A New Erasure Code Decoding Algorithm

Di Fan, Feng Xiao, and Dan Tang

(Corresponding author: Dan Tang)

SoftwareEngineering College, Chengdu University of Information Technology
No. 24, Xue-Fu Road, Chengdu 610225, China
(Email: tangdan@foxmail.com)

(Received Nov. 23, 2017; Revised and Accepted June 18, 2018; First Online Feb. 24, 2019)

Abstract

Erasure code, as a fault-tolerant technology which is widely used in storage and communication fields. It can reduce the storage space consumption and provide the fault-tolerant capability of the replication backup. Therefore, a simple and efficient erasure coding and decoding algorithm is also paid much attention. Deenadhayalan has proposed a matrix methods for lost data reconstruction for erasure code that uses the pseudo-inverse principle to recover a random raw data element and apply to any erasure code, but cannot recover both the data element and the redundant element at the same time. After the data elements are recovered, the redundant elements can recovered by encoding, which increases the computational complexity. In response to this situation, this paper presents an improved decoding algorithm suitable for any theoretically recoverable cases. The algorithm is an erasure code decoding algorithm based on matrix that can reconstruct data elements and redundant elements at the same time. It also has high practicability and simple,easy algorithm steps, is easy to implement and has a wide range of applications. Through the simulation experiment it can be concluded that the efficiency is also high.

Keywords: Erasure Codes; Improved Decoding Algorithm; Matrix Decoding; Theory Can be Restored

1 Introduction

With the rapid development of computer technology, information technology has been widely popularized in various industries and fields. The data were explosively growing, and the demand for the storage system [6, 12, 13] is getting higher and higher. With the increasing storage demand, both the number of storage nodes and the capacity of single node in storage system are increasing exponentially, which means that the probability of node failure and the failure of sectors in single node are larger than before, so data fault-tolerant is an indispensable key technology in storage system.

The most widely used fault tolerance technology is multi copy replication technology, that is, fault-tolerance by replica copy. The other is erasure code technology, through the encoding of fault-tolerance. Erasure code technology [7, 10] mainly relies on the erasure code algorithm [14] to store the original data after obtaining redundant elements, so as to achieve the purpose of fault tolerance. In the storage system, its main idea is to encode the original data element of the k block to obtain the m block redundancy element, and when there is a m block element failures, the lost element can be recovered through the remaining elements by using a certain decoding algorithm. Compared with multi-rseplica fault-tolerant technology, erasure code fault-tolerant technology can reduce the storage space significantly while providing the same or even higher data fault tolerance.

In recent years, most of the research on erasure code is focused on the encoding process [8], and the decoding process is rarely involved. The decoding process of the original erasure code is processed by cyclic iteration or matrix inversion. Each code has different decoding algorithm. And the original decoding type is node loss, when an element or sector is lost in a node, the entire node is considered invalid. However, as the amount of data is increasing and the number of hardware is increasing, there are more and more failures of sectors in one node. When the whole node is rebuilt, those sectors that are not needed to be rebuilt are also rebuilt, thus causing repetition and increasing unnecessary computation. Therefore, the restoration of random elements or sector losses has also become an important problem in erasure code decoding.

In [9], an algorithm for merging and decoding in binary domain (hereinafter referred to as merger decoding) is proposed. This algorithm restores the node error by rebuilding the data block on the fault-tolerant storage system and and can be used to restore the loss of the random element. However, the calculation of this algorithm involves the calculation of the inverse matrix. Therefore, when the single error is restored, the efficiency will be higher. Once there are many errors, the operation of inversion will greatly affect the speed of operation, thus affecting the decoding efficiency.

In [3], Deenadhayalan proposed a decoding algorithm for erasure codes. This algorithm is based on generator matrix and its pseudo-inverse matrix (hereinafter referred to as matrix decoding), and is generally declared as two results for lost data sectors. One is that the algorithm is recoverable, which is theoretically recoverable, when the algorithm provides a formula made up of readable data to restore the lost sector, the other is an unrecoverable sector in theory. The matrix decoding algorithm not only solves the problem of recovery of random sector loss, but also abandons the calculation of the inverse matrix to make it highly efficient. At the same time, it is also a universal decoding algorithm that is applicable to any array code and can also be used for non-XOR erasure code, but most suitable for the array code. Therefore, this paper describes the array code as an example. However, the matrix decoding algorithm has a disadvantage at the same time. The loss of redundant elements can only be solved by the encoding algorithm after the data elements are recovered, but cannot recover the data elements and redundant elements at the same time. In this paper, an improved algorithm which can restore random lost sector including redundant sector is proposed for the problem of matrix decoding. It reduces the complexity of algorithm from the algorithm level, and can restore any loss that can be recovered theoretically, and the efficiency has been improved through experiments.

Here is a description of the organizational structure of this paper. The second part introduces some basic theories and principles implicated in the algorithm. The third part presents the example of the original matrix decoding algorithm and the improved algorithm steps, and gives concrete examples. The fourth part analyzes the experimental data of the algorithm, and compared with the other decoding algorithm by performance. The fifth part gives the summary of this paper.

2 Basic Concepts and Principles

In order to better describe the algorithm and get a clearer understanding of the algorithm, this section will introduce some basic concepts and principles involved in the paper.

2.1 Basic Concepts

There is no consistent definition of erasure-tolerant technology to erasing codes in storage systems. In order to facilitate the description and understanding of this paper, based on the literature [3, 4], the relevant concepts commonly used in this paper are as follows.

- Data (or information): The original piece of data string used to store the information needed by real users.
- Parity (or redundant): By using the erasure code algorithm, a data string of redundant information obtained by calculating the data, the existence of these redundancies is to ensure the erasure code's fault-tolerance.
- Element (or symbol): A fundamental unit of data or parity; this is the building block of the erasure code. In the process of erasure code calculation, an element is usually regarded as a basic unit of computation.
- Stripe: Collection of all information independently related to the same erasure algorithm. A storage system can be regarded as a collection of multiple stripes. The stripe is a set of information that independently constitutes an erasure code algorithm.
- Strip: A unit of storage consisting of all continguous elements (data, parity or both) from the same disk and stripe. In coding theory,this is associated with a code symbol. It is sometimes called a stripe unit.The set of strips in code instance form a stripe.Typically,the strips are all of the same size(contain the same number of elements). A collection of data belonging to the same stripe on the same disk. The size of a strip is determined by the number of elements contained in the strip.

The symbols and descriptions of some principles are described in Table 1.

Table 1: Symbols

Symbols	Size	Explain
G	$R \times C$	Generate matrix
H	$(R-C) \times R$	Check matrix
U	$C \times R$	Left pseudo-inverse
O_R	$C \times C$	Partial unit matrix
$d_{i,j}$	—	The i element of the j strip in the disk array
D	$C \times 1$	Raw data
T	$R \times 1$	After encoding the element data
H'	$(R-C) \times R$	Redundancy matrix

2.2 Basic Principles and Proofs

The basic principle of the improved algorithm in this paper is divided into two parts: the pseudo-inverse matrix U used to recover the data elements and the redundancy matrix H' used to recover the redundant elements. Then we describe separately and give the proof at the same time.

2.2.1 Pseudo-Inverse Matrix U

First describe some of the basic theories about linear algebra in binary.

Definition 1.

(Left pseudo-inverse): If matrix A is left multiplied by matrix B to get the unit matrix, then B is called the left pseudo-inverse matrix of A. When the matrix is full rank and $R \leq C$, the left pseudo-inverse matrix must exist.

(Null space): Null space refers to a set of all vectors orthogonal to each row vector of the matrix. Null Space Base refers to the largest set of linearly independent vectors in null space.

Suppose that G is a $R \times C$ matrix, and $R \leq C$.If B is the null space of G, U is the left pseudo-inverse matrix of G, X varies over all binary $C \times (R - C)$ matrices, then get Equation (1):

$$(U + (X \cdot B)) \cdot G = O_R. \tag{1}$$

$U + (X \cdot B)$ runs over all partial pseudo-inverses, X is to add a null space vector for each column of U. There are two important equations in the coding theory, $G \times D = T$ and $H \times T = 0$ respectively. This can be introduced Equation (2):

$$H \times G \times D == 0. \tag{2}$$

It can be seen from Equation (2) that H is a zero-space basis of G, so that the pseudo-inverse matrix of the generated matrix can be found using the check matrix.

Theorem 1. *The left pseudo-inverse matrix obtained by the improved algorithm, in which any theoretically recoverable data element corresponds to a non-zero row of the pseudo-inverse matrix, and the non-zero positions in these non-zero row indicate which data elements and redundant elements whose XOR is a data element. A directly readable element corresponds to a labeled row in the pseudo-inverse matrix (ie, a row vector containing only one.) An unrecoverable data element corresponds to an all-zero row of pseudo-inverse matrix.*

Proof. Suppose that T represents the vector containing all the elements after encoding, T' represents the lost coding vector, that is, the lost element corresponding position is 0, and obviously get Equation (3)

$$G' \times D = T' \tag{3}$$

□

In Equation (3), the missing element corresponds to all-zero rows in G'. So,here has Equation (4)

$$U \cdot T' = U \cdot G' \cdot D = O_R \cdot D = D' \tag{4}$$

Where the 0 element of D' corresponds to a zero position on the diagonal of O_R, and O_R corresponds to all-zero rows in the pseudo-inverse matrix. Non-zero position on the diagonal of O_R corresponds to the non-zero position in D', while the non-zero position on the diagonal of O_R corresponding to non-zero row of pseudo-inverse matrix U. Thus can be see each row of the pseudo-inverse matrix U corresponds to each of the elements of D',that is the data element. In the meantime, since $U \cdot T' = D'$, therefore, each row in the pseudo-inverse matrix U, each bit corresponds to one element in T, so that each row corresponds to one data element and each bit corresponds to one element.

2.2.2 Check Matrix H and Redundancy Matrix H'

Check matrix is a very important concept in coding theory.This section describes the concepts of check matrices,redundancy matrices and the theory of recovery parity elements. In this paper, we referred to the matrices transformed by the improved parity check matrix as redundancy matrices.

Check matrix is a matrix used to check whether a codeword is correct. Each column represents an element location. Each row represents a redundant element and is also an equation (the result is 0 after XOR for all non-zero positions in each row). For example, Equation (5) is the check matrix of STAR (3,6) code.

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

Theorem 2. *The redundancy matrix obtained by the improved algorithm in this paper, whose non-zero rows represent a theoretically recoverable redundant element, the exclusive-OR of elements corresponding to each non-zero position in this row is the redundancy element corresponding to this row. Each of all-zero row represents a known readable redundant element.*

Proof. As described above for the check matrix, it can be clearly seen that the result of the exclusive-OR of each row in the check matrix is 0, so the result of the XOR between the row and the row is also 0, and the XOR transformations between rows and rows does not affect the property of the check matrix. Suppose that the redundant elements of STAR (3, 6) code are $(P_0, P_1|Q_{0,0}, Q_{1,0}|Q_{0,1}, Q_{1,1})$, that is, each row represents a redundant element. If the second row is added to the first row and placed in the first line, the result of XOR for all the non-zero position elements is still 0. As in Equations (6), (1) and (3) add the same result to zero.

$$\begin{cases} d_{0,0} + d_{0,1} + d_{0,2} + P_0 = 0(1) \\ d_{0,0} + d_{0,1} + d_{0,2} = P_0(2) \\ d_{1,0} + d_{1,1} + d_{1,2} + P_1 = 0(3) \\ d_{1,0} + d_{1,1} + d_{1,2} = P_1(4) \end{cases} \tag{6}$$

□

On this basis, if zero redundant element corresponding to the row, the result of the difference of the remaining elements is equal to the redundant element, so that the

redundant element can be obtained. Use the above check matrix formula for example. The Equations (6) (1) and (3) two formula will be combined make P_0 into 0, then get the available Equation (7).

$$d_{0,0} + d_{0,1} + d_{0,2} + d_{1,0} + d_{1,1} + d_{1,2} + P_1 = P_0 \quad (7)$$

From this we can prove that the theory 2 is correct.With the above concepts and theories, the next section will describe our improved algorithm process based on these contents and give examples.

3 Decoding Algorithm

In Paper [3], a matrix based erasure decoding algorithm matrix decoding is described. In the literature, specific methods and steps are given for the reconstruction of EVENODD array code [11]. For the case of missing elements including redundant elements, the matrix decoding algorithm first recovers the missing data elements and then encodes the missing redundant elements. This section first describes the decoding algorithm of the matrix decoding algorithm, and gives an example. Next, the improved algorithm of the matrix decoding algorithm in this paper is introduced, and the algorithm steps are described in detail.

3.1 Matrix Decoding Algorithm

The matrix decoding algorithm is based on the matrix theory and the pseudo-inverse principle. The core of the algorithm is to construct the pseudo-inverse matrix of the generate matrix. When the pseudo-inverse matrix is constructed, each column of the pseudo-inverse matrix represents one data element, each non-zero position of a column represents a known-readable element. H in the original algorithm is a vertical matrix. So let's describe the structure of the pseudo-inverse matrix.

1) Construct a square matrix W of size $R \times R$, $W = (B|H)$,the initial B consists of a unit matrix and all-zero rows.Write all missing element positions to a uniform list L - lost list.

2) For each lost element in list L,let r indicate the lost element corresponding to the row of W, then:

   - Find any column b in H that has a one in row r. If none exists,Zero any column in B that has a one in row r and continue to the next lost element;
   - For each one in row r of W, say in column c,if $c \neq b$, sum and replace column b into column c.
   - Zero column b in H.

3) Use the resulting B to recover lost data elements.

Example 1. *The following uses STAR(3,6) as an example. The data encoded by the STAR code is arranged as Equation (8)*

$$T = (d_{0,0}, d_{1,0} | d_{0,1}, d_{1,1} | d_{0,2}, d_{1,2} | P_0, P_1 | Q_{0,0}, Q_{1,0} | Q_{0,1}, Q_{1,1}) \quad (8)$$

Suppose that the lost elements list L = (0, 2, 4, 5, 8, 9), then the data is arranged as T = (0, $d_{1,0}$|0, $d_{1,1}$|0, 0|P_0, P_1|0, 0|$Q_{0,1}$, $Q_{1,1}$). From the steps above can get the pseudo-inverse matrix as Equation (9).

$$U = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (9)$$

Each column in the pseudo-inverse matrix represents a primitive data element, and each non-zero position represents a known available data element. After the original data elements 0, 2, 4, and 5 are obtained therefrom, an encoding algorithm get 8,9 and multiply the original data element by the generator matrix, as in Equation (10).

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} d_{0,0} \\ d_{1,0} \\ d_{0,1} \\ d_{1,1} \\ d_{0,2} \\ d_{1,2} \end{pmatrix} = \begin{pmatrix} d_{0,0} \\ d_{1,0} \\ d_{0,1} \\ d_{1,1} \\ d_{0,2} \\ d_{1,2} \\ P_0 \\ P_1 \\ Q_{0,0} \\ Q_{1,0} \\ Q_{0,1} \\ Q_{1,1} \end{pmatrix} \quad (10)$$

3.2 Improved Decoding Algorithm

This section will describe the improved algorithm proposed in this paper, will give the algorithm specific steps and examples. The algorithm proposed in this paper is based on the improvement of the matrix decoding algorithm. But it can recover all the theoretically recoverable cases at the same time, including recover the original data elements and redundant data elements at the same time. It can be applied to any array code, and it can be extended to non-binary erasure codes, such as RS codes.

3.2.1 Improved Algorithm Steps

Algorithm steps are as follows:

1) Construct a square matrix A, $A = \binom{U}{H}$, $U = (I_c|0)$, H is check matrix. Is the lost elements list;

2) Judging whether the right half of the check matrix formed is a unit matrix or not, if not, transform matrix between rows and rows to make it a identify matrix;

3) The transformation of A is equivalent to the inversion process;

4) Get the converted A can recover the lost elements, a row represents a data element, in the row a non-zero position corresponding to a known data elements.

The following steps for the transformation of the specific steps:

1) For each data element s in the lost element list L, first determine whether the type of the data element belongs to the original data or to the redundant data. If s belongs to original data,then continue; if not,then skip. Loop through the data block element s in L;

2) Find h in H, its s column is 1. If none exist, the U in the s column has a line set to zero;

3) After found h, if L does not contain redundant elements, then select the most sparse row f from the found result h, if the redundant elements are included, remove the missing redundant elements from the found result after select the most sparse row f in h (in order to retain the value of the missing redundant element row, the last can be found at the same time);

4) For one in column s of A in row e, if e is not equal to f, add f(exclusive-or) to e and replace e;

5) Set the row f in H to zero;

6) After traversing the data block elements in L, set the redundant elements in L correspond to the columns in H to zero.

3.2.2 Examples of Improved Algorithm

In order to better understand and explain the above algorithm, the following is illustrated by taking STAR [5](3,6) and RDP [2](3,4) as examples.

Eg1. STAR code: The structure of the STAR code is shown in Table 2.

Table 2: STAR code structure

S_0	S_1	S_2	P	Q_0	Q_1
$d_{0,0}$	$d_{0,1}$	$d_{0,2}$	P_0	$Q_{0,0}$	$Q_{0,1}$
$d_{1,0}$	$d_{1,1}$	$d_{1,2}$	P_1	$Q_{1,0}$	$Q_{1,1}$

Expand the disk data, change to $D = (d_{0,0}, d_{1,0}|d_{0,1}, d_{1,1}|d_{0,2}, d_{1,2})$, then $T = (d_{0,0}, d_{1,0}|d_{0,1}, d_{1,1}|d_{0,2}, d_{1,2}|P_0, P_1|Q_{0,0}, Q_{1,0}|Q_{0,1}, Q_{1,1})$. Construct the square matrix as shown in Equation (11).

$$A=\left(\frac{U}{H}\right)=\begin{pmatrix}1&0&0&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0&0\\1&0&1&0&1&0&1&0&0&0&0&0\\0&1&0&1&0&1&0&1&0&0&0&0\\1&0&0&1&1&1&0&0&1&0&0&0\\0&1&1&1&1&0&0&0&0&1&0&0\\1&0&1&1&0&1&0&0&0&0&1&0\\0&1&1&0&1&1&0&0&0&0&0&1\end{pmatrix} \tag{11}$$

In order to facilitate the comparison with the original algorithm, we assume that the missing element is the same as the missing element in Section 3.1.Suppose that the lost elements list $L = (0, 2, 4, 5, 8, 9)$. The data in list L correspond to rows s of A.

First determine the check matrix, the right half of it is the unit matrix, then the following operation.

For column $s = 0$, find some row in H that has a one in this column,we can find $h = (6, 8, 10)$, but 8 is in list L, so we choose $h = 6$, because the row 6 is more sparse than the row 10, after select, add row 6 to row 0,8,10, then set row 6 to 0; For row $s = 2$, we can find $h = (8, 9, 11)$, but 8,9 is in list L, so we choose $h = 11$, after select,add row 11 to row 0,2,8,9, then set row 11 to 0; For column $s = 4$, find some row in H that has a one in column 4, we can find $h = (8, 10)$, but 8 is in list L,so we choose $h = 10$, after select, add row 10 to row 2,4,8, then set row 10 to 0; For column $s = 5$, we can choose $h = (7, 8, 9)$, but 8,9 is in list L,so we choose $h = 7$, after choose, add row 7 to row 0,4,5,8,9, set row 7 to 0;For row $s = 8, 9$, because 8,9 is in list L, so end the traverse, set the column 8,9 to 0.The final result becomes Equation (12):

$$A=\begin{pmatrix}0&0&0&1&0&0&1&1&0&0&0&1\\0&1&0&0&0&0&0&0&0&0&0&0\\0&1&0&1&0&0&1&0&0&0&1&1\\0&0&0&1&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&1&1&0&0&1&0\\0&1&0&1&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&1&0&0&1&1\\0&1&0&0&0&0&0&1&0&0&0&1\\0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&0&0\end{pmatrix} \tag{12}$$

A row with multiple non-zero positions in A is a theoretically solvable element, which results in Equation (13):

$$\begin{cases}d_{0,0}=d_{1,1}+P_0+P_1+Q_{1,1}\\d_{0,1}=d_{1,0}+d_{1,1}+P_0+Q_{1,0}+Q_{1,1}\\d_{0,2}=d_{1,0}+P_0+P_1+Q_{1,0}\\d_{1,2}=d_{1,0}+d_{1,1}+P_1\\Q_{0,0}=d_{1,1}+P_1+Q_{1,0}+Q_{1,1}\\Q_{0,1}=d_{1,0}+P_1+Q_{1,1}\end{cases} \tag{13}$$

Eg2. RDP code: The structure of the RDP code is shown in Table 3.

Table 3: RDP code structure

S_0	S_1	P	Q
$d_{0,0}$	$d_{0,1}$	P_0	Q_0
$d_{1,0}$	$d_{1,1}$	P_1	Q_1

Expand the disk data, change to $D = (d_{0,0}, d_{1,0}|d_{0,1}, d_{1,1})$, then $T = (d_{0,0}, d_{1,0}|d_{0,1}, d_{1,1}|P_0, P_1|Q_0, Q_1)$. Construct the square matrix as shown in Equation (14).

$$A = \left(\frac{U}{H}\right) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{14}$$

Suppose that the lost elements list $L = (0, 2, 4, 5)$. First determine the check matrix, the right half of it is the unit matrix, it can be clearly seen is not, so after some transformation between rows, the row 5 added to the row 6, we can make the right half of H into a unit matrix, the final transformation results is:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{15}$$

For column $s = 0$,find some row in H that has a one in this column, we can find $h = (4, 6)$, but 4 is in list L, so we choose $h = 6$, after select,add row 6 to row 0,8,10,then set row 6 to 0, the result matrix is:

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{16}$$

For column $s = 2$, find some row in H that has a one in this column, $h = (4, 7)$, but 4 is in list L, so choose $h = 7$, add row 7 to row 0,4, set row 7 to 0,the result matrix is :

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{17}$$

A row with multiple non-zero positions in A is a theoretically solvable element, which gives the following Equation (18):

$$\begin{cases} d_{0,0} = d_{1,0} + d_{1,1} + Q_0 \\ d_{0,1} = d_{1,0} + Q_1 \\ P_0 = d_{1,1} + Q_0 + Q_1 \\ P_1 = d_{1,0} + d_{1,1} \end{cases} \tag{18}$$

4 Analysis and Discussions

For a code system, the performance of coding system is usually evaluated from the encoding rate and the utilization ratio of space. For decoding, the decoding rate, recovery efficiency, and loss type of the decoding algorithm can be used for evaluation. This section will experimentally analyze this algorithm in terms of its applicability and decoding rate. The following is encoded and simulated data loss in the Python 3 environment supposing a folder represents a disk to emulate the raid.

Array code is a code system constructed only through XOR operations, and its own decoding algorithm uses cyclic iterative decoding. When an element in a strip is lost, it is considered the loss of the entire strip or even the entire disk. The entire disk is rebuilt upon recovery, and the original decoding of each array code is different. Deenadhaylan proposes an algorithm for restoring random data elements - matrix decoding, using the pseudo-inverse theory of the generating matrix to reconstruct the data elements, which is suitable for any erasure code, but can not restore the redundant elements at the same time. Tang proposed a merger decoding algorithm in paper [9], which reconstructed disk data elements by chunking and inverting the check matrix to recover both data elements and redundant elements. But this algorithm needs to compute the inverse matrix, which increases the computational complexity and the efficiency is not high. The improved decoding algorithm proposed in this paper is based on the improvement of the matrix decoding algorithm, and can recover any theoretically possible recovery, including the simultaneous restoration of data elements and redundant elements. Table 4 compares the three decoding algorithms and the cyclic iterative decoding algorithm for the decoding of the array code in terms of whether it can restore the random elements, whether it can restore data and redundant data and versatility at the same time.

It can be seen from Table 4 that the three algorithms of matrix decoding, merge decoding and improved decoding algorithm can recover random elements under the same fault-tolerant capacity. The merge decoding and the improved decoding algorithm can also recover the data and redundant elements. Compared with the original decoding algorithm only for one array code, the improved decoding algorithm and the merger decoding algorithm can be applied to any array code.

The following is the analysis of experimental data. For the matrix decoding algorithm, when the pseudo-inverse

Table 4: Properties of algorithm

Decoding algorithm	Whether the random elements can be recovered	Recover data and redundant elements at the same time	Versatility
Cyclic iterative	False	False	Bad
Matrix decoding	True	False	Good
Merger decoding	True	True	Good
Improved decoding	True	True	Good

matrix is taking, it needs to be encoded to obtain the redundant elements. Therefore, an extra matrix operation is added to the algorithm complexity of the improved algorithm in this paper. Taking EVENODD [1] for example, the matrix decoding algorithm requires 60 more XOR operations than the algorithm proposed in this paper, each adding 60 more operations. Even if the XOR operation is fast, the efficiency will still be affected. This paper simulates the data loss and compares efficiency by decoding different file sizes. The final experimental results are shown in Figure 1.

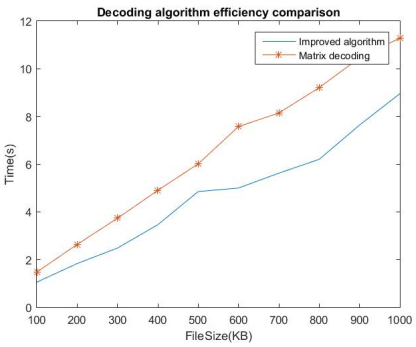

Figure 1: Efficiency comparison

It can be seen from the figure above that the improved decoding algorithm mentioned in this paper improves the time-consuming decoding algorithm of the original algorithm matrix, and it also improves the efficiency. Thus, the improved algorithm is efficient and simple.

For the two algorithms of merger decoding and improved decoding algorithm, we compare their efficiency by computing time and use EVENODD code to carry out experiments. Through the cyclic iteration, the merging decoding and the improved decoding algorithm, the experimental conditions of the data loss are hypothesized and the lost data are reconstructed gradually. Finally draw the experimental results as shown in Figure 2.

It can be seen clearly from the above figure that the

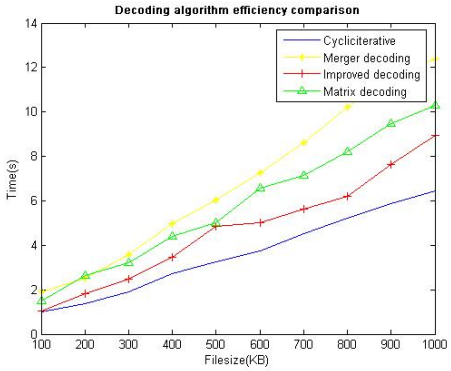

Figure 2: Four algorithm efficiency comparison

cyclic iterative decoding algorithm has the highest efficiency. The improved decoding algorithm takes longer than the cyclic iterative decoding algorithm, but the difference is not much. And because the merger decoding is used in the process of inversion calculation, it will be twice as much as the cyclic iterative decoding algorithm. It can be seen that the efficiency of the improved algorithm proposed in this paper is not bad.

Next, take RDP-code as an example to conduct double fault experiments. Four different algorithms are used respectively. Finally, the experimental diagram is shown as follows in Figure 3:

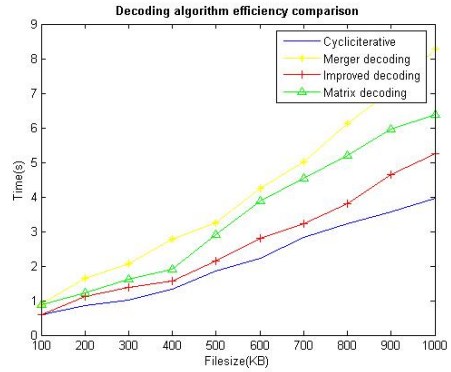

Figure 3: Four algorithm efficiency comparison

It can be seen that the above results are similar to those of EVENODD-code, so the algorithm proposed in this paper is more efficient.

5 Summary

In this paper, an improved erasure code decoding algorithm based on matrix decoding algorithm is proposed to recover the random sector loss of erasure codes. It is applicable to any theory recoverable situation, which includes the situation that the original algorithm can not be recovered, and it continues the good generality of the original algorithm. It is found that the efficiency of the algorithm is more efficient than the original one, and the calculation efficiency is higher, which can be widely used in the

random sector loss. This improved algorithm proposed in this paper is currently running on the binary matrix array code for the operation. After that, this algorithm can be extended to non-binary decoding operations, such as RS code.

Acknowledgments

The author thanked Pro.Dan Tang for the affirmation and guidance of the algorithm, the research of this paper can not be separated from Professor Tang's continuous encouragement and correction.This work was supported by the National Natural Science Foundation of China under Grant No. 61501064 and Science and technology program of SICHUAN 2018GZ0099.The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. Blaum, J. Brady, J. Bruck, and Jai Menon, "Evenodd: An efficient scheme for tolerating double disk failures in raid architectures," *IEEE Transactions on Computers*, vol. 22, vol. 2, pp. 245–254, 1994.

[2] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," in *Usenix Conference on File and Storage Technologies*, pp. 1, 2004.

[3] J. L. Hafner, V. Deenadhayalan, K. K. Rao, and J. A. Tomlin, "Matrix methods for lost data reconstruction in erasure codes," in *Conference on Usenix Conference on File and Storage Technologies*, pp. 14, 2005.

[4] J. L. Hafner, V. Deenadhayalan, T. Kanungo, and K. K. Rao, "Performance metrics for erasure codes in storage systems," *Chaos An Interdisciplinary Journal of Nonlinear Science*, vol. 18, vol. 3, pp. 150–156, 2004.

[5] C. Huang and L. Xu, "Star: An efficient coding scheme for correcting triple storage node failures," *IEEE Transactions on Computers*, vol. 57, vol. 7, pp. 889–901, 2008.

[6] M. S. Hwang, T. H. Sun, C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service," *Journal of Circuits, Systems, and Computers*, vol. 26, no. 5, 2017.

[7] E. Mugisha, G. Zhang, M. Zine, E. Abidine, and M. Eugene, "A TPM-based secure multi-cloud storage architecture grounded on erasure codes," *International Journal of Information Security and Privacy*, vol. 11, no. 1, pp. 52-64, 2017.

[8] J. Qureshi and A. Malik, "On optimization of wireless xor erasure codes," *Physical Communication*, vol. 27, no. 1, pp. 74–85, 2018.

[9] D. Tang, "Research of methods for lost data reconstruction in erasure codes over binary fields," *Journal of Electronics and Technology*, vol. 14, no. 1, pp. 43–48, 2010.

[10] P. Teng, L. Chen, D. Yuan, and X. Wang, "Sparse random erasure code: A large-scale data storage disaster recovery method," *Journal of Xi'an Jiaotong University*, vol. 51, no. 5, pp.48-53, 2017.

[11] P. Teng, J. Zhang, L. Chen, and X. Wang, "Random array code: A highly disaster-tolerant and expandable raid storage disaster recovery method," *Engineering Science and Technology*, vol. 49, pp. 3, pp. 110–116, 2017.

[12] Y. Wang, F. Xu, and X. Pei, "Research on erasure-code fault-tolerance technology in distributed storage," *Journal of Computer*, vol. 1, pp. 236–255, 2017.

[13] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[14] S. Yang and G. Zhang, "A survey of data repair methods in erasure code storage system," *Computer Science and Exploration*, vol. 11, no. 10, pp. 1531–1544, 2017.

Biography

Di Fan, master candidate. She is currently an Master student in Chengdu University of Information Technology, Chengdu, China. Her research interests include coding theory and information security.

Feng Xiao, Master candidate. He is currently an Master student in Chengdu University of Information Technology, Chengdu, China. His research interests include coding theory and database theory.

Dan Tang received his Ph.D. degree from Graduate University of Chinese Academy of Sciences (CAS), Beijing, China in 2010. He is currently an associate professor with Chengdu University of Information Technology, Chengdu, China. His research interests include coding theory and secret sharing scheme.