基于V码的一种数据布局研究

万武南[1,2]，索 望[2]，陈 运[2]

(1. 电子科技大学计算机科学与工程学院 成都 610054; 2. 成都信息工程学院信息安全研究所 成都 610225)

【摘要】提出一类新的双容错编码——V阵列码，冗余数据均匀分布在每个磁盘中，能容许任意两个磁盘同时故障。并证明基于V码阵列布局是最优双容错数据布局方法，给出了恢复任意两个磁盘同时故障的快速译码算法。与其他的编码方案相比，基于V码阵列布局同时具有较高的可靠性和吞吐量、较好的I/O性能、简单的编码和解码算法，以及编译码的复杂度最低和较好的平衡特性。

关 键 词 数据布局; EVENODD码; X码; V阵列码

中图分类号 TP333 **文献标识码** A doi:10.3969/j.issn.1001-0548.2010.01.030

Data Distribution Strategy Based on V Codes

WAN Wu-nan[1,2], SUO Wang[2], and CHEN Yun[2]

(1. School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 610054;
2. Institute of Information Security, Chengdu University of Information Technology of China Chengdu 610225)

Abstract A class of new binary maximum distance separable (MDS) array codes called V-Code is presented. The V-Codes have a simple geometrical structure, namely, they can correct any double column erasure errors. In addition, the parity symbols are evenly distributed among all the columns, thus the bottleneck effects of repeated write operation are naturally overcome. A detailed V code's decoding algorithms for correcting various double node failures with a geometrical structure is provided. The complexity of encoding and decoding and the small write performance of other types of codes are compared and analyzed.

Key words data distribution strategy; EVENODD codes; X codes; V Array codes

目前，冗余磁盘阵列(RAID)已成为网络存储的主流技术[1]。然而由于硬件系统本身的脆弱性和其他各种不确定因素造成系统的不可用性，数据经常遭受破坏，因此如何提高RAID系统的可靠性是一个重要的课题[2-5]。早期的RAID技术采用奇偶校验码来保障数据的安全性(RAID-3,RAID-5)[4]。由于阵列码的二维码字结构刚好较符合RAID的结构，并且其编译码只需要异或运算，在相同的编码效率下，阵列码比一般线性码更为有效，因此，阵列码被广泛应用于RAID系统[5-11]。

在RAID结构中，容许单个和两个磁盘同时故障的阵列布局已有大量的研究，如EVENODD码[5]、X码[6]、B码[7]、S码[8]等。EVENODD码能够承受两个磁盘故障，但这种码的数据布局限定了磁盘数为素数，并且EVENODD码的校验信息集中在某两个校验盘上，小写额外开销大，容易造成系统I/O瓶颈[5]。基于B码的数据布局也能够承受两个磁盘故障，但是B码是基于图论知识构建的编码，目前缺乏构建B码的代数方法，编译码实现较复杂[7]。X码和S码是另外两种能够承受两个磁盘故障的RAID结构，但是这两种码的码长都有一定的条件约束，限制了磁盘阵列的结构[8]。

本文提出了一种新的基于V码数据布局策略，能容许任意两个磁盘同时故障。此外，并给出了V码编码方法的几何图描述，并利用几何图模型给出了V码的译码过程。与其他RAID结构相比，基于V码数据校验信息不是集中在某一列或者某一行上，而是均匀分布在每个盘的不同位置，有利于解决磁盘阵列I/O问题，编码和译码过程只需要简单的异或运算，空间利用率和系统吞吐量的影响非常小。

1 V码的编码方法

V码是在X码的基础上扩展出来的，因此在介绍V码之前，先简要介绍X码。本文使用以下符号：$\langle a \rangle_m$

收稿日期：2009 – 05 – 23； 修回日期：2009 – 09 – 27
基金项目：国家自然科学基金(60873216)；四川省教育厅青年基金(07ZB012)；电子产业发展基金；四川省科技厅应用基础资助项目(2008JY0078)
作者简介：万武南(1978 –)，女，博士，主要从事信息安全、编码理论等方面的研究.

表示模 m 的运算，即 $\langle a\rangle_m = a(\mathrm{mod}\, m)$；$\langle a\rangle$ 表示模 $2m+1$ 的运算，$\langle a\rangle = a(\mathrm{mod}\, 2m+1)$；$m$、$2m+1$ 的值为大于等于2的素数。

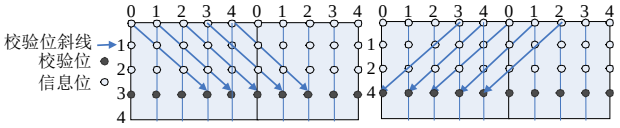
1.1 X码

文献[6]提出X码能够同时容许两个存储设备的故障。X码的码字放在一个 $m\times m$ 的阵列中，其中源信息放前 $m-2$ 行中，最后两行存放校验信息。X码可以记为 $(m, m-2, 3)$ X码，二维码字记为 $C=[c_{i,j}](0\leqslant i\leqslant m-1, 0\leqslant j\leqslant m-1)$，$c_{i,j}$ 为第 i 行第 j 列的信息位或校验位。则两行校验位构造为：

$$c_{m-2,i} = \bigoplus_{t=0}^{m-3} c_{t,\langle i+t+2\rangle_m}$$

$$c_{m-1,i} = \bigoplus_{t=0}^{m-3} c_{t,\langle i-t-2\rangle_m}$$

X码编码过程如图1所示。从图中可知，X码的二维阵列码字C可看作是平面坐标图上的格点，横坐标表示信息位的列号，纵坐标表示信息位的行号，并且坐标轴的取值范围为 $0,1,\cdots,m-1$。如某点坐标为(1,2)，该点表示码字C的 $c_{1,2}$ 信息位，即对应的二维阵列中第2行第3列的信息位。从几何图的角度可得，第1行第i个校验位是从 $(0, i+2(\mathrm{mod}\, m))$ 格点开始，沿固定斜率为1的直线经过的格点对应的信息位异或运算的值。第2行第i个校验位则是从 $(0, i-2(\mathrm{mod}\, m))$ 格点开始，沿固定斜率为–1的直线经过的格点对应的信息位异或运算的值。



a. 第1行校验列　　b. 第2行校验列

图1　X码编码过程

1.2 V码编码方法几何图描述

本文在X码的基础上进行扩展，提出了V码，其码字放在 $m\times(2m+1)$ 阵列中，其中校验数据不再放在阵列的某行，而是均匀分布在阵列的不同位置，每一列(盘)有 $m-1$ 个信息位和1个校验位。每个信息位被用来计算两个校验位；每个校验位是 $2m-2$ 个信息位的异或值，并且 $2m-2$ 个信息位来自阵列不同的列。每个校验位按照以下步骤进行计算：

(1) 当$1\leqslant j\leqslant m$时，有：

$$c_{j,j} = \bigoplus_{\substack{i=1\\ j\text{为奇数},\, i\neq m-\frac{j-1}{2}}}^{m} c_{i,\langle i+j\rangle} \oplus \bigoplus_{\substack{i=m+1\\ j\text{为偶数},\, i\neq\frac{j}{2}}}^{2m} c_{\langle -i\rangle,\langle i+j\rangle}$$

(2) 当$m<j\leqslant 2m$时，有：

$$c_{\langle -j\rangle,j} = \bigoplus_{\substack{i=1\\ j\text{为奇数},\, i\neq m-\frac{j-1}{2}}}^{m} c_{i,\langle i+j\rangle} \oplus \bigoplus_{\substack{i=m+1\\ j\text{为偶数},\, i\neq\frac{j}{2}}}^{2m} c_{\langle -i\rangle,\langle i+j\rangle}$$

下面从几何图的角度描述V码的编码过程。如图2所示，与X码编码结构不同，V码横坐标表示信息位的列号，取值范围为 $0,1,\cdots,2m$，坐标轴的取值则是模 $2m+1$ 运算；纵坐标表示信息位的行号，取值范围为 $0,1,\cdots,m-1$，坐标轴的取值则是模 m 运算。从几何图的角度可得，每列的校验位的值是从该列的后一列斜率为1的直线，以及该列前一列斜率为–1的直线所经过的 $2m-2$ 个信息位的异或结果；并且这两直线形状类似一个V字，因此称为V码。

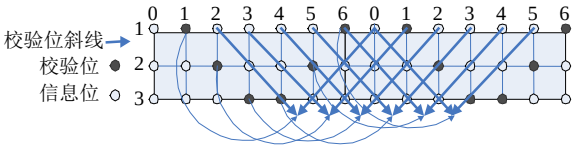


图2　V码编码过程几何示意图

2 V码的译码方法

V码译码过程主要可分为两种情况：

1) 出错1列的情况，这种情况比较简单，本文不进行讨论。

2) 出错为2列的情况，也有以下两种：

(1) 丢失2列数据中1列为信息列，1列为校验列。该情况下译码过程与丢失1列信息列情况译码类似，比较简单，本文不进行讨论。

(2) 丢失2列全部为信息列。设丢失的2列信息列分别为 u_1 和 u_2，其范围为 $0\leqslant u_1<u_2\leqslant 2m-1$。$u_1$ 和 u_2 两列未知变量记为 $c_{u_1}=(c_{1,u_1},\cdots,c_{i,u_1},\cdots,c_{m,u_1})^{\mathrm{T}}$，$1\leqslant i\leqslant m$，$c_{i,u_1}\in c_{u_1}$；$c_{u_2}=(c_{1,u_2},\cdots,c_{i,u_2},\cdots,c_{m,u_2})^{\mathrm{T}}$，$1\leqslant i\leqslant m$，$c_{i,u_2}\in c_{u_2}$。

首先通过剩余列中校验位方程，得到只含有 u_1 和 u_2 列出错信息位的校验算子 $S_{j,j}$，计算规则如下：

(1) 当$1\leqslant j\leqslant m$时，有：

$$S_{j,j} = c_{j,j} \oplus \bigoplus_{\substack{i=1\\ j\text{为奇数},\, i\neq m-\frac{j-1}{2}\\ j\neq u_1,u_2}}^{m} c_{i,\langle i+j\rangle} \oplus \bigoplus_{\substack{i=m+1\\ j\text{为偶数},\, i\neq\frac{j}{2}\\ j\neq u_1,u_2}}^{2m} c_{\langle -i\rangle,\langle i+j\rangle}$$

(2) 当$m<j\leqslant 2m$时，有：

$$S_{\langle -j\rangle,j} = c_{\langle -j\rangle,j} \oplus \bigoplus_{\substack{i=1\\ j\text{为奇数},\, i\neq m-\frac{j-1}{2}\\ i\neq\langle -j\rangle}}^{m} c_{i,\langle i+j\rangle} \oplus \bigoplus_{\substack{i=m+1\\ j\text{为偶数},\, i\neq\frac{j}{2}}}^{2m} c_{\langle -i\rangle,\langle i+j\rangle}$$

V码译码几何示意图如图3所示，u_1=4，u_2=7，2m+1=11。从图中可知，$S_{j,j}$ 每个校验算子至多有两个出错信息位，其中存在两个校验算子 $S_{j,j}$ 含有 u_1 或 u_2 的校验位，即校验算子只有一个信息位出错，

这是V码解码链的起始点，根据这两个校验算子，　　可以依次把出错信息位恢复出来。

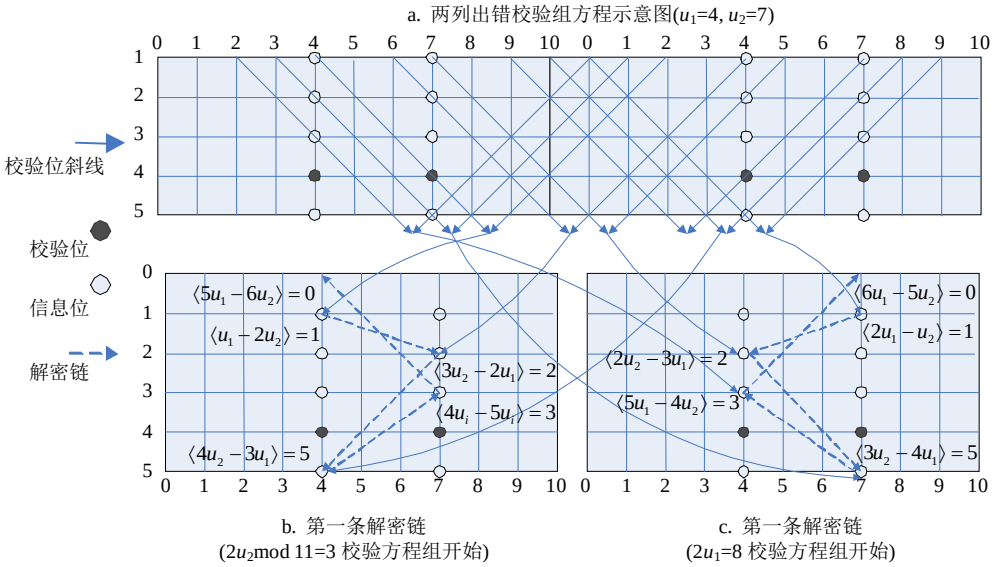


图3　V码译码几何示意图

从几何图模型的角度，V码译码过程描述如下：如图3c所示，当$\langle 2u_2\rangle \neq u_1$时，$\langle 2u_2\rangle$列的校验位的值是从$\langle 2u_2\rangle$列的前一列，沿着斜率为–1和后一列沿斜率为1的两直线经过的2m–2个信息为异或值。刚好其中一条直线横穿u_2列时经过的是u_2列的校验位，因此不含有u_2列的信息位，即计算$\langle 2u_2\rangle$列的校验位的2m–2个信息位中，肯定含有u_1列的信息位。没有u_2列信息位。因此校验算子只含有一个出错信息位，为u_1列的信息位，记为c_{i,u_1}（若$\langle u_1-2u_2\rangle \leqslant n$，则$i=\langle u_1-2u_2\rangle$；否则$i=\langle 2u_2-u_1\rangle$）。因此，通过校验算子可以恢复出第$u_1$列的出错信息位$c_{\langle u_1-2u_2\rangle,u_1}$（或$c_{\langle 2u_2-u_1\rangle,u_1}$）。根据V码编码方法可知，每个信息位用于计算两个校验位，因此可以找含有$c_{\langle u_1-2u_2\rangle,u_1}$（或$c_{\langle 2u_2-u_1\rangle,u_1}$）的另外一个校验算子，该算子含有$u_2$列的一个出错信息位，因此可以计算出$u_2$列的出错信息位$c_{\langle 3u_2-2u_1\rangle,u_2}$（或$c_{\langle 2u_1-3u_2\rangle,u_2}$）⋯；依次计算，直到计算到$u_1$或$u_2$列的信息位的行下标等于零结束，形成一条链，则该条链上u_1列信息位的行下标记为A_0，u_2列的信息位的行下标记为B_0，如图3b所示。

同理从$\langle 2u_1\rangle$列出发，可以由校验算子依次推出u_1和u_2列出错信息位形成的的另外一条链，则该条链上u_1列信息位的行下标记为A_1，u_2列信息位的行下标记为B_1，如图3c所示。

通过这两条解码链，u_1列和u_2列所有的源信息位恢复，最后通过编码公式，把u_1列和u_2列的校验位恢复，至此u_1列和u_2列所有的信息位全部恢复，译码结束。

3　V码性能分析

3.1　编译码复杂度分析

在数据分布式策略中，码的编译码复杂度是衡量磁盘阵列性能的一个重要参数[11]。下面以每个码字编码总的异或次数与码字信息位的总比特位之比来比较码的编译码复杂度。根据上述V码编码方法可知，V码的每个信息位只参与两个校验位的计算，即V码的每比特信息位需要异或的次数为2，达到了双容错数据构造方法最小值；而在EVENODD码编码方法中，EVENODD码需要共同因子参与异或，因此每个信息位需要异或的次数大于2[5]。

下面比较V码与X码、EVENODD码以及RS译码的复杂度，如表1所示。若磁盘总盘数为n=2m+1，根据V码译码算法，首先计算校验算子，需要的异或运算次数为(2m–1)(m–1)+1，通过循环校验算子恢复出两列全部信息位则需要的异或次数为2m–4，V码译码总异或次数为(2m–3)(m–1)+1，因此译码复杂度大约为m–3/2，即(n–4)/2。X码译码过程中恢复出每比特信息位需要的异或次数为n–3[6]，则EVENODD码每比特译码需要的异或次数为n–(n–4)/2(n–3)[5]。基于XOR的RS码译码过程中需要异或的总次数为krL^2(k、r分别为码的信息位和校验位，L为有限域的大小$GF(2^L)$)，忽略有限域的操作r^2，基于XOR的RS的译码复杂度为(n–2)L，只与有限域的大小相关[9]。

表1 四类阵列码译码所需平均异或次数

信息磁盘数(m)	总磁盘数(n)	X码	EVENODD码	V码	RS码
3	5	2	4.00	1.00	9
5	7	4	6.25	1.83	15
7	9	6	8.34	2.75	21
11	13	10	12.40	4.50	44
13	15	12	15.41	5.50	52
17	19	16	18.50	7.50	85
19	21	18	20.50	8.50	95
23	25	22	24.50	10.50	115
29	31	28	30.50	13.50	145
31	33	30	32.50	14.50	155

从表1中可以看出，码的长度比较短时(信息列的列数)，V码的每个信息位需要的异或次数最少，译码性能最好。而基于XOR的RS码的译码复杂度最高，并且随域的扩大而快速增加。

3.2 小写性能和平衡特性的分析

影响数据布局性能的另外两个重要参数为小写性能和平衡特性。在RAID中，当一次写入数据远远小于(或等于)一个信息位时，称为小写。小写时，由于信息位改变，则相应的校验位需要修改，而带来的额外开销会降低阵列的吞吐量，从而影响整个系统的I/O性能[5-6]。在EVENODD码数据布局中，一个信息位改变，平均需要 $4-1/(N-2)$ 次读写操作[5]。而在V码数据布局中，每个信息位只用来计算两个校验位，所有的校验位只依赖信息位，校验位之间是相互独立的，更新一个信息位会导致两个校验位更新，因此只需要3次读写操作，达到了容许两磁盘错误更新复杂性的最低限。

在EVENODD码、RS码的数据布局中，校验位都集中在某两个磁盘中。当对磁盘频繁小写时，需要对校验盘频繁地写操作，造成系统的I/O瓶颈问题。而在V码数据布局中，校验位均匀分布在阵列的每个盘的不同位置。因此小写操作时，修改相应的校验位，不需要集中对某两个盘进行写操作，而是分散到每个盘上，有利于解决磁盘I/O问题。

4 结 论

本文提出了一种基于V码的数据布局，与其他容许两个磁盘故障的数据布局相比较，EVENODD码小写额外开销较高，而RS码编译码计算复杂度较高。V码的数据布局冗余率、编译复杂度、小写额外开销都达到了最优，并且编码和译码编译码算法简单，校验位均匀分布在每个盘上，平衡性好。

参 考 文 献

[1] PATTERSON D A, GIBSON G A, KATZ R H. A case for redundant arrays of inexpensive disks(RAID)[C]//ACM SIGMOD Conference Proceeding. Chicago, USA: ACM, 1988.

[2] FRøLUND S, MERCHANT A, SAITO Y, et al. FAB: enterprise storage systems on a shoestring[C]//Proceedings of the 9th Workshop on HotOS-IX. Kauai, HI: [s.n.], 2003.

[3] 陈华英. 磁盘阵列可靠性分析[J]. 电子科技大学学报, 2006, 35(3): 403-405.
CHEN Hua-ying. Reliability analysis of RAID[J]. Journal of University of Electronic Science and Technology of China, 2006, 35(3): 403-405.

[4] HELLERSTEIN L, GIBSON G A, KARP R M, et al. Coding techniques for handling failures in large disk arrays[J]. Algorithmica, 1994, 12(3-4): 182-208.

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

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

[7] XU L, BOHOSSIAN V, BRUCK J, et al. Low density MDS codes and factors of complete graphs[J]. IEEE Trans on Information Theory, 1999, 45(6): 1817-1826.

[8] KATTI R, RUAN Xiao-yu. S-code: new distance-3 MDS array codes with optimal encoding[C]//Proceedings of IEEE ICASSP' 05. Philadelphia: IEEE, 2005.

[9] LEE N K, YANG S B, LEE K W. Efficient parity placement schemes for tolerating up to two disk failures in disk arrays[J]. Journal of Systems Architecture, 2000, 46(15): 1383-1402.

[10] BLOEMER J, KALFANE M, KARPINSKI M, et al. An XOR-based erasure-resilient coding scheme[R]. ICSI TR-95-048, Technical Report at ICSI, 1995.

[11] JIANG Ming-hua, ZHOU Jing-li, HU Ming. Fuzzy reliability of mirrored disk organizations[C]//Convergence Information Technology. Washington D C, USA: IEEE Computer Society, 2007: 1345-1348.

编 辑 黄 莘