant p-2)$计算 $p-1$ 列上的校验位,计算公式如下:

$$c_{i,p-1}=\bigoplus_{k=0}^{p-2}c_{i,k},0\leqslant i\leqslant p-1 \qquad (1)$$

然后利用前 $p-1$ 列上的信息位和第 $p-1$ 列上的校验位计算第 p 列上的校验位:

$$c_{i,p}=\bigoplus_{k=0}^{p-1}c_{<i+k>,k},0\leqslant i\leqslant p-1 \qquad (2)$$

图 1、图 2 为 $p=5$ 时 HRC 码的编码过程及码字结构,其中第 1 行为虚拟 0 行。

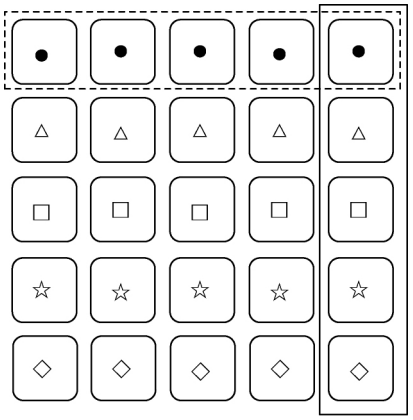


图 1　HRC 码第 1 列校验码字结构

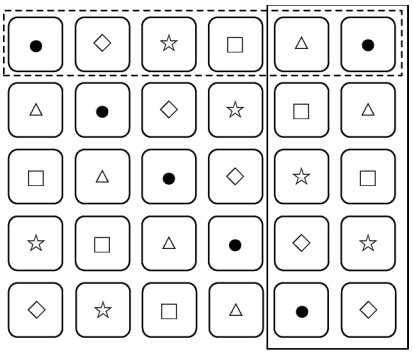


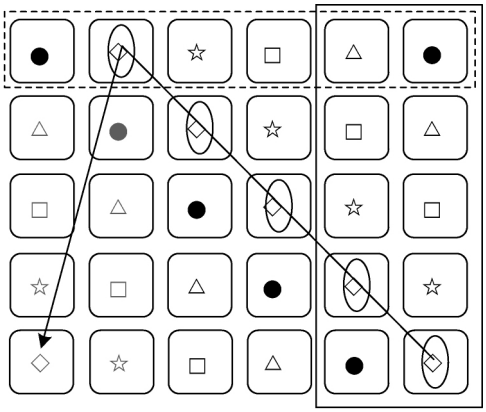
图 2　HRC 码第 2 列校验码字结构

图 1 描述了 HRC 的第 1 列校验数据为每行信息数据的水平校验,图 2 描述了 HRC 的第 2 列校验数据是信息数据的斜率为 -1 的斜向校验。

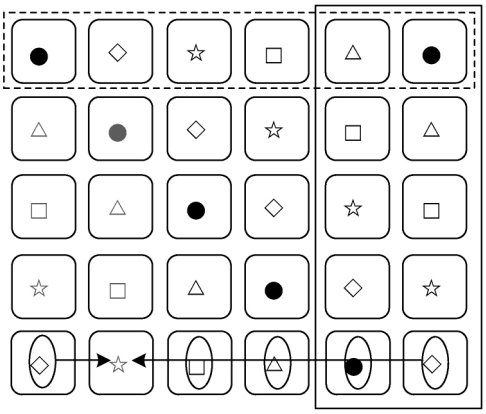
2　HRC 译码算法

当节点发生故障时,需要使用剩余存活数据通过译码算法对节点进行修复,文献[16]中的译码算法使用到第 1 行虚拟行的对角线校验数据 $c_{0,p+1}$ 对丢失数据块进行修复,这就需要在编码时额外地开辟存储空间对虚拟行的对角线校验块进行存储。本文提出一种优化的译码算法,该译码算法不需要使用 $c_{0,p+1}$ 对数据进行修复,能够节约存储空间,提升译码算法的效率。译码过程依据丢失数据块的不同可以分为 2 种情况: 1) 丢失的两列均为校验列,则由编码式(1)和式(2)直接计算求得; 2) 若丢失的两列中至少有一列为原始信息数据,则可采用迭代方式进行数据恢复,如图 3 所示。假设丢失的为第 1 列和第 2 列信息数据。首先,选择2 列信息数据中只有一个参与了斜校验信息位的数据进行恢复,其次,使

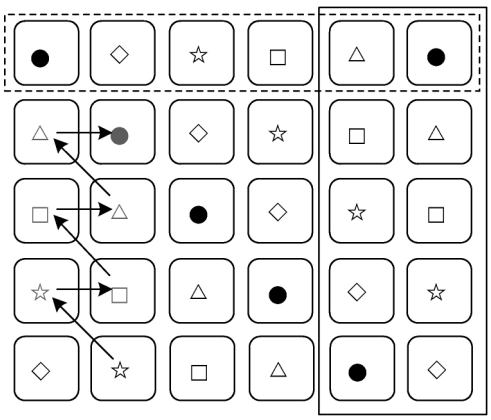
用行校验列对与刚修复的数据位处于同一行的丢失数据位进行恢复，循环操作以上 2 个步骤，对丢失的两列数据信息进行迭代修复，直至所有的信息数据修复完成。详细译码算法过程如算法 1 所示。



(a)使用斜校验集合的数据恢复



(b)使用行校验集合的数据恢复



(c)所有丢失数据块的交叉恢复

图 3 HRC 码递归译码算法的实现过程

算法 1 HRC 中恢复两行丢失的原信息列数据

输入 丢失两列原信息数据的码字阵列

输出 完整的码字阵列

1. 令 $s \leftarrow j_2 - j_1$

2. 执行循环，for($l=0; l<p-1; l++$){

a. $c_{p-<s+l\times s>,j_1} \leftarrow c_{p-<j_1+s+l\times s>,p} \oplus_{k=0,k\neq j_1}^{p-1} c_{<p-j_1-s-l\times s+k>,k}$;

b. $c_{p-<s+l\times s>,j_2} \leftarrow c_{p-<s+l\times s>,p-1} \oplus_{k=0,k\neq j_2}^{p-2} c_{p-<s+l\times s>,k}$.}

3. end for

3 HRC 码的偏移重复码: HRCSD

3.1 HRCSD 码的编码过程

文献[15]在 RAID5 的基础上提出了一种 S^2-RAID 纠删码，S^2-RAID 采用了数据偏移的子结构，能够并行地对数据进行重构，加快恢复过程。本文在研究 HRC 编码原理的基础上，给出容错性能更高的 HRC 偏移重复码(HRC Skewed Duplication，HRCSD)，该码具有较高的容错性能，并能满足大规模分布式存储系统中并行读写的需求。

HRC 码能够容忍存储系统中任意不多于2 个节点磁盘同时发生故障。HRCSD 是 HRC 的偏移重复码，该码参照 S^2-RAID 的工作原理，将数据块分割成数据片，并将数据片信息进行偏移分散保存于不同的磁盘之上，HRCSD 采用内外层分层结构，内部的冗余仍由 HRC 码的编码结构组成，外层采用偏移复制策略，将原始信息进行旋转存储，而外层采用 S^2-RAID 相似的存储结构对数据进行复制备份。

HRCSD 将原始数据块信息划分成 Group1 和 Group2 两组。编码结构由内外两层组成，内层结构使用 HRC 编码对数据位信息进行编码，第 1 列校验位 P_0 为数据信息的横向校验，第 2 列校验位 P_1 为数据信息的斜率为 -1 的斜校验。同理，Group2 中的内层编码结构与 Group1 中使用的 HRC 编码结构相同，Group2 中的原始数据信息为 Group1 中的原始数据信息递归右旋分散排列到磁盘 $D4 \sim D7$ 中。图 4 给出一个 HRCSD 的结构，12 列($N=12$)磁盘数据(其中数据信息块有 8 列，校验信息块有 4 列)被分成 2 组($G=2$)，每个列数据块被分成 4 个部分($K=4$)，Group2 中 $D_4 \sim D_7$ 的第 1 行条带存储的数据信息与 Group1 中 $D_0 \sim D_3$ 的第 1 行条带存储的数据信息相同，$D_4 \sim D_7$ 的第 2 行条带存储的数据信息由 $D_0 \sim D_3$ 第 2 行条带存储的数据信息右移一位得到，$D_4 \sim D_7$ 的第 3 行条带存储的数据信息由 $D_0 \sim D_3$ 第 3 行条带存储的数据信息右移两位得到，$D_4 \sim D_7$ 的第 4 行条带存储的数据信息由 $D_0 \sim D_3$ 第 4 行条带存储的数据信息右移三位得到。具有 N 列数据的磁盘阵列被分成两组，每个列数据块横向分割为 k 个部分，且 k 等于每组中的数据块列数，这样每组中的

数据块被划为一个 $k \times k$ 的数据阵列,其中 Group0 的数据块信息由 M_0 表示,Group1 的数据块信息由 M_1 表示,$P_{i,j}$ 表示第 i 组中第$(j+1)$ 行上的 k 个数据信息。$M_0=(P_{0,0},P_{0,1},P_{0,2},P_{0,3})$,$M_1=(P_{1,0},P_{1,1},P_{1,2},P_{1,3})$,$P_{0,0}=(0,1,2,3)$,$P_{0,1}=(4,5,6,7)$ 分别表示 Group0 中第 1 行和第 2 行的条带数据信息,$P_{1,0}=(0,1,2,3)$,$P_{1,1}=(7,4,5,6)$ 分别表示 Group1 中的第 1 行和第 2 行条带数据信息。假设一个 N 列数据列的磁盘阵列,包含 k 行条带数据的 HRCSD 码的两组数据映射关系可由式(3)和式(4)获得。

$$M_0=\begin{pmatrix}P_{0,0}\\P_{0,1}\\\vdots\\P_{0,k-1}\end{pmatrix} \tag{3}$$

$$M_1=\begin{pmatrix}P_{1,0}\\P_{1,1}\\\vdots\\P_{1,k-1}\end{pmatrix}=\begin{pmatrix}SH_r^0(P_{0,0})\\SH_r^1(P_{0,1})\\\vdots\\SH_r^{k-1}(P_{0,k-1})\end{pmatrix} \tag{4}$$

其中,$SH_r^a(P_{i,j})$ 是一个循环移位操作,r 表示移位的方向,a 表示移位的偏移量,$SH_r^a(P_{i,j})$表示将向量循环右移 a 个位置,如 $SH_r^1(P_{0,1})=SH_r^1(4,5,6,7)=(7,4,5,6)$。

Group1						Group2					
D_0	D_1	D_2	D_3	P_0	P_1	D_4	D_5	D_6	D_7	P_2	P_3
0	1	2	3	A_0	B_0	0	1	2	3	C_0	D_0
4	5	6	7	A_1	B_1	7	4	5	6	C_1	D_1
8	9	10	11	A_2	B_2	10	11	8	9	C_2	D_2
12	13	14	15	A_3	B_3	13	14	15	12	C_3	D_3

图 4　HRCSD 编码结构

HRCSD 的内层结构中仍使用 HRC 两列校验位的编码方式进行编码,编码后的校验值存储于 P_0 ~ P_3 中,其中 P_0 和 P_2 均为行校验,且校验的结果相同,P_1 和 P_3 为斜对角线检验,校验的结果不同。HRCSD 具有更高的容错能力,且能够实现数据的并行读取,单节点的故障恢复速度更快。

3.2　HRCSD 的容错性能分析

HRCSD 最多能够同时容忍任意 6 列磁盘数据发生故障,在 HRCSD 中同时发生 6 列磁盘数据故障的情况可以分为以下 3 种: 1) 发生故障的 6 列磁盘属于同一个 Group 组中,即 Group1 或 Group2 中的全部数据丢失; 2) 有一个组中发生 1 列或 2 列故障,另一组中发生 5 列或 4 列故障; 3) 2 个组中均有 3 列数据发生故障。从以上 3 种情况分析可以看出,HRCSD 能够恢复出所有失效数据块。

第 1 种情况: 由于 Group2 是 Group1 数据偏移得到的,如果 Group1 中的信息数据全部丢失,则需要对 Group2 中的原始数据块信息相应地进行左移偏移得到 Group1 的原始数据信息,而 Group1 中丢失的校验位信息则可由 HRC 码编码公式获得; 如果丢失的 6 列数据均在 Group2 中,则可使用式(3)、式(4)对 Group2 的数据信息进行恢复,再使用 HRC 码的编码公式得到 Group2 的校验信息列。因此,当丢失的数据块均在一组中时,能够对数据进行恢复。

第 2 种情况: 当 Group1 或 Group2 中的 1 列或 2 列数据发生故障时,可以先采用上文中提到的 HRC 的译码算法对丢失的 1 列或 2 列数据进行修复,得到 Group1 或 Group2 完整的数据信心,再采用偏移复制的方法对丢失了 5 列或 4 列数据信息的组进行原始信息的偏移复制,丢失的校验列可对原始数据信息进行编码恢复。

第 3 种情况: 这种情况下数据的恢复较复杂,以 2 组中丢失的 3 列均为原始数据信息列的数据修复情况进行说明。假设丢失的 6 列为 Group1 中的 D_0 ~ D_2 原始信息列和 Group2 中的 D_4 ~ D_6 原始信息列,如图 5(a)所示。恢复过程可分为以下 5 步:

1) 读取 Group1 和 Group2 中未丢失的原始信息位并对 Group2 和 Group1 中相应的丢失信息位进行恢复,Group1 中恢复出{6,9,12},Group2 中恢复出{7,11,15}。

2) 使用斜校验信息集合对 2 组中只丢失了一位信息位的原始数据进行恢复,如图 5(b)所示,恢复出 Group1 中原始数据信息{2}。

3) 将 Group1 中恢复出的原始数据信息复制到 Group2 中对应位置,并使用斜校验集合对 Group2 中只丢失了一位信息位的原始数据进行恢复,如图 5(c)所示,恢复出 Group2 中原始数据信息{13}。

4) 将 Group2 中恢复的原始数据信息复制到 Group1 中,同时使用斜校验集合和行校验集合对 Group1 和 Group2 中的原始数据信息进行恢复,如图 5(d)所示,恢复出 Group1 中原始数据信息{8}和 Group2 中原始数据信息{14}。

5) 将恢复的原始数据信息进行复制,使用循环迭代,依次修复{4,10}、{5,0}、{1},直至 2 组中所有丢失的原始数据信息都得到恢复。

图 5 HRCSD 修复过程

由以上分析可知，HRCSD 最多能够容忍任意 6 列磁盘同时发生故障，并能对发生故障的磁盘进行有效的修复，若发生故障的磁盘包含校验列，仍然可以使用以上修复算法进行故障修复。因此，HRCSD 的容错能力为 6，可以为分布式存储系统提供更高的容错能力，提升系统的可靠性。

3.3 HRCSD 的并行修复算法

通过对原始数据的偏移复制能够实现并行读的功能，在恢复过程中，可以使用并行的方法对丢失的原始数据进行修复，能够有效降低修复时间，提升修复效率。算法 2 是 HRCSD 的并行修复算法，其中，Group1 中丢失了 $a(0\leqslant a\leqslant 3)$ 列信息块，Group2 中丢失了 $b(0\leqslant b\leqslant 3)$ 列信息块。

算法 2 HRCSD 中恢复故障磁盘的并行修复算法

输入 丢失 m 列信息块($0\leqslant m\leqslant 6$)的阵列

输出 完整的磁盘阵列

1. 读取 Group1 和 Group2 中未丢失的数据信息，恢复对应的丢失的数据信息;
2. if a = = 0 && b = = 0
3. print("no error")
4. while(a ! = 0 ‖ b ! = 0)
5. if a < = 2 ‖ b < =2
6. 使用 HRC 译码算法对 Group1 和 Group2 进行并行修复;
7. else if
8. 同时寻找 Group1 和 Group2 中能够被恢复的信息块进行修复;
9. 将恢复出来的数据块进行交叉复制;
10. end if
11. end while
12. return Group1 和 Group2 的完整码字阵列

3.4 HRCSD 性能分析

基于多副本的容错技术容易实现，并且能够提供并行的数据读取，因此在实际的分布式存储系统中得到了广泛的应用，但其需要消耗大量的存储空间。基于纠删码的容错技术能够极大地节省存储空间，越来越适用于大数据存储环境下的数据容错。纠删码的优点是能够解决存储空间，实现高性能的容错，但也具有编解码计算开销较大、实现较为复杂等缺点。本节首先讨论基于 HRCSD 码的存储开销，然后理论分析 HRCSD 码的容错性能，最后对 HRCSD 码的修复开销进行分析，并分别与三副本容错技术和 S^2-RAID 码进行对比。

3.4.1 存储开销

存储效率是指数据对象的实际大小 M 与数据对象在存储系统中实际占用的存储空间 S 之间的比例。本文考虑对存储了 p 列原始信息的磁盘阵列进行3 种方式的容错，其中，p 为素数，三副本容错技术需要 $3p$ 的存储空间来存储数据，S^2-RAID 码由于使用了 RAID5 容错和 2 次偏移复制容错技术，使得存储容量大大增加，存储 p 列原始信息需要产生 $4p$ 列的内存开销。而本文提出的 HRCSD 码，内层中使用 HRC 码的 MDS 高性能存储结构进行内部容错，外部采用一次偏移复制，只需要增加 4 列校验信息位，所以存储 p 列数据所需的存储开销为 $2p+4$。随着 p 的增大，3 种容错策略所占用的存储开销如图 6 所示。

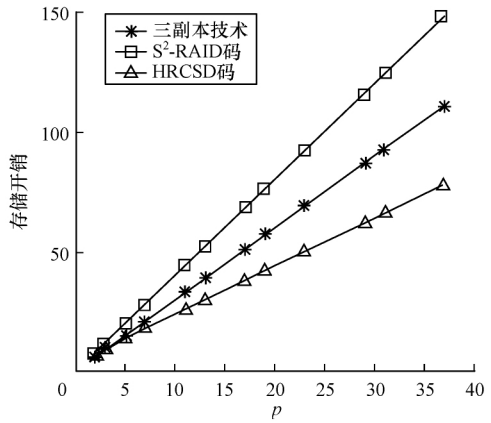


图 6　3 种容错方式存储开销对比

3.4.2　**容错能力**

容错能力指分布式存储系统中在保证数据可用的情况下,可以接受的任意节点失效的最大数目。如果容错能力为 k,意味着对于任意小于等于 k 块的失效都是可以恢复的,但是存在一个有 $k+1$ 个失效块的组合,当这个组合中的数据失效时,数据对象将无法恢复。多副本的容错能力通常与副本的个人有关,将数据备份了 k 个副本的分布式存储系统的容错能力为 $k-1$,通常 HDFS 使用三副本备份策略,备份的副本数为 2,容错能力为 2。文献[7]中提出的 S^2-RAID 码的容错能力为 1,3.2 节证明了 HRCSD 具有最多能够容忍 6 列磁盘数据发生故障的能力,所以 HRCSD 的容错能力为 6,通过3 种编码容错能力的对比,本文 HRCSD 码具有较高的容错性能。

3.4.3　**成本修复**

成本修复是指在修复故障数据块时,需要读取的未失效节点的数据量和待修复的数据块的大小之间的比例。本节针对三副本、S^2-RAID 码和 HRCSD 码单故障节点情况的修复成本进行分析比较。

三副本容错策略在修复单节点故障时只需要选取未失效的 2 个副本中的数据进行复制,所以修复单节点的数据量与待修复的数据量之比为 1,具有较低的修复成本; S^2-RAID 由 l 个分组组成,每个组中包含 g 个磁盘,使用 RAID5 结构进行编码容错,修复过程中需要从另外的组中下载未失效的数据进行恢复,所以修复一个失效的数据块需要读取 $g\times(l-1)$ 个数据块的数据量进行解码恢复,修复成本为 $l-1$; 当 HRCSD 中发生单故障节点失效时,可以采用复制容错的修复方法复制未失效的数据块,对失效的节点进行恢复,所以修复成本为 1。

3.4.4　**时长计算**

分布式存储系统中使用纠删码对数据进行容错会增加系统的计算开销,计算时长取决于纠删码的计算复杂度,也可由纠删码编码过程中使用的异或次数反映。三副本技术只需要对数据进行复制实现容错,不需要使用基于异或方式的计算,所以不会产生计算开销。S^2-RAID 码是基于 RAID5 的容错,产生冗余的方式与 RAID5 的冗余方式相同,即对条带数据依次进行异或,产生一位的冗余校验数据,k 位原始数据信息需要进行 $k-1$ 次异或操作,$k\times k$ 阵列的数据信息进行容错共需要 $k\times(k-1)$ 次异或操作,S^2-RAID 码是对原始数据进行 2 次偏移复制得到,每个 Group 中都使用 RAID5 的容错方式进行容错,所以 3 个 Group 共需要 $3\times k\times(k-1)$ 次异或操作。HRCSD 是在 HRC 的基础上进行偏移复制,计算复杂度与 HRC 的计算复杂度相同,HRC 相比于 RS 纠删码和 EVENODD 纠删码具有更低的计算复杂度[16],因此,HRCSD 的计算复杂度也相对较低,HRC 的 $k\times k$ 阵列的数据信息进行容错产生两列校验位共需要 $2\times(k-1)\times(k-2)$ 次异或操作,HRCSD 共需要 $4\times(k-1)\times(k-2)$ 次异或操作。3 种容错方式的计算复杂度对比如表 1 所示,其中,NA 表示三副本技术不需要对原始数据进行异或运算。

表 1　三副本技术、S^2-RAID 码和 HRCSD 码比较

方法	存储开销	容错能力	修复成本	计算复杂度
三副本技术	$3p$	2	1	NA
S^2-RAID 码	$4p$	1	$l-1$	$3\cdot k\cdot(k-1)$
HRCSD 码	$2p+4$	6	1	$4\cdot(k-1)\cdot(k-2)$

4　结束语

本文针对目前大规模分布式存储系统中数据容错存储效率低、纠删码的编译码计算复杂等问题,提出一种具有较高容错能力且能够快速实现的纠删码容错编码 HRCSD。HRCSD 基于 HRC 进行偏移复制得到,通过复制技术,可满足在分布式系统中对数据的并行读取需求。实验结果表明,与三副本技术和 S^2-RAID 码相比,HRCSD 具有较低的存储开销、较高的容错能力和单节点故障下快速修复的性能。本文提出的 HRCSD 在故障修复过程中考虑的是单一节点发生故障的情况,下一步将研究多节点发生故障的修复方式,以降低修复带宽,加快数据的修复。

参考文献

[1] 林轩,王意洁,裴晓强,等. GRC:一种适用于多节点失效的高容错低修复成本纠删码[J]. 计算机研究与发展,2014,51(增刊):172-181.

[2] 丁尚,童鑫,陈艳,等. 基于简单再生码的带宽感知的分布式存储节点修复优化[J]. 软件学报,2017,28(8):1940-1951.

[3] 刘佩,蒋梓逸,曹袖. 一种基于分布式存储系统中多节点修复的节点选择算法[J]. 计算机研究与发展,2018,55(7):1557-1568.

[4] 罗象宏,舒继武. 存储系统中的纠删码研究综述[J]. 计算机研究与发展,2012,49(1):1-11.

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

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

[7] CORBETT P,ENGLISH B,GOEL A,et al. Row-diagonal parity for double disk failure correction[C]//Proceedings of the 3rd USENIX Conference on File and Storage Technologies. Washington D. C.,USA: IEEE Press,2004:1-14.

[8] PLANK J S. The raid-6 liber8tion code[J]. The International Journal of High Performance Computing Applications,2009,23(3):242-251.

[9] BLAUM M,BRADT J,BRUCK J,et al. The EVENODD code and its generalization: an efficient scheme for tolerating multiple disk failures in RAID architectures,high performance mass storage and parallel I/O[EB/OL]. [2018-09-20]. https://www.researchgate.net/publication.

[10] HUANG Cheng,XU Lihao. STAR: an efficient coding scheme for correcting triple storage node failures[J]. IEEE Transactions on Computers,2008,57(7):889-901.

[11] HAFNER J L. WEAVER codes: highly fault tolerant erasure codes for storage systems[C]//Proceedings of Conference on File and Storage Technologies. San Francisco,USA:[s. n.],2005:11-16.

[12] HAFNER J L. HoVer erasure codes for disk arrays[C]//Proceedings of IEEE International Conference on Dependable Systems and Networks. Washington D. C.,USA: IEEE Press,2006:217-226.

[13] HARTILINE J. R5X0: an efficient high distance parity-based code with optimal update complexity[EB/OL]. [2018-09-20]. https://www.researchgate.net/publication.

[14] 王玉林. 多节点容错存储系统的数据与缓存组织研究[D]. 成都:电子科技大学,2010.

[15] WAN Jiquan,WANG Jibin,YANG Qing,et al. S^2-RAID: a new RAID architecture for fast data recovery[C]//Proceedings of the 26th IEEE Symposium on Mass Storage Systems and Technologie. Washington D. C.,USA: IEEE Press,2010:1-9.

编辑 索书志

(上接第 73 页)

[11] AILEM M,ROLE F,NADIF M. Co-clustering document-term matrices by direct maximization of graph modularity[C]//Proceedings of the 24th ACM International Conference on Information and Knowledge Management. New York,USA: ACM Press,2015:1807-1810.

[12] NEWMAN M E J,GIRVAN M. Finding and evaluating community structure in networks[EB/OL]. [2019-04-20]. http://dx.doi.org/10.1103/PhysRevE.69.026113.

[13] LEE D D,SEUNG H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature,1999,401(6755):788.

[14] YU Xianghao,SHEN Juei Chin,ZHANG Jun,et al. Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems[J]. IEEE Journal of Selected Topics in Signal Processing,2016,10(3):485-500.

[15] GILARRANZ L J,RRYFIELD B. Effects of network modularity on the spread of perturbation impact in experimental metapopulations[J]. Science,2017,357(6347):199-201.

[16] GUNASEKAR S,WOODWORTH B E,BHOJANAPALLI S,et al. Implicit regularization in matrix factorization[EB/OL]. [2019-04-20]. https://www.doc88.com/p-4522872867906.html.

[17] WEI Jie,HE Jie,CHEN Ke,et al. Collaborative filtering and deep learning based recommendation system for cold start items[J]. Expert Systems with Applications,2017,69:29-39.

[18] MELVILLE P,SINDHWANI V. Recommender systems[M]. Berlin,Germany: Springer,2017:1056-1066.

[19] WU C Y,AHMED A,BBUTEL A,et al. Recurrent recommender networks[C]//Proceedings of the 10th ACM International Conference on Web Search and Data Mining. New York,USA: ACM Press,2017:495-503.

[20] ADOMACICIUS G,ZHANG Jingjing. Stability of recommendation algorithms[J]. ACM Transactions on Information Systems,2012,30(4):47-54.

编辑 索书志