Comparative Study of Erasure Code in Distributed Systems and Blockchain

Yuewen Xu *

School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China

* Corresponding Author Email: Yuewen.Xu19@student.xjtlu.edu.cn

Abstract. As the data generated by various applications continues to grow exponentially, the need for efficient methods of storage and transmission increases. Erasure codes offer a promising solution by providing redundant information without completely recreating the data. Although erasure codes have been extensively studied, their integration and effectiveness in distributed systems and blockchain networks are still being explored. This paper explores and compares the foundations and applications of erasure codes in distributed systems and blockchain networks, and provides a comprehensive literature review of erasure code implementations and their impact on data storage and transmission. This study explores the foundations of erasure codes in distributed systems, including array and RS codes, and LDPC codes. The study also explores the use of erasure codes in distributed systems and blockchain and analyzes how these codes can improve the efficiency and scalability of decentralized storage solutions. The study examines the strengths and weaknesses of different erasure codes based on analysis and comparisons. The findings highlight the potential of erasure codes to revolutionize data storage and transmission in distributed systems and blockchain networks.

Keywords: Erasure Code; Distributed Systems; Blockchain.

1. Introduction

With the exponential growth of data generated by various applications and services in recent years, the need for reliable and cost-effective data storage and transmission has increased. In distributed systems and blockchains, traditional replication-based approaches suffer from high storage overhead and high network bandwidth consumption, whereas erasure codes offer an alternative solution by introducing redundancy without the need for full replication. Erasure code is a method for recovering partially lost information, adding redundancy to the system, and making the system tolerant to failures, and plays a crucial role in dealing with the problem of data loss in large-scale data and network communications [1]. Previous research on erasure codes focuses on their use in distributed storage systems. Array erasure codes (e.g., Reed-Solomon (RS) codes and Low-Density Parity Check (LDPC) codes) have been extensively studied and proven to be highly effective in preventing node failures and reducing storage overhead. However, despite significant research on erasure codes, their application to distributed systems and blockchains is still a relatively new and understudied area. For instance, in geographically distributed cloud storage systems, erasure codes may result in high service latency as end-users access remote storage nodes to retrieve data [2].

The main focus of this paper is to examine the foundations and applications of erasure codes in distributed systems and blockchains. This paper will begin by examining the fundamental concepts of erasure codes in distributed systems, including array and RS codes, and then delve into their practical applications in distributed systems and blockchain networks. In distributed systems, their effectiveness in terms of fault tolerance, data reliability, and storage optimization will be evaluated. On the blockchain side, it will be investigated how erasure codes can improve the efficiency and scalability of decentralized storage solutions. In conclusion, a comparative analysis is conducted to assess the performance and effectiveness of different erasure codes in various scenarios and reveal their potential and obstacles.

2. Foundations and Applications of Erasure Code in Distributed Systems

2.1. Foundations of Erasure Code in the Distributed System

In distributed computing and artificial intelligence, erasure code is mainly used in distributed storage systems to improve storage space utilization at the expense of CPU computation and network load, while providing near-copy reliability. Currently, there are three main types of erasure codes studied in distributed storage systems: array erasure codes (RAID5, RAID6, etc.), RS (Reed-Solomon) erasure codes, and LDPC (Low-Density Parity Check Code) erasure codes.

2.1.1 Array erasure code

This is a data redundancy protection technology that is an extension of RAID. When the redundancy level is $n+m$, the checksum block of m is calculated from n source data blocks, and these $n+m$ data blocks are stored on $n+m$ hard discs, which can tolerate any m hard disc failures. When the hard disc fails, only n normal data blocks are arbitrarily selected to compute all source data. Array erasure code is a type of code that uses binary XOR coding and a decoding operation, which has high computational efficiency [3].

2.1.2 RS erasure code

Reed Solomon (RS) code is a non-binary, finite-domain arithmetic-based erasure code. It can detect and correct multiple symbol errors compared to EVENODD and row-diagonal parity check (RDP) codes, which can correct up to two erasures [4,5]. The Reed-Solomon (RS) code is one of the most commonly used erasure codes in storage systems, with two parameters, n and m, and is denoted as RS (n,m). n stands for the number of original data blocks. m stands for the number of checksum blocks, and RS code is a non-binary, finite-domain arithmetic-based erasure code. The principle of RS code is as follows.

In the case of $n=5$ and $m=3$, there are 5 original data blocks multiplied by a $(n+m,n)$ matrix, resulting in an $(n+m,1)$ matrix. Based on the characteristics of the matrix, it can be observed that the first 5 values in the resulting matrix are equal to the original 5 data blocks, while the last 3 values are computed parity blocks.

$$\begin{matrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\\B_{11}&B_{12}&B_{13}&B_{14}&B_{15}\\B_{21}&B_{22}&B_{23}&B_{24}&B_{25}\\B_{31}&B_{32}&B_{33}&B_{34}&B_{35}\end{matrix} * \begin{matrix}D_1\\D_2\\D_3\\D_4\\D_5\end{matrix} = \begin{matrix}D_1\\D_2\\D_3\\D_4\\D_5\\C_1\\C_2\\C_3\end{matrix} \quad (1)$$

The above process is the encoding process. D is the original data block, and the resulting C is the checksum block.

Suppose 5 blocks of data are lost. As follows,

$$\times \quad D_2 \quad D_3 \quad \times \quad D_5 \quad C_1 \quad \times \quad C_3 \quad (2)$$

The following decoding process is required to retrieve the original n data blocks from the remaining n data blocks (Note that the remaining n blocks may contain several original data blocks and several checksum blocks).

Step 1: Delete the missing data blocks and corresponding rows of the missing coding blocks from the coding matrix. The $(n+m,1)$ matrix with m blocks deleted is deformed to the $(n,1)$ matrix, and the B matrix also needs to delete the corresponding m rows to obtain a deformed matrix of B', which is the $n*n$ matrix. As follows: If $D1,D4,C2$ are lost, we get the following B' matrix and equation.

$$\begin{matrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ B_{11} & B_{12} & B_{13} & B_{14} & B_{15} \\ B_{31} & B_{32} & B_{33} & B_{34} & B_{35} \end{matrix} * \begin{matrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \end{matrix} = \begin{matrix} D_2 \\ D_3 \\ D_5 \\ C_1 \\ C_3 \end{matrix} \tag{3}$$

$$B' * D = Survivors \tag{4}$$

Step 2: Find the inverse matrix of B' and then multiply each side of the equation by the inverse matrix of B'.

$$\begin{matrix} & & \\ 1\ 0\ 0\ 0\ 0 & & \\ 0\ 1\ 0\ 0\ 0 & * & \\ & & \\ 0\ 0\ 1\ 0\ 0 & & \end{matrix} \begin{matrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ B_{11} & B_{12} & B_{13} & B_{14} & B_{15} \\ B_{31} & B_{32} & B_{33} & B_{34} & B_{35} \end{matrix} * \begin{matrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \end{matrix} = \begin{matrix} & & \\ 1\ 0\ 0\ 0\ 0 & & \\ 0\ 1\ 0\ 0\ 0 & * & \\ & & \\ 0\ 0\ 1\ 0\ 0 & & \end{matrix} \begin{matrix} D_2 \\ D_3 \\ D_5 \\ C_1 \\ C_3 \end{matrix} \tag{5}$$

$$B'^{-1} * B' * D = B'^{-1} * Survivors \tag{6}$$

$\boldsymbol{B'}$ and its inverse matrix $\mathbf{B'^{-1}}$ are multiplied to obtain the unit matrix I as follows.

$$\begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{matrix} * \begin{matrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \end{matrix} = \begin{matrix} & & \\ 1\ 0\ 0\ 0\ 0 & & \\ 0\ 1\ 0\ 0\ 0 & * & \\ & & \\ 0\ 0\ 1\ 0\ 0 & & \end{matrix} \begin{matrix} D_2 \\ D_3 \\ D_5 \\ C_1 \\ C_3 \end{matrix} \tag{7}$$

$$I * D = B'^{-1} * Survivors \tag{8}$$

Therefore, the original data matrix D is

$$\begin{matrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \end{matrix} = \begin{matrix} & & \\ 1\ 0\ 0\ 0\ 0 & & \\ 0\ 1\ 0\ 0\ 0 & * & \\ & & \\ 0\ 0\ 1\ 0\ 0 & & \end{matrix} \begin{matrix} D_2 \\ D_3 \\ D_5 \\ C_1 \\ C_3 \end{matrix} \tag{9}$$

$$D = B'^{-1} * Survivors \tag{10}$$

2.1.3 LDPC erasure codes

LDPC (Low-Density Parity-Check) codes have slightly lower coding efficiency compared to RS codes, but their coding and decoding performance is better than that of RS codes as well as other erasure codes due to the relatively few and simple heterodyne operations used in the coding and decoding process. Currently, LDPC codes are mainly used in the fields of communication, video and audio coding.

Notably, LDPC codes are a very efficient coding method because they provide a practical implementation close to the reliable transmission capacity of the Shannon channel. According to the Shannon channel capacity theorem, as the code rate approaches the capacity number, the error in decoding by a maximum likelihood decoder will tend to zero as the block length increases [6]. This condition can be achieved by using random linear packet codes that are encoded as polynomials in time [7].

The checksum matrix of LDPC codes (Low-Density Parity Check Codes) is very sparse. This means that the checksum matrix has a large number of zero elements and relatively few non-zero elements. Specifically, in each row, very few elements take the value of 1 and the rest of the elements are 0. Due to this distributional characteristic, the checksum matrix is called a low-density matrix.

What the checksum matrix does is when you receive a string of information bits, multiply it by each row of the checksum matrix and add the results. Eventually, you get several results equal to the number of rows. If all the results are 0, the checksum passes. The formula is expressed as:

$$H \cdot C^{\tau} = 0 \tag{11}$$

If the LDPC code to be sent has a code word length of N and a message bit length of K, then the length of the checksum message bits is M = N - K. The code rate R = K/N. Therefore, the size of the LDPC code checksum matrix H required is M * N. An example is the following matrix:

$$H = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{12}$$

Here are some characteristics of the checksum matrix H:

Each row has the same number of "1" elements, k, where k is the row weight.

Each column has the same number of "1" elements, j, where j is the column weight.

In the matrix H, the number of "1" in the same position in any two columns does not exceed 1.

Based on the matrix, a Tanner diagram can be drawn that represents the same code as the above matrix, with the number of verification nodes M and variable nodes N. The relationship between them can be expressed by the following equation (Figure 1).

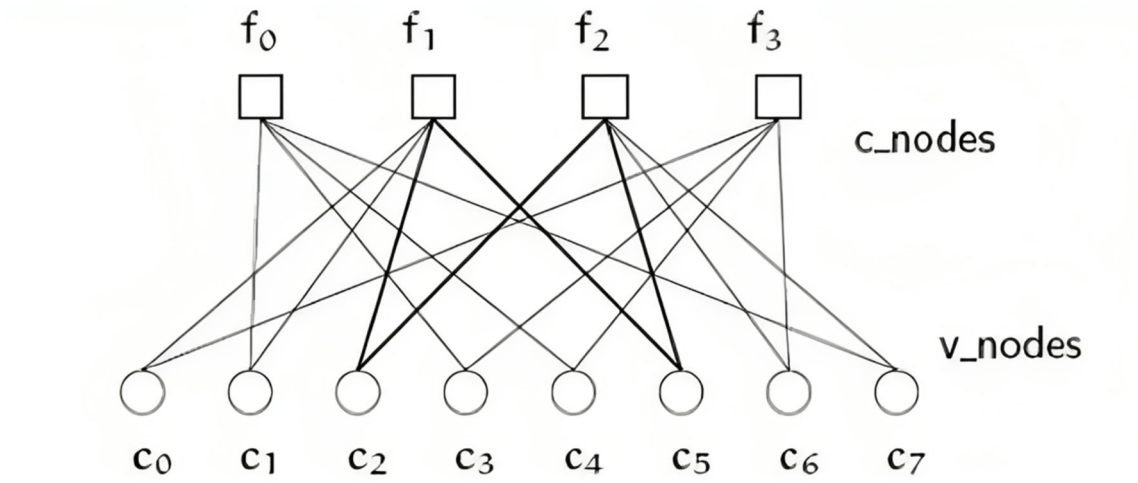

Fig 1. Tanner graph [4]

The relationship between them can be expressed by the following equation.

$$f_0 = C_1 \oplus C_3 \oplus C_4 \oplus C_7 \tag{13}$$

$$f_1 = C_0 \oplus C_1 \oplus C_2 \oplus C_5 \tag{14}$$

$$f_2 = C_2 \oplus C_5 \oplus C_6 \oplus C_7 \tag{15}$$

$$f_3 = C_0 \oplus C_3 \oplus C_4 \oplus C_6 \tag{16}$$

2.2. Applications of Erasure Code

2.2.1 Erasure code in hadoop distributed file systems

Hadoop Distributed File System (HDFS) is a fault-tolerant and high-throughput solution for storing massive data, running on general-purpose hardware in a distributed manner [8]. In HDFS, large files are divided into multiple data blocks and stored redundantly on different compute nodes. The Erasure Code is widely used in HDFS to encode and generate redundancy in the data blocks, improving data reliability and fault tolerance.

The erasure code storage scheme in HDFS is comprised of three main components: the client, the codec server, and the service cluster for distributed storage. The client program allows users to back up, restore, and delete data files, as well as encrypt transferred files for secure transmission. The coding and decoder system is responsible for coding and decoding client data, serving as the interface between the client and the distributed storage server. The distributed storage system serves as the storage medium for this disaster recovery solution [9].

3. Foundations and Applications of Erasure Codes in Blockchain

3.1. Foundations of Erasure Code in Blockchain

Erasure code is a forward error correction technique for high availability and reliability in storage and communication systems [10]. In network transmission, it is mainly used for packet loss recovery; in storage systems, it is mainly applied to improve the reliability of storage. Erasure code extends and encodes redundant data blocks by splitting the data into segments and storing them in different locations, such as disks, storage nodes, or other geographical locations. The total data block consists of the original data block and the checksum block, commonly denoted as $n = k + m$.

When the redundancy level is n, these data blocks are stored on n separate hard drives. This keeps the data safe even if m drives (assuming k initial data) fail. As long as no more than m drives fail, all the original data can be computed by arbitrarily selecting k normal data blocks.

3.2. Applications of Erasure Code in Blockchain

CITA is an open-source blockchain operating system kernel designed for high stability, performance, and scalability. In the CITA blockchain, erasure codes and Byzantine error-tolerant algorithms are used in combination to reduce storage consumption per block from $O(n)$ to $O(1)$, thus improving the scalability of the system [11].

This system is based on a blockchain system using erasure code slices and dice storage. The user sends transactions to the system through the client and stores them temporarily in the transaction pool. At regular intervals, the transactions in the transaction pool are packaged into a block, consensus is reached among the nodes, and then it is stored in the system, and finally, after encoding with erasure code, it is distributed and stored on the nodes of the system.

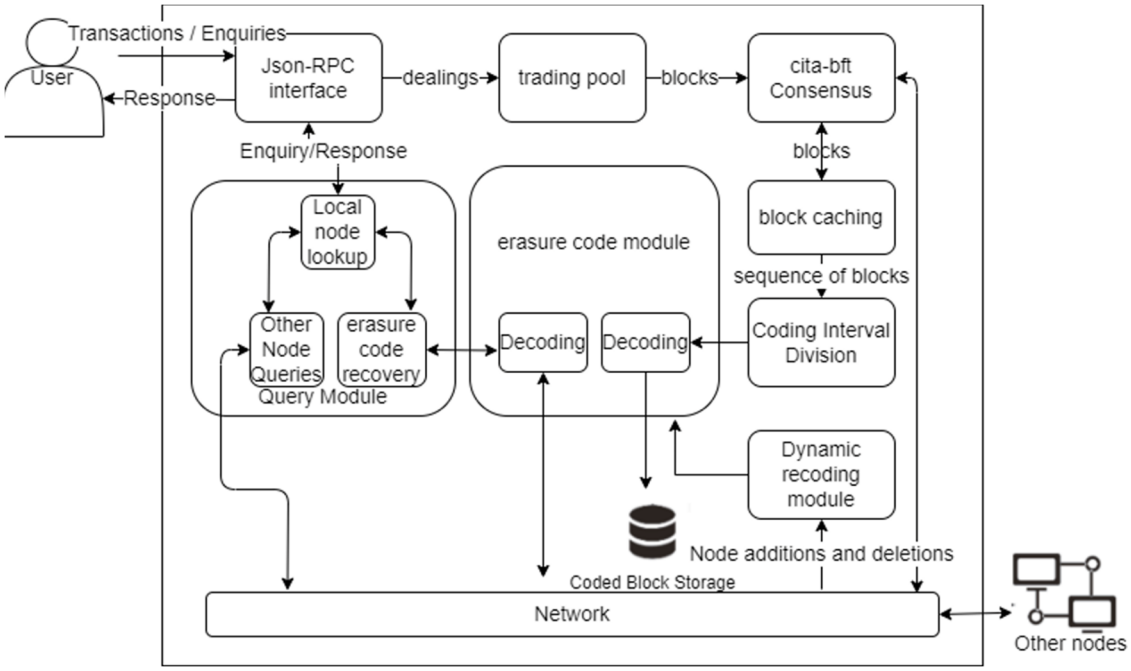

Fig 2. System architecture of node [11]

The system's framework is depicted in Figure. 2. This consists of three main parts: the erasure code module, the query module, and the dynamic recoding module. The functions of these three modules are as follows:

Erasure Code Module: It is responsible for encoding blocks stored in the system with erasure codes and storing the encoded data distributed on the nodes of the system.

Query module: Responsible for processing user queries and retrieving the original data through decryption.

Dynamic Recoding Module: Responsible for dynamic recoding of data stored on nodes when blockchain nodes are joined or exited to ensure system stability and reduce system recoding overhead.

This implementation of slice storage based on erasure code significantly improves the storage efficiency and data reliability of the CITA blockchain.

4. Comparison of Erasure Codes in Distributed Systems and Blockchain

In distributed computing, erasure codes are usually combined with distributed storage systems, e.g., a system combining Hadoop with erasure codes can provide high disaster recovery performance and security. However, since the encoding and decoding of erasure codes takes up a certain amount of time and computational resources, it may reduce the computational efficiency of the system.

In blockchain, erasure codes are mainly used to optimize storage performance. For example, the CITA blockchain-based erasure code slice-and-dice storage method can reduce redundancy in block storage and reduce communication overhead between nodes. However, the encoding and decoding processes still introduce some time consumption.

In both distributed computing and blockchain, erasure codes are used to improve storage resource utilization and data reliability. The difference is that they do have some differences.

Table 1 below compares the application of erasure codes in distributed systems and blockchain in four ways.

Table 1. Comparison of erasure codes utility in distributed systems and blockchain systems

Number	Distributed Computing	Blockchain
1.Application Scenario	Mainly used to improve fault tolerance and data backup security.	Mainly used to optimize storage performance and reduce redundancy.
2.Storage Mode	Usually cuts data into multiple code blocks and stores them in different nodes.	Reduce the amount of data generated from the blockchain, ensuring that any block of the chain can be easily reconstructed from a small number of nodes [12].
3.Efficiency Impact	Helps minimize the amount of remediation data transmitted over the network, the amount of data accessed on auxiliary nodes and the number of auxiliary nodes contacted [13].	Can reduce storage redundancy and communication overhead and improve system performance.
4. Security	Can provide high disaster recovery performance and security.	Can reduce the risk of data leakage and improve block query speed.

5. Conclusion

This paper delves into the foundations and applications of erasure codes in both distributed systems and blockchain through a thorough study. A distributed system is a network of multiple computers that work together to perform various tasks. Blockchain, a decentralized technology, has shown great potential in areas such as finance and the Internet of Things. In distributed systems, erasure codes are widely used for data fault tolerance and capacity optimization. The codes used for erasure include array erasure codes, RS erasure codes, and LDPC erasure codes. These erasure codes can achieve

high data reliability and fault tolerance by dividing data into different blocks and generating redundant data. In the field of blockchain technology, erasure codes play a crucial role. Erasure codes can help solve storage issues in blockchain, improve storage efficiency, and reduce storage overheads. By encoding and slicing data using erasure codes, data redundancy and recovery can be achieved, thus increasing data reliability and security. At the same time, erasure codes can also effectively reduce the storage space and bandwidth requirements for blockchain data and speed up transaction processing.

This paper provides a comprehensive overview of understanding and utilizing erasure codes, which provides new ideas and methods for related research and enhances the understanding and application of erasure code technology for distributed systems and blockchain. Furthermore, the application of erasure codes not only improves the reliability and security of data, but also reduces storage overhead and bandwidth requirements, which is of great significance for resource-constrained environments, providing researchers with more research directions and challenges.

Research on erasure codes has made significant progress, but there are still many unresolved problems and challenges. Further investigation into the application and optimization methods of erasure codes can lead to improved fault-tolerant performance and data reliability. In blockchain, the impact of erasure codes on different types of chains can be further studied and combining them with other cryptographic algorithms can enhance data security and privacy protection. Therefore, future research should further explore these issues and find new methods and techniques to address these challenges.

References

[1] StoneFly. Understanding erasure coding and its difference with RAID. StoneFly Blog, 2023.

[2] Liu K, Peng J, Wang J, et al. Adaptive and scalable caching with erasure codes in Distributed Cloud-edge storage systems. IEEE Transactions on Cloud Computing, 2022, 11(2): 1840–1853. DOI:10.1109/tcc.2022.3168662.

[3] Tang D, Wang Y, Yang H. Array Erasure Codes with Preset Fault Tolerance Capability. International Jounrnal Netw. Secur., 2018, 20(1): 193-200.

[4] Wang Z, Dimakis A G, Bruck J. Rebuilding for array codes in distributed storage systems. IEEE Globecom Workshops, 2010: 1905-1909.

[5] Corbett P, English B, Goel A, et al. Row-diagonal parity for double disk failure correction. Proceedings of the 3rd USENIX Conference on File and Storage Technologies, 2004: 1-14.

[6] Borwankar S, Shah D. Low Density Parity Check Code (LDPC Codes) Overview. ArXiv preprint, 2022.

[7] Shannon C E. A mathematical theory of communication. The Bell system technical journal, 1948, 27(3): 379-423.

[8] Vora M N. Hadoop-HBase for large-scale data. Proceedings of 2011 International Conference on Computer Science and Network Technology, 2011, 1: 601-605.

[9] XU Dongxu. Application of erasure coding in distributed fault-tolerant storage. Computer CD Software and Application, 2013, 16(03):103-104.

[10] Lin W K, Chiu D M, Lee Y B. Erasure code replication revisited. Proceedings. Fourth International Conference on Peer-to-Peer Computing, 2004: 90-97.

[11] Furong Yin, Chengyu Zhu, Bin Z. Erasure code partition storage based on the CITA blockchain. Journal of East China Normal University (Natural Science), 2021, 2021(5): 48.

[12] Perard D, Lacan J, Bachy Y, et al. Erasure code-based low storage blockchain node. IEEE International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data, 2018: 1622-1627.

[13] Balaji S B, Krishnan M N, Vajha M, et al. Erasure coding for distributed storage: An overview. Science China Information Sciences, 2018, 61: 1-45.