The Efficient Erasure code with Low Recovery-Overhead and degraded reads optimization

Hang Zhang¹, Shan Zheng Liu² and Dang Tan*¹

School of Software Engineering, Chengdu University of Information Technology, Chengdu, Sichuan, 610225, China

*Corresponding author’s e-mail: 1521717495@qq.com

Abstract. At present, the distributed storage systems are being adopted by more and more Internet companies because of their reliability and efficiency. In the distributed storage systems, some fault-tolerant technologies are often used to ensure the data reliability. Since the erasure code technology can store more data with lower storage space, it is often used as one of fault-tolerant technologies by some distributed storage systems. However, the existing erasure code technology has many problems. The large amount of repair overhead and I/O overhead consumed to recover data after data loss has become the biggest obstacle in actual use. This paper starts from the RS code, and constructs a new packet repair SLRC code by using the coding idea of stripe set, which can effectively reduce the repair cost required for single-node data recovery. At the same time, for the structure of SLRC code, a new data storage method, Curve storage, can effectively reduce the extra I/O overhead required for SLRC code degraded read operation. The experimental horizontal comparison with the LRC code and the RS code shows that the new method can effectively reduce the single node repair overhead and the additional I/O overhead required for the degraded read operation.

1. Introduction

Currently, distributed storage systems often use replica technology and erasure code technology as fault tolerance techniques. Among them, the erasure code is receiving more and more attention due to its lower storage overhead. However, as the amount of data continues to increase, the number of nodes continues to increase, leading to a gradual increase in the probability of abnormal data loss due to nodes. To recover lost data in a distributed storage system, the existing erasure code technology requires a large amount of repair overhead, which cannot meet the needs of current Internet companies. How to reduce the repair overhead of erasure code in data recovery has become a hot research issue.

Moreover, in a distributed storage system, due to a node failure or a busy network, data cannot be accessed at all times. At this time, the system enters a degraded read mode. In order not to affect the user experience, it is necessary to study how to effectively reduce the extra I/O overhead.

In recent years, many scholars have done a lot of work on reducing repair overhead and reducing the extra I/O overhead about degraded reads. The LRC code[1] is proposed by Lin W K. The LRC code uses the intra-group redundancy block to reduce the repair overhead required to recover data by grouping the stripes. The Pyramid code[2] was proposed by Huang. The Pyramid code reduces the repair bandwidth during data recovery by adding additional local redundancy blocks to the stripe. A new hybrid recovery method has been proposed by Chang Gan[3], which can reduce the amount of data read during the EVENODD code recovery process. A new fault repair method based on continuous disk readout was proposed by Zhang Yan[4] to ensure the minimum repair cost of
repairing node failure data recovery in RDP. A new EXPyramid code[5] was proposed by Zhou Song, which can greatly reduce the repair cost of multi-node data recovery. The RDOR algorithm[6] was proposed by Xu. RDOR reconstructs the failed elements with diagonal check elements and horizontal check elements to minimize the number of data elements read in the degraded mode. A new RAID-6 encoding algorithm[7] was proposed to reduce the I/O overhead of degraded reads by exploiting the characteristics of consecutive data elements in horizontal parity that may share the same level of checksum.

However, statistics on various data loss node anomalies in distributed storage systems are made, with 99.75% being single node failures[8]. Existing methods still cost a lot of repair costs if they are for single-node failure data recovery. Therefore, a new packet repair code-SLRC code is proposed to solve the problem of excessive cost recovery of single node failure data in the existing erasure code. At the same time, for the structure of SLRC code, a new data storage method, Curve storage, can effectively reduce the extra I/O overhead required for SLRC code degraded read operation.

The organization of this paper is as follows: Section 2 introduces the basic idea of coding. Section 3 specifies the specific encoding and decoding methods of SLRC proposed in this paper, and gives specific examples. Section 4 introduces the new data storage method—Curve storage. Section 5 performs specific experiments on the SLRC code and compares the LRC code and RS code horizontally. Finally, the summary of the full text.

2. Basic idea of RS coding
RS coding is an encoding method with MDS characteristics, and its coding can be divided into two types, one is Cauchy RS coding, and the other is Vandermonde RS coding. The \((n,k)\) RS code indicates that the \(k\) data blocks are encoded to generate \(n-k\) block redundancy blocks to form a code group of a total of \(n\) blocks. The Cauchy matrix is shown below.

\[
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
\frac{1}{x_1+y_1} & \frac{1}{x_1+y_2} & \cdots & \frac{1}{x_1+y_n} \\
\frac{1}{x_2+y_1} & \frac{1}{x_2+y_2} & \cdots & \frac{1}{x_2+y_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{1}{x_m+y_1} & \frac{1}{x_m+y_2} & \cdots & \frac{1}{x_m+y_n}
\end{bmatrix}
\tag{1}
\]

Where \(x_i, y_j\) are two elements in the galois field, taken from the set of elements \(X\) and \(Y\), and satisfy the following restrictions:

1) \(x_i + y_j \neq 0\) \quad (2)

2) \(\forall i, j \in \{1,2,\cdots,m\}, \quad i \neq j, \quad x_i \neq x_j\) \quad (3)

3) \(\forall i, j \in \{1,2,\cdots,n\}, \quad i \neq j, \quad y_i \neq y_j\) \quad (4)

3. SLRC encoding and decoding
The SLRC code is derived from the RS code, constructs a local repair code and reconstructs the local repair code.

3.1 SLRC encoding
The SLRC is constructed as follows:

Step 1: Starting from the RS code, multiply the data block by the Cauchy matrix to get \(n\) blocks to represent \(C_1, C_2, \cdots, C_n\). Among them, \(n\) blocks contain \(m\) global redundant blocks denoted as
and one data strip is divided into $L$ groups, which are recorded as $L_i, i = 1, 2, \ldots, n$, each group contains $k_y$ data blocks. Keep the $m_y$ redundant blocks unchanged before the erasure code, and calculate $m_t = m - m_y$ intra-group redundancy blocks for each group, expressed as $R_y$.

Step 2: Combine a plurality of strips into a strip set, forming each strip set comprising $S$ strips, each strip comprising a structure of $L$ groups. The last redundant block of each group $L_s$ uses the down-layer coding method. The down-layer coding method is to take the first $t$ data blocks in each group, and add the post $k_0 - t$ data blocks in the same position group in the next stripe to form a new data block set, and participate in the coding calculation with the new data block set.

Suppose the generation matrix is as follows, $g_{ij}$ represents the elements in the matrix, $SLR_{s,l}$ represents the $s$ strip in the strip set, and the rotating redundant block in the $l$ group, the down-layer redundancy block coding formula is as follows:

$$
SLR_{s,l} = \sum_{j=1}^{t} g_{s,j} D_{s,j} + \sum_{j=1}^{t} g_{n,j} D_{(s+1) \mod S,j},
$$

$$
0 < l \leq L, 0 < t \leq (k / L), 0 < s \leq S
$$

The following describes the encoding process of SLRC starting from the (11,8) RS code. Figure 1 shows the coding structure of the (11,8) RS code grouping. The $D_1, D_2, \ldots, D_8$ is 8 data blocks, the strips are divided into 2 groups. The $\bigcirc$ data block and the $\bigotimes$ data block respectively generate two groups of redundant blocks, and the $\bullet$ data block generates a global redundant block.

![Figure 1. RS code grouping.](image)

As shown in Figure 2, the local redundant block is calculated according to the introduction method in the second step, and $t = 1$ is set, where $\bullet$ represents the normal RS code structure, and $\bigotimes$ indicates the down-layer coding structure.

![Figure 2. Stripe set coding structure.](image)

Compared to the RS code, the SLRC code first generates a partial redundancy block by grouping the data strips, and the combined strips form a stripe set. The last redundant block of each group uses the down-layer coding method. This design ensures that the recovery can be quickly performed within...
the group when a single node fails. At the same time, the recovery is performed in units of strip sets, which can greatly reduce the data blocks that need to be read. The number reduces the cost of repair.

### 3.2 SLRC decoding

When the failed node is a single node, decoding is performed in units of stripe sets. Assuming the failed single node is in the \( L_i \) group, the decoding process proceeds as follows:

1. **Step 1:** When the failed node is in the previous \( t \) column of the \( L_i \) group, the data is recovered according to the conventional RS decoding algorithm. The former \( k_0 \) blocks that have not expired in the \( L_i \) group are combined into a new block set, and the RS data decoding mode is used to recover the failed data blocks.

2. **Step 2:** When the failed node is not in the previous \( t \) column, the stripe set is decoded in groups of 2 stripes. Suppose the two stripes are \( S_t, S_{t+1} \). To recover the failed data block in the \( S_t \) stripe, the first \( k_0 \) valid blocks of the \( L_i \) group in the \( S_t \) stripe need to be read. When recovering the failed data block in the \( S_{t+1} \) stripe, it is necessary to read the block buffered by the \( S_t \) stripe decoding, the block that has not failed in the last \( k_0 - t \), and the cyclically encoded redundant block.

In the figure 3, the \( S = 2 \) SLRC code is invalidated in the single node of the \( L_1 \) group, and the decoding process is demonstrated. The shading in Figure 3 indicates that the data block has failed. \( \bigtriangleup \) denotes a block that needs to be used to recover two strips, \( \bullet \) denotes a block to be used for restoring the first strip, and \( \bigcirc \) denotes a block to be used for restoring the second strip. Compared with the RS code, it is necessary to read 16 blocks for the failure of recovering 2 stripes, and the SLRC code only needs to read 6 blocks of data, and the repair cost is reduced by 62.5%.

![Decoding diagram](image)

Figure 3. Decoding diagram.

### 4. degraded reads optimization

If one node in the distributed storage system fails and no redundant nodes are available to recover the data, the system enters a degraded read mode. For the encoding structure of SLRC, a new data storage method, Curve storage, is proposed to reduce the extra I/O overhead required for SLRC code degraded read operations. The specific data storage method of Curve storage is as follows:

1. **Step 1:** The stripe set is divided into two strips, \( A_i = \{S_q, S_p\} \) (where \( i = 1, 2, \ldots, S/2 \) ), and \( S_q, S_p \) are the two strips associated with the rotation encoding.

2. **Step 2:** Gradually store the data blocks in the leftmost position of each group in the \( S_q \) strip until the end of the group data block storage.

3. **Step 3:** After the \( S_q \) strip data block is stored, each group in the \( S_p \) strip starts to store the data block at the far right until the group data block is stored.

In the figure 4, two strips are taken as an example to show how the data blocks are stored in the group.
Assuming $t = 2$, when the third column fails, the system needs to read the data block information of $D_1 - D_6$, only need to read $D_1 - D_6$ and redundant blocks to recover the data block of the invalid third column, no additional reading is required. Taking two data blocks, $D_7, D_8$, can reduce the extra I/O overhead by 25% compared with the traditional method.

5. Experimental results and analysis
In order to test the encoding and decoding performance of the SLRC code and reduce the I/O overhead saved by the read. An experimental platform was built based on the Ceph distributed storage system. The parameters of each machine are: CPU Intel Core i5, memory 8GB, disk 500GB.

5.1 Average repair cost
Figure 5 is a comparison of the average repair cost of a single node failure in the $t = 2$, $k_0 = 5$ SLRC code of the RS code, the LRC code. The storage file size is 10M, 20M, 30M, 40M, 50M, and the default size of the data block is 1K. The amount of file data, the ordinate indicates the average repair cost required for single-node failure recovery. The single node failures are all in the back $k_0 - t$ row. It can be seen from Figure 5 below that the repair cost of SLRC is about 58.5% lower than that of RS. It can be seen that the repair cost of SLRC is reduced by 18.6% compared with LRC.

5.2 Average read overhead for downgrade read savings
To test the average overhead of degraded read savings, five different sets of data were provided to test the downgrades within the group. The read elements in the test, the starting elements read are randomly generated by the system. Let the structure of each group be $S = 2$, $t = 2$, $k_0 = 5$, and the third column in the group is the failure column. The data tested is shown in the table below.
Table 1. Test data

| Numbering | Starting element | Read length | Actual number of reads |
|-----------|------------------|-------------|------------------------|
| 1         | 3                | 8           | 8                      |
| 2         | 1                | 9           | 9                      |
| 3         | 3                | 7           | 7                      |
| 4         | 4                | 9           | 9                      |
| 5         | 1                | 7           | 7                      |

Figure 6 is a comparison of the number of block reads required for the degraded read of the data stored in the group using the Curve storage method and the conventional method of storing data. All data is the data shown in the above table.

It can be seen from the figure that the group in the SLRC uses the Curve storage method to store data, which can effectively reduce the extra I/O overhead when degrading the read.

6. Conclusion

The SLRC code is an improved code based on the RS code, and the intra-group redundant block is down-layer coding in units of strip sets. This method can greatly reduce the repair cost required for single-node failure data recovery. At the same time, a new data storage method, Curve storage, is proposed for the coding structure of SLRC code, which can effectively reduce the extra I/O overhead required for downgrade reading. The experimental results show that the repair cost of SLRC is about 58.5% lower than that of RS, and the repair cost of SLRC is reduced by 18.6% compared with the LRC code. At the same time, the Curve storage data storage method in the SLRC code, can reduce the data block required for the degraded reading by about 10%-20%.

Acknowledgments

This work was financially supported by the project of Sichuan Science and Technology Department (2018CC0093) and the project of Sichuan Educational Commission(2018GZDZX0030).

References

[1] Lin, W. K, Chiu, et al. Erasure Code Replication Revisited[M]// Erasure code replication revisited. 2004.

[2] HUANG, C., CHEN, M., AND LI, J. Pyramid Codes: Flexible Schemes to Trade Space for Access Efficiency in Reliable Data Storage Systems.

[3] Chang Gan, Xu Weilong, Xiang Liping, et al. Fast recovery algorithm for single disk fault based on EVENODD code[J]. Journal of Computer Applications and Software, 2011(06):21-24.
[4] Zhang Yan. Disk Fault Repair of Distributed Storage System Based on RDP Encoding[D]. University of Science and Technology of China, 2014.

[5] Song Z, Wang Y. EXPyramid: An Array-Based Flexible Coding Scheme with High Fault-Tolerance and Low Recovery-Overhead[J]. Journal of Computer Research & Development, 2011.

[6] Xiang L, Xu Y, Lui J C S, et al. Optimal recovery of single disk failure in RDP code storage

[7] Lai Burning, Feng Xingjie, Wang Qing, et al. A feasible RAID-6 coding algorithm for optimizing degraded read performance [J]. Journal of Civil Aviation University of China, 2018, 36(04): 52-56.

[8] A. Gibson, "Disk failures in the real world: what does an MTTF of 1, 000, 000 hours mean to you?" FAST, vol. 7, pp. 1-16, 2007