一种三容错数据布局

万武南*①② 王 拓① 索 望①
①(成都信息工程学院网络工程学院 成都 610225)
②(电子科技大学计算机科学与技术学院 成都 610054)

摘 要：随着存储介质的增多，单容错、双容错的数据布局方案已经无法满足现有分布式存储系统对可靠性要求。该文在双容错行对角奇偶校验(Row Diagonal Parity, RDP)码的基础上，提出一种新的扩展行对角奇偶校验(Extending Row Diagonal Parity, E-RDP)码，能够容许任何 3 存储节点出错，具有最大距离可分(Maximum Distance Separable, MDS)编码特性，冗余率与纠错能力达到 3 容错编码最优。并采用不同斜率几何直线图描述编译码过程，给出了一种快速译码算法，易于软硬件实现。与其它纠删码数据布局方案进行比较，理论分析结果表明，E-RDP 码的空间利用率、编译码效率、小写性能以及平衡性的综合性能达到最优，具有实用价值。

关键词：数据存储；编码；纠删码；行对角奇偶校验(RDP)码；可靠性
中图分类号：TP333 **文献标识码**： A **文章编号**：1009-5896(2013)10-2341-06
DOI: 10.3724/SP.J.1146.2013.00153

A Data Placement Based on Toleration Triple Failures

Wan Wu-nan①② Wang Tuo① Suo Wang①
①(*Network Engineering Department*, *Chengdu University of Information Technology*, *Chengdu* 610225, *China*)
②(*College of Computer*, *University of Electronic Science and Technology of China*, *Chengdu* 610054, *China*)

Abstract: With increase of storage devices, the data placements based on toleration single or double failures can not meet the requirement of the reliability in the distributed storage systems. On the basis of the Row Diagonal Parity (RDP) code for double toleration failures, a new class of array codes for triple storage failures is presented which is called Extending Row Diagonal Parity (E-RDP) code. The E-RDP code has the Maximum Distance Separable (MDS) property, and it is optimal in redundancy rate and erasure correcting capability among triple erasure-correcting codes. The procedures of encoding and decoding are depicted by geometrical lines of different slope, then a fast decoding algorithm is given and it is more easily implemented by software and hardware. The theoretical analysis shows that the comprehensive properties of the E-RDP code such as encoding and decoding efficiency, small writes and balance performance, are better than other popular MDS codes, thus the E-RDP code is practically meaningful for storage systems.

Key words: Data storage; Coding; Erasure-correcting codes; Row Diagonal Parity (RDP) code; Reliability

1 引言

近年来，随着海量存储系统的发展和在复杂环境下的应用，存储系统的可靠性受到严重挑战。特别是随着系统中存储节点的增多，存储介质出错或者存储潜在扇区出错概率越来越大。为了最大程度上减少对应用的影响以及提高系统数据的可靠性，可以通过系统冗余提供业务连续性，这样可以大大降低系统的维护成本[1]。常用的冗余技术主要有基于复制和基于纠删码(erasure code)两大类。基于复制的容错技术把数据复制多个副本分别存储，以实现冗余备份。这种方法不涉及编码和重构问题，简单直观，易于部署，但随着系统规模的扩大，存储开销也巨大，导致存储成本非常高。纠删码是一类源于通信传输的编码技术，用于数据传输过程中的检错，后来逐渐被引入到存储系统，以提高存储系统可靠性。基于纠删码冗余技术的基本思想是多个数据块的信息采用纠删码进行融合生成较少的冗余信息，能够有效节省空间[2,3]。

目前，对于双容错纠删码方法已经有许多研究。文献[4–10]分别提出了 EVENODD 码[4]，X 码[5]，B 码[6]，C 码[7]，RDP(Row Diagonal Parity)码[8]，H 码[9] ，HDP[10]码的双容错数据布局，这些双容错编译码只需要异或运算，并具有最大距离可分编码特性，数据冗余率达到了最优。文献[11–13]提出了

2013-01-29 收到，2013-07-02 改回
国家自然科学基金(60873216)和四川省教育厅重点项目(12ZA223)资助课题
*通信作者：万武南 nan_wwn@cuit.edu.cn

GRID[11], HoVer 码[12], WEAVER 码[13]的数据布局，能够纠 4 个以上错误，但是这 3 类编码都不是 MDS 码，随着存储介质的增多，其空间利率只有大概 50%。文献[14–16]提出了 Blaum 码[14], T 码[15], Zigzag Codes[16]低密度奇偶 MDS 码，理论上能够容许多个错误，但其编码矩阵难构造。文献[17,18]分别提出了 STAR 码[17]，EEOD 码[18]，能够容 3 错 MDS 阵列码，但这 2 类编码是基于 EVENODD 码扩展，并且码的小写额外运算分布不平衡，容易造成 I/O 瓶颈。

RDP 码是 Corbett 等人[8]在 EVENODD 码基础上改进的一种双容错阵列码，相比 EVENODD 码，编译码效率提高，并改善了存储设备的小写能力。本文在 RDP 码的基础上提出了一种扩展 RDP 码(E-RDP 码)，保留了 RDP 码具有的小写均衡的性能，并能容许任意 3 个存储节点同时故障，具有 MDS 特性，并采用不同斜率几何直线图描述法，给出了一种快速译码算法，易于软硬件实现。与其它纠删码数据布局方案进行比较，E-RDP 码的空间利用率、编译码效率、小写平衡的综合性能达到最优，且结构简单，具有实用价值。

2 E-RDP 码的编码方法

RDP码是一类双容错阵列码，其中源数据单元放在前 $m-1$ 列中，最后2列存放冗余校验数据单元。E-RDP码是在RDP码的基础上扩展的，增加1列校验列。其编码矩阵为 $(m-1)\times(m+2)$，前 $m-1$ 列中仍然存放源数据信息，最后3列存放冗余校验信息。假设 $a_{i,j}$ 表示为第 j 列第 i 行的源数据单元或者校验单元。令 $0\le u\le m-2$，最后3列校验列各校验数据单元构造公式如式(1)–式(3)所示。

$$a_{u,m-1}=\bigoplus_{t=0}^{m-2}a_{u,t} \tag{1}$$

$$a_{u,m}=\bigoplus_{\substack{t=0\\ t\neq u+1}}^{m-1}a_{\langle u-t\rangle_m,t} \tag{2}$$

$$a_{u,m+1}=\bigoplus_{\substack{t=0\\ \langle u-2t\rangle_m\neq m-1}}^{m-1}a_{\langle u-2t\rangle_m,t} \tag{3}$$

其中 $\langle a\rangle_m$ 表示模 m 的运算，文中 m 表示为大于等于2的素数。

根据式(1)–式(3)可知，E-RDP 码中前 2 列校验列与 RDP 码完全一样，而扩展的最后 1 列校验列是沿着斜率为 2 的直线所经过的源数据单元和水平校验单元的异或值。为了更好描述 E-RDP 码编码的几何特性，在 2 维码字的最后增加 1 行，E-RDP 码变为 $m\times(m+2)$ 的 2 维阵列，增加的最后 1 行对应的数据单元全部为零，称为虚拟数据单元，即 $a_{m-1,i}=0\,(0\le i\le m+1)$，这 1 行实际是不存在的，增加此虚拟行目的是更好描述译码算法。

图1(a)，图1(b)，图1(c)给出了 $m=5$ 时，E-RDP 码3列校验数据单元编码的几何结构。其中横坐标对应E-RDP码字的列号，其范围为 0,1,…, $m+2$。纵坐标为行号，取值范围为 $0,1,\cdots,m-1$，从图中可以很清楚看出，第1列水平校验列的校验单元构造的几何特性就是沿着斜率为0的直线所经过的前 $m-2$ 列的源数据单元的异或值，第2列斜校验列校验单元则为从左下到右上 45° 的斜率为1的直线经过前 m 列的数据单元的异或值，其斜率为1，第3列斜校验列单元校验为从左下到右上30°的斜率为2的直线过前 m 列的数据单元的异或值。从图1还可以看出最后两斜校验列需要水平校验列的数据单元进行异或，因此在编码过程中，需要首先构造水平校验列，才能构造后2校验列。

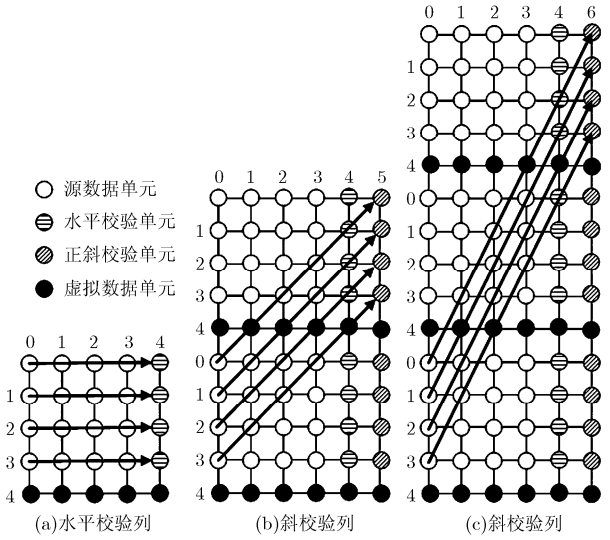

图1 E-RDP码编码几何构造

3 E-RDP 码译码算法

E-RDP 码的译码过程主要可分为两种情况，一种情况数据失效为 2 列(或者说 2 个存储节点)。此时 E-RDP 码实际等同于 RDP 码，可以采用 RDP 双容错和单容错译码算法，此种情况本文不作讨论。另一种情况失效数据为 3 列，此情况又可分为两种：(1)失效数据列有源数据列又有校验列；(2)失效数据列全部为源数据列，校验列没有数据失效。这两种情况译码算法的基本原理都是利用未失效列数据获取校验算子，然后通过几何直线图描述校验算子迭代过程而译码，本文仅对丢失 3 列为源数据列的最

复杂译码情况进行描述。

假设失效的3列源数据列为 i , j , k 列，即 $0 \le i < j < k \le m-2$ 。

第1步　根据式(1)-式(3)计算出每个校验单元的校验算子，3列校验列的校验算子集合分别记为：$S^{(0)}=(S_0^{(0)},S_1^{(0)},\cdots,S_{m-1}^{(0)}), S^{(1)}=(S_0^{(1)},\cdots,S_{m-1}^{(1)}), S^{(2)}=(S_0^{(2)},S_1^{(2)},\cdots,S_{m-1}^{(2)})$ ，令 $0 \le u \le m-2$ ，校验算子计算公式如式(4)-式(6)。

$$S_u^{(0)} = a_{u,m-1} \oplus \bigoplus_{\substack{t=0 \\ t\ne i,j,k}}^{m-2} a_{u,t} \tag{4}$$

$$S_u^{(1)} = a_{u,m} \oplus \left(\bigoplus_{\substack{t=0 \\ t\ne i,j,k}}^{m-1} a_{\langle u-t\rangle_m,t} \right) \tag{5}$$

$$S_u^{(2)} = a_{u,m+1} \oplus \left(\bigoplus_{\substack{t=0 \\ t\ne i,j,k}}^{m-1} a_{\langle u-2t\rangle_m,t} \right) \tag{6}$$

$S_{m-1}^{(1)}$ 和 $S_{m-1}^{(2)}$ 两校验算子则分别由 $S_{m-1}^{(1)} = \left(\bigoplus_{t=0}^{m-2} a_{t,m}\right) \oplus a_{0,m-1}$ 和 $S_{m-1}^{(2)} = \left(\bigoplus_{t=0}^{m-2} a_{t,m+1}\right) \oplus a_{1,m-1}$ 计算而得。

为了下步更好描述校验算子迭代过程，图 2 给出校验算子几何描述图。从图 2 校验算子的几何结构可以看出，每个校验算子只含有第 0，第 2，第 3 列的 1 个未知数据单元，并且每个校验算子含有 2 或 3 未知数据单元。

第2步　第1步的校验算子可以看作方程组，方程的变量为失效的数据单元。则可以通过式(7)或者式(8)变换，得到只含有第 j 列失效数据单元的变换式子，并且每个变换式子至多只有两个未知数据单元，因此可以依次把第 j 列的源数据单元求解出来。

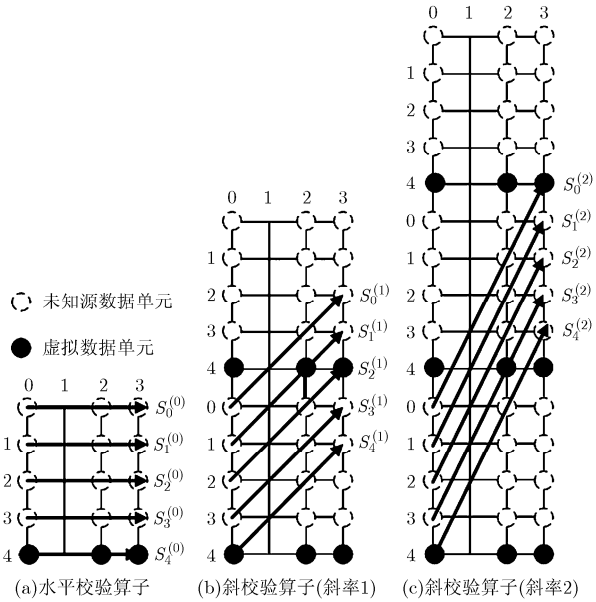

图 2 3 列数据失效校验算子几何结构

假设 $0 \le u \le m-1$, $\beta_1 = j-i$, $\beta_2 = k-j$ ，则一定存在 l_d , l_h ($1 \le l_d, l_h < m$)，若 $\beta_1 \le \beta_2$ 满足 $\langle \beta_2 - l_d\beta_1 \rangle_m = 0$, $\langle \beta_2 + l_h\beta_1 \rangle_m = 0$ ，则有式(7)，式(8)成立。

$$a_{u,j} \oplus a_{\langle u+2\beta_2\rangle_m,j} = \sum_{v=0}^{l_d-1} \Big(S^{(1)}_{\langle u+j+v\beta_1\rangle_m} \oplus S^{(2)}_{\langle u+k+j+v\beta_1\rangle_m} \oplus S^{(0)}_{\langle u+j-i+v\beta_1\rangle_m} \oplus S^{(1)}_{\langle u+k+j-i+v\beta_1\rangle_m} \Big) \tag{7}$$

$$a_{u,j} \oplus a_{\langle u+2\beta_2\rangle_m,j} = \sum_{v=0}^{l_h-1} \Big(S^{(1)}_{\langle u+j-v\beta_1\rangle_m} \oplus S^{(2)}_{\langle u+k+j-v\beta_1\rangle_m} \oplus S^{(0)}_{\langle u+j-i-v\beta_1\rangle_m} \oplus S^{(1)}_{\langle u+k+j-i-v\beta_1\rangle_m} \Big) \tag{8}$$

若 $l_h \le l_d$ ，则根据式(8)恢复第 j 列的源数据单元，否则式(7)恢复第 j 列的源数据单元。

若 $\beta_1 > \beta_2$ 满足 $\langle \beta_1 - l_d\beta_2 \rangle_m = 0$, $\langle \beta_1 + l_h\beta_2 \rangle_m = 0$ ，则式(9)，式(10)成立。

$$a_{u,j} \oplus a_{\langle u+2\beta_1\rangle_m,j} = \sum_{v=0}^{l_d-1} \Big(S^{(1)}_{\langle u+j+v\beta_2\rangle_m} \oplus S^{(2)}_{\langle u+k+j+v\beta_2\rangle_m} \oplus S^{(0)}_{\langle u+j-i+v\beta_2\rangle_m} \oplus S^{(1)}_{\langle u+k+j-i+v\beta_2\rangle_m} \Big) \tag{9}$$

$$a_{u,j} \oplus a_{\langle u+2\beta_1\rangle_m,j} = \sum_{v=0}^{l_h-1} \Big(S^{(1)}_{\langle u+j-v\beta_2\rangle_m} \oplus S^{(2)}_{\langle u+k+j-v\beta_2\rangle_m} \oplus S^{(0)}_{\langle u+j-i-v\beta_2\rangle_m} \oplus S^{(1)}_{\langle u+k+j-i-v\beta_2\rangle_m} \Big) \tag{10}$$

若 $l_h \le l_d$ ，则根据式(10)恢复第 j 列的源数据单元，否则根据式(9)恢复第 j 列的源数据单元。

式(7)-式(10)的证明方法完全一样，本文仅给出式(7)的详细证明。

证明　根据式(4)-式(6)校验子计算公式，当 $0 \le u \le m-1$ ，可得

$$S^{(1)}_{\langle u+j\rangle_m} = a_{\langle u+\beta_1\rangle_m,i} \oplus a_{u,j} \oplus a_{\langle u-\beta_2\rangle,k} \tag{11}$$

$$S^{(2)}_{\langle u+k+j\rangle_m} = a_{\langle u+2\beta_1+\beta_2\rangle_m,i} \oplus a_{\langle u+\beta_2\rangle_m,j} \oplus a_{\langle u-\beta_2\rangle_m,k} \tag{12}$$

$$S^{(0)}_{\langle u+j-i\rangle_m} = a_{\langle u+\beta_1\rangle_m,i} \oplus a_{\langle u+\beta_1\rangle_m,j} \oplus a_{\langle u+\beta_1\rangle_m,k} \tag{13}$$

$$S^{(1)}_{\langle u+k+j-i\rangle_m} = a_{\langle u+2\beta_1+\beta_2\rangle_m,i} \oplus a_{\langle u+\beta_1+\beta_2\rangle_m,j} \oplus a_{\langle u+\beta_1\rangle_m,k} \tag{14}$$

因此由式(11)-式(14)可得只含 j 列的数据单元式(15)。

$$a_{u,j} \oplus a_{\langle u+\beta_1\rangle_m,j} \oplus a_{\langle u+\beta_2\rangle_m,j} \oplus a_{\langle u+\beta_2+\beta_1\rangle_m,j} = S^{(1)}_{\langle u+j\rangle_m} \oplus S^{(2)}_{\langle u+k+j\rangle_m} \oplus S^{(0)}_{\langle u+j-i\rangle_m} \oplus S^{(1)}_{\langle u+k+j-i\rangle_m} \tag{15}$$

因此

$$\begin{aligned}
&\sum_{v=0}^{l_d-1}\Big(S^{(1)}_{\langle u+j+v\beta_1\rangle_m}\oplus S^{(2)}_{\langle u+k+j+v\beta_1\rangle_m}\oplus S^{(0)}_{\langle u+j-i+v\beta_1\rangle_m}\\
&\quad\oplus S^{(1)}_{\langle u+k+j-i+v\beta_1\rangle_m}\Big)\\
&=a_{u,j}\oplus \not{a}_{\langle u+\beta_1\rangle_m,j}\oplus\cdots\\
&\quad\oplus \not{a}_{\langle u+(l_d-2)\beta_1\rangle_m,j}\oplus \not{a}_{\langle u+(l_d-1)\beta_1\rangle_m,j}\\
&\quad\oplus \not{a}_{\langle u+\beta_1\rangle_m,j}\oplus \not{a}_{\langle u+2\beta_1\rangle_m,j}\oplus\cdots\\
&\quad\oplus \not{a}_{\langle u+(l_d-1)\beta_1\rangle_m,j}\oplus a_{\langle u+l_d\beta_1\rangle_m,j}\\
&\quad\oplus a_{\langle u+\beta_2\rangle_m,j}\oplus \not{a}_{\langle u+\beta_2+\beta_1\rangle_m,j}\oplus\cdots\\
&\quad\oplus \not{a}_{\langle u+\beta_2+(l_d-2)\beta_1\rangle_m,j}\oplus \not{a}_{\langle u+\beta_2+(l_d-1)\beta_1\rangle_m,j}\\
&\quad\oplus \not{a}_{\langle u+\beta_1+\beta_2\rangle_m,j}\oplus \not{a}_{\langle u+\beta_2+2\beta_1\rangle_m,j}\oplus\cdots\\
&\quad\oplus \not{a}_{\langle u+\beta_2+(l_d-1)\beta_1\rangle_m,j}\oplus a_{\langle u+\beta_2+l_d\beta_1\rangle_m,j}
\end{aligned}\tag{16}$$

其中 $\not{a}_{\langle x\rangle_m,j}$ 表示 $a_{\langle x\rangle_m,j}$ 数据单元异或消除。因为 $\langle\beta_2-l_d\beta_1\rangle_m=0$，所以 $a_{\langle u+\beta_2\rangle_m,j}=a_{\langle u+l_d\beta_1\rangle_m,j}$，因此

$$\begin{aligned}
a_{u,j}\oplus a_{\langle u+2\beta_2\rangle_m,j}=\sum_{v=0}^{l_d-1}\Big(&S^{(1)}_{\langle u+j+v\beta_1\rangle_m}\oplus S^{(2)}_{\langle u+k+j+v\beta_1\rangle_m}\\
&\oplus S^{(0)}_{\langle u+j-i+v\beta_1\rangle_m}\oplus S^{(1)}_{\langle u+k+j-i+v\beta_1\rangle_m}\Big)
\end{aligned}\tag{17}$$

由于 $1\le\beta_2=k-j\le m-1$，因此 $u\ne\langle u+2\beta_2\rangle_m$，$a_{u,j}$ 与 $a_{\langle u+2\beta_2\rangle_m,j}$ 肯定是第 j 列中两个不同未知数据单元。 证毕

根据从式(7)-式(10)中选择的式子求解得到只含第 j 列至多2个未知数据单元的方程。因 $a_{m-1,j}=0$，令 $u=m-1$，得到含有 $a_{m-1,j}$ 和 $a_{\langle 2\beta_2-1\rangle_m,j}$ 两未知数据单元的方程，作为译码起始方程求解出 $a_{\langle 2\beta_2-1\rangle_m,j}$，然后依次求解出第 j 列源数据单元。再根据RDP双容错译码算法，依次恢复出其余2列的失效数据。

实例 假若 $m=5$，失效源数据列 $i=0$，$j=2$，$k=3$，第 j 列源数据单元译码过程如图3所示。从图3可知，通过4个校验算子进行两轮迭代运算，最后得到只含有 j 列的源数据单元的式子。

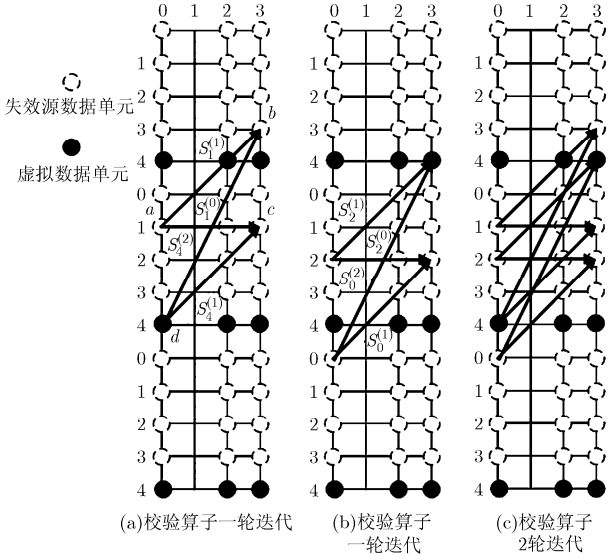

图3 恢复中间失效数据几何示意图

图3(a)中 $a\to b\to c\to d$ 对应的校验算子迭代为 $a_{4,2}\oplus a_{0,2}\oplus a_{1,2}\oplus a_{2,2}=S^{(1)}_1\oplus S^{(0)}_1\oplus S^{(2)}_4\oplus S^{(1)}_4$；

图3(b)中 $A\to B\to C\to D$ 对应校验算子迭代为 $a_{0,2}\oplus a_{1,2}\oplus a_{2,2}\oplus a_{3,2}=S^{(1)}_2\oplus S^{(0)}_2\oplus S^{(2)}_0\oplus S^{(1)}_0$；

则图3(c)对应的是图3(a)与图3(b)的迭代，得到只含第2列2个未知数据单元的变换式为

$$\begin{aligned}
a_{4,2}\oplus a_{3,2}=&S^{(1)}_1\oplus S^{(0)}_1\oplus S^{(2)}_4\oplus S^{(1)}_4\\
&\oplus S^{(1)}_2\oplus S^{(0)}_2\oplus S^{(2)}_0\oplus S^{(1)}_0
\end{aligned}\tag{18}$$

又因为已知 $a_{4,2}$ 为虚拟数据单元，因此可以依次求出 $a_{3,2}\to a_{2,2}\to a_{1,2}\to a_{0,2}$。

第3步 通过第2步，把第 j 列的失效源数据单元恢复之后，只有2列源数据失效，则可采用RDP码双容错译码算法恢复其余2列失效源数据列。

定理 1 E-RDP 编码能够纠正任意小于或等于 3 列的数据丢失，即具有 MDS 特性。

证明 E-RDP 能够纠任意 3 列数据失效，又知 E-RDP 是 RDP 扩展，RDP 能够纠任意 2 列数据失效，因此可知 E-RDP 码能够纠任意小于等于 3 列的数据失效。又因为 E-RDP 码只有 3 列数据校验，因此根据编码理论可知，E-RDP 具有 MDS 特性。

证毕

4 E-RDP 码性能分析

基于纠删码的数据布局方案，纠删码空间利用率、编译码效率以及读取数据的小写能力是衡量一类编码数据分布策略性能的重要指标，直接决定方案的修复成本。为了分析 E-RDP 码的性能，假定每个数据单位为 1 bit。

4.1 空间利用率和纠删能力

数据布局的空间利用率是指校验数据与源信息数据之间的比例。若有 n 个存储节点，容许 k 个节点故障，其空间利用率至少为 $(n-k)/n$。E-RDP 码的 $m-1$ 列为源数据列，3 列为校验列，其空间利用率为 $(m-1)/(m+2)$。并且 E-RDP 码具有 MDS 特性，能容许任意 3 个存储节点失败，根据编码理论中 Singleton bound 定理[19]可知，空间利用率和纠删能力达到了最优。

4.2 编译码效率

E-RDP 码是一类阵列码，其编译码运算只需异或运算，因此编码效率定义为 1 bit 源数据需要的异或次数，即编码过程中总异或次数与源数据总 bit 数之比。根据 E-RDP 的 3 列校验列编码过程可知，水平校验列中每个校验单元需要 $(m-2)$ 次异或运算，而水平校验列总共有 $m-1$ 个校验单元，因此水

平校验列编码过程总共需要 $(m-2)(m-1)$ 次异或操作。根据式(2)斜率为 1 的校验列的每个校验单元也需要 $(m-2)$ 次异或运算，因此总异或次数总共需要 $(m-2)(m-1)$ 次异或操作。而斜率为 2 的校验列与斜率为 1 的校验列需要异或次数完全一样，因此 3 列校验列总共需要 $3(m-2)(m-1)$ 次异或运算。而 E-RDP 码有 $(m-1)$ 列源数据，每列数据有 $(m-1)$ 数据单元，因此编码效率 1 bit 源数据需要参与 $3-1/m$ 次异或运算，根据编码理论中 Singleton bound 定理[19]，达到 3 容错数据布局方案最优值。

根据第 3 节 E-RDP 码的译码算法可知，E-RDP 码译码算法比编码复杂很多。这里只考虑最复杂的失效 3 列数据都是源数据列的译码效率。

根据 3.1 节的译码算法，可知第 1 步中计算 $S^{(0)}$ 需要异或次数为 $(m-1)(m-4)$，斜列校验算子 $S^{(1)}$, $S^{(2)}$ 需要异或次数为 $2(m-1)(m-3)+2(m-2)$，因此计算校验算子总异或次数为 $(3m-8)(m-1)-2$。

第 2 步中恢复第 j 列需要的异或运算次数为 $3(l-1)(m-1)+(m-2)$，其中 l 为 l_d, l_h 中较小的数，值范围为小于等于 $(m-1)/2$。为了方便比较，l 取平均值为 $(m-1)/4$。

第 3 步恢复其余失效数据列需要的异或次数为 $4(m-2)-1$。译码过程中总异或操作次数为 $(3.75m-2.75)(m-1)-8$。

将 E-RDP 码与 RS 码，STAR 码，EEOD 码纠 3 错的译码性能进行比较。译码效率定义为恢复失效数据列所需的总异或次数与所有源数据比特数之比，即每 bit 译码所需要的异或次数。由文献[17]可知 STAR 码总的异或次数为大约为 $(3m+2l_d+l_h)\cdot(m-1)$，EEOD 码译码异或次数大约为[18] $(4l_d-2+3m)(m-1)-3$，Bloemer 码[19]是一类 RS 码，其译码过程需要的总异或次数为 krL^2，以及 r^2 的有限域操作（其中 k, r 分别表示码的源数据位和校验位的长度，L 表示有限域的大小 $\mathrm{GF}(2^L)$）。为了比较方便，忽略有限域的操作 r^2（实际有限域的计算对译码的速度有影响），而源数据位的总 bit 数为 kL。

从表 1 可以看出。存储介质数比较小时，STAR 码，EEOD 码译码过程中每 bit 源数据需要的异或次数逐渐增大，接近 4，而 E-RDP 码译码中每 bit 所需异或次数则超过 4，因此 E-RDP 码在存储介质数比较小时，译码效率不如 STAR 码和 EEOD 码。但随着存储介质的增大，E-RDP 码译码次数逐渐减小，译码效率优于 STAR 码和 EEOD 码。而 RS 码则随着磁盘数的增大，只与有限域的大小相关，计算复杂度最高。

表1 4类纠删码每bit译码所需异或次数

磁盘数	EEOD 码	STAR 码	E-RDP 码	RS 码
5	3.620	3.550	4.667	9.000
7	3.680	3.714	4.333	9.000
11	3.810	3.836	4.189	12.000
13	3.846	3.865	4.121	12.000
17	3.882	3.901	4.033	15.000
19	3.894	3.912	4.003	15.000
23	3.850	3.929	3.959	15.000
29	3.931	3.945	3.915	15.000
31	3.935	3.949	3.901	15.000

4.3 小写性能和平衡特性的分析

影响存储系统性能的另外两个重要参数为小写性能和存储节点数据读取平衡性。存储介质小写(small writes)定义为一次输入数据远远小于(或等于)一个数据单元。当对源数据单元小写，此源数据单元影响的校验单元需要修改，会带来额外的开销降低系统的吞吐量，影响存储系统的 I/O 性能。因此在保证系统可靠性前提下，要求所采用的数据容错技术小写额外开销尽可能小和均衡。

假设 3 容错 STAR 码，EEOD 码中源数据列为 m，当更新的数据单元没有参与调节因子的计算，则只需更新水平方向和 2 斜线方向的校验单元，即一次小写需要 4 次 RMW(Read Modify Write)操作。而更新的数据单元参与调节因子计算时，则需要更新与调节因子相关校验单元，需要 $m+1$ 次 RMW 操作。不参与调节因子计算的数据单元总共有 $(m-2)(m-1)$，因此总共需要 $4(m-2)(m-1)$ 次 RMW 操作。作为调节因子的数据单元为 $2(m-1)$，则需要 $2(m-1)(m+1)$ 次 RMW。因此 EEOD 码和 STAR 码每个数据单元平均需要的小写操作为 $6-1/m$ 次。

而 E-RDP 码中，若源数据单元涉及水平校验列计算，源数据单元 1 次小写需要 6 次 RMW，若不涉及水平校验列计算，则只需要 4 次 RMW。根据 2.1 节可知，涉及水平校验列计算的源数据单元总共有 $(m-1)^2-4(m-2)$ 个，不涉及水平校验列计算的数据单元有 $4(m-2)$ 个，因此总的小写操作为 $6(m-1)^2-16(m-2)$ 次，每个数据单元平均需要的小写操作为 $6-8(m-2)/(m-1)^2$ 次。

随着 m 值的增大，E-RDP 码，STAR 码，EEOD 码的每个数据单元其平均小写操作大约为 6 次 RMW。但是在 STAR 码和 EEOD 码中涉及调节因子的源数据单元，其小写次数为 $m+1$ 次，不涉及的是 4 次，容易造成小写操作不均衡，影响存储系统的 I/O 性能。而 E-RDP 码数据单元小写次数为 6

或者 4，将小写负载操作均衡到每个数据单元。因此小写操作时，不需要集中对 3 校验列数据单元进行频繁读写操作，而是分散到每个数据单元上，有利于解决存储系统 I/O 问题，不会由于小写产生额外瓶颈。

5 结论

为了保证存储系统数据的高可靠性和高可用性，本文在 RDP 码的基础上进行扩展，给出一种具有几何结构的数据布局方案。E-RDP 码是一类 3 容错 MDS 阵列码，不但可以容许任意 3 个设备失效，并且纠删能力与冗余量达到了 3 容错码的最优值，其保留了 RDP 小写性能好的特点，并且编译码复杂度和更新复杂度都相对较低，结构简单易实现，为纠删码在存储系统的研究提供了理论依据。

参 考 文 献

[1] 李国杰. 大数据研究的科学价值[J]. 中国计算机学会通讯, 2012, 8(9): 8–15.
Li Guo-jie. The scientific value of big data research[J]. *China Computer Federation Communication*, 2012, 8(9): 8–15.

[2] 王意洁, 孙伟东, 周松, 等. 云计算环境下分布存储关键技术[J]. 软件学报, 2012, 23(4): 962–986.
Wang Yi-jie, Sun Wei-dong, Zhou Song, *et al*.. Key technologyies of distributed storage for cloud computing[J]. *Journal of Software*, 2012, 23(4): 962–986.

[3] 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.
Luo Xiang-hong and Shu Ji-wu. Summary of research for erasure code in storage system[J]. *Journal of Compute Research and Development*, 2012, 49(1): 1–11.

[4] 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.

[5] Xu L and Bruck J. X-code: MDS array codes with optimal encoding[J]. *IEEE Transactions on Information Theory*, 1999, 45(1): 272–276.

[6] Xu L, Bohossian V, Bruck J, *et al*.. Low density MDS codes and factors of complete graphs[J]. *IEEE Transactions on Information Theory*, 1999, 45(6): 1817–1826.

[7] Li Ming-qiang and Shu Ji-wu. C-Codes: cyclic lowest-density MDS array codes constructed using starters or RAID 6[OL]. http://arxiv.org/abs/1104.2547, 2011.10.

[8] Corbett P, English B, Goel A, *et al*.. Row diagonal parity for double disk failure[C]. Proceedings of the Third USENIX Conference on File and Storage Technologies, Berkeley, CA, USA, 2004: 1–14.

[9] Wu Chen-tao, Wan Sheng-gang, He Xu-bin, *et al*.. H-Code: a hybrid MDS array code to optimize partial stripe writes in RAID-6[C]. Proceedings of the IEEE Congress on International Parallel&Distributed Processing Symposium, Alaska USA, 2011: 782–793.

[10] Wu Chen-tao, He Xu-bin, Wu Guan-ying, *et al*.. HDP code: a horizontal-diagonal parity code to optimize I/O load balancing in RAID-6[C]. Proceedings of the IEEE/IFIP 41st International Conference on Dependable System & Network (DSN), HongKong, 2011: 209–220.

[11] Li M, Shu J, and Zheng W. GRID codes: strip-based erasure code with high fault tolerance for storage systems[J]. *ACM Transactions on Storage*, 2009, 4(4): 1–22.

[12] Hafner J L. HoVer erasure codes for disk arrays[C]. Proceedings of DSN-06: International Conference on Dependable Systems and Networks, Philadelphia, 2006: 217–226.

[13] Hafner J L. WEAVER codes: highly fault tolerant erasure codes for storage systems[C]. Proceedings of FAST-2005: 4th Usenix Conference on File and Storage Technologies, San Francisco, 2005: 211–224.

[14] Tamo I, Wang Zhi-ying, and Bruck J. Zigzag codes: MDS array codes with optimal rebuilding[J]. *IEEE Transactions on Information Theory*, 2013, 59(3): 1597–1616.

[15] Blaum M, Bruck J, and Vardy A. MDS array codes with independent parity symbols[J]. *IEEE Transactions on Information Theory*, 1996, 42(2): 529–542.

[16] Sheng L, Gang W, Stones D S, *et al*.. T-code: 3 erasure longest lowest-density MDS codes[J]. *IEEE Journal on Selected Areas in Communications*, 2010, 28(2): 289–296.

[17] Huang C and Xu L. STAR: an efficient coding scheme for correcting triple storage node failures[C]. Proceedings of the 4th USENIX Conference on File and Storage Technologies, Berkeley, CA, 2005: 197–210.

[18] 万武南, 吴震, 等. RAID-EEOD: 一种基于 3 容错阵列码 RAID 数据布局研究[J]. 计算机学报, 2007, 30(10): 1–10.
Wan Wu-nan, Wu Zheng, *et al*.. RAID-EEOD: the study of data placement based on toleration on triple failures array codes in RAID[J]. *Chinese Journal of Computers*, 2007, 30(10): 1–10.

[19] Bloemer J M, Kalfane M, Karpinski R, *et al*.. An XOR-based erasure-resilient coding scheme[R]. Report of International Computer Science Institute, Berkeley, CA, August 1995.

万武南： 女，1978 年生，博士，副教授，主要研究方向为安全存储、编码理论.

王 拓： 男，1989 年生，硕士生，研究方向为安全存储、密码算法.

索 望： 男，1978 年生，博士，讲师，主要研究方向为安全存储、网络安全.