Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/31/20230168

Examination of essential technologies and representative implementations of storage encoding

Yutong Cai

College of Electronic Engineering/College of Artificial Intelligence, South China Agricultural University, Guangzhou, 510642, China

202034510201@stu.scau.edu.cn

Abstract. As we continue to navigate the wave of advancements in 5G technology, alongside artificial intelligence and cloud computing, we're witnessing an exponential surge in data generation. Consequently, the performance of storage systems has become increasingly critical. Erasure coding techniques, which originated as channel coding strategies for addressing data transmission errors, have since found application in storage systems. Primarily, erasure coding techniques for distributed storage systems deploy algorithms to generate some redundant data following the computation of the original data. If some of the data becomes invalid, the original data, along with the remaining valid original data and redundancy, can be recomputed using these algorithms. RAID6 is an approach designed to bolster the data protection of arrays. Coding algorithms such as RDP and EVENODD, grounded in RAID6 theory, have been extensively explored. For array codes like RAID6, there's a distinct advantage in dispersing two independent verification pieces of information across different disks. Even if two disks fail, the data and parity information on the remaining disks can be recovered. However, a clear drawback of the RAID6 array code surfaces when a single disk failure occurs: the recovery involves a significant number of disks, leading to extended recovery times, thereby elevating the risk of multiple disk failures. Consequently, the time allocated for data recovery and the number of nodes engaged during data recovery emerge as crucial factors for system stability. For RDP and EVENODD coding algorithms, a maximum of two errors can be tolerated. This paper introduces local redundancy to these two array codes to minimize the number of nodes visited during data recovery, thereby enhancing the system's stability.

Keyword: redundancy technology, erasure coding, RAID6, RDP, EVENODD.

1. Introduction

In the face of advancements in 5G technology, along with the proliferation of artificial intelligence, big data, and cloud computing, society is witnessing an exponential growth in data generation. As shown in the figure 1, it presents the actual and projected data volumes from 2010 through 2035 [1].

This vast data growth, paired with the evolution in distributed storage, cloud storage, and alternative technologies, has led to a more diverse, decentralized [2], and unstructured landscape of network storage. These factors add complexity to data management and pose significant challenges to the reliability of stored system data. Simultaneously, they raise higher requirements, thus drawing increasing attention, research, and discussion in the data storage domain [3]. The investigation of storage system fault-tolerance mechanisms and data recovery post-failure is of critical importance to overall storage system performance, making it a topic of substantial scientific interest. A robust fault tolerance mechanism is paramount for any storage system. Most contemporary storage systems consist of numerous individual disks, and as the number of disks escalates, the probability of data failure increases [4]. The replication technique involves creating a complete copy of the original data on a backup disk. Should the original data fail, the lost data can be retrieved from the backup. This method has the benefits of quick data recovery and minimal bandwidth usage [5]. However, its downside lies in its excessive redundancy, which can lead to space wastage. Consequently, it is usually employed for managing hot data that undergoes regular access [6]. On the other hand, erasure code techniques establish a small amount of check data and forge connections between the original data and the check data through a series of computational rules. When the original data fails, information about the original data and the link between the original data and the validation data can be derived to complete data recovery. The advantage of this technique is its low redundancy, saving storage space. Yet, its downside includes a slow data recovery rate and high bandwidth usage, so it is typically used for handling cold data that is not frequently accessed [7].

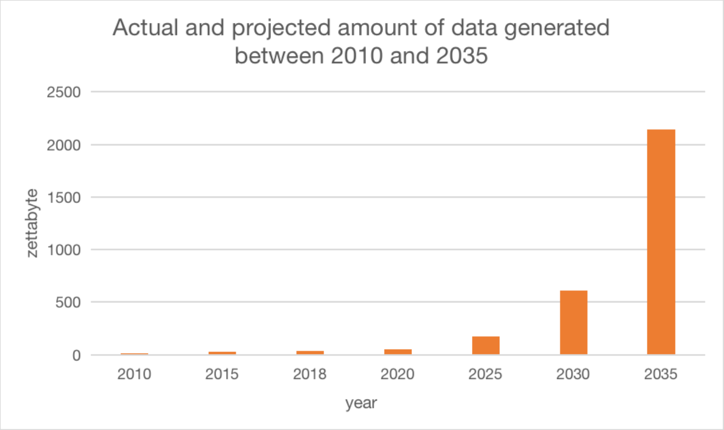

Figure 1. Actual and projected amount of data generated between 2010 and 2035 (Photo/ Picture credit: Original).

2. Coding techniques

2.1. RAID6 encoding

RAID technology is one of the effective ways to use disks. The name of RAID-6 is ‘Independent Data disks with two independent distributed parity schemes’.

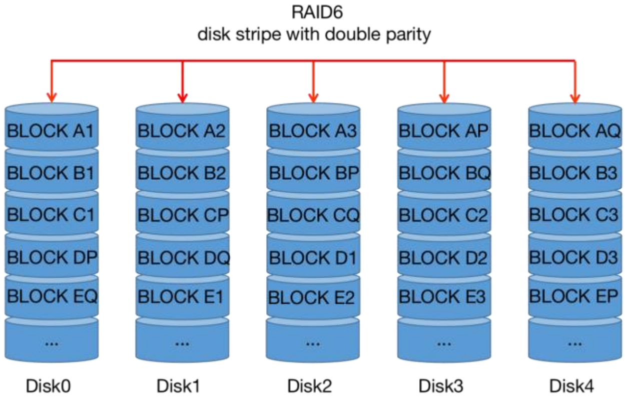

Figure 2. The structure of RAID6(Photo/ Picture credit: Original).

As shown in Figure 2, RAID6 adopts a P+Q dual-disk verification strategy, allowing any two disks to fail, therefore, the reliability of the storage system can be significantly improved The RAID6 storage system generates two parity chains during data storage. Therefore, RAID6 has two parity disks P and Q compared to RAID5. The two parity disks store P and Q information, respectively [8]. RAID6 has better reliability than RAID5 and RAID4, and popular erasure coding techniques are implemented on RAID6.

The coding performance of RAID6 largely determines the overall performance of RAID6 arrays. Three metrics are commonly used to measure the coding performance of RAID6, namely: codec computational complexity, which means the number of XOR calculations required in the encoding or decoding process; The more XOR times, the higher the computational complexity; update complexity, which means that the system changes the number of check blocks when updating a data block; rate refers to the proportion of data blocks in the RAID6 encoding structure [9]. RAID6 code is a typical MDS code, which follows the rule of Singleton dmin<=n-k-1, n means the length of the data message after coding while k is the length of the original message. Theoretically the MDS encoding has the best storage efficiency.

2.2. EVENODD encoding

EVENODD was proposed by Blaum et al. to solve the problem of dual disk failure in RAID systems. EVENODD is a horizontal encoding and a standard MDS code. The EVENODD encoding stores all user data on m (m is a prime number) disks and the parity data on the other two disks [10].

The theory of EVENODD codes has made an essential contribution to the development of fault-tolerant techniques. It is gaining popularity in a simple way and is widely used in various systems, especially in disk array layout schemes. The core operation is to make the data simply different or according to certain rules. Therefore, the study of EVENODD coding and its implementation is of great practical interest.

2.3. Analysis of improved coding

To compensate for the excessive number of nodes visited during data recovery by RDP coding and EVENODD coding, the number of nodes visited during data recovery can be reduced by adding local redundancy. The specific operations are as follows: The improve coding of RDP (take m=5 as an example) (figure 3).

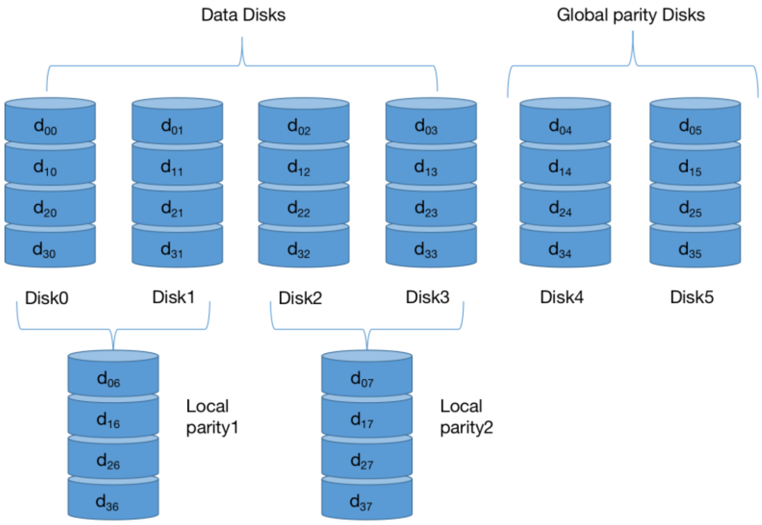

Figure 3. The structure of improved RDP coding scheme (Photo/ Picture credit: Original).

Here local parity1 is the XOR sum of Disk0 and Disk1, local parity2 is the XOR sum of Disk2 and Disk3. The improve coding of RDP (take m=5 as and example). Here local parity1 is the XOR sum of Disk0 and Disk1, local parity2 is the XOR sum of Disk2 and Disk3 (figure 4).

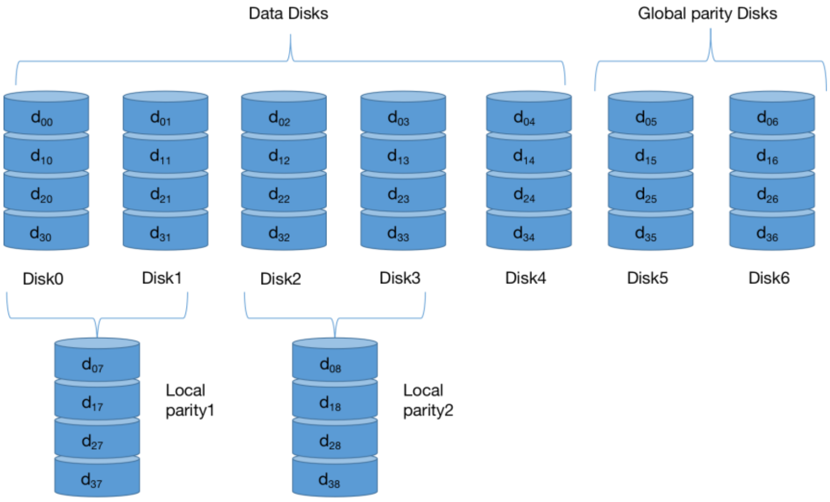

Figure 4. The structure of improved EVENODD coding scheme (Photo/ Picture credit: Original).

3. Typical application analysis

Among the failure cases of the storage system, the proportion of single-disk failure reaches 99.75%, accounting for almost all cases. So let's take a single disk failure as an example.

3.1. RDP code storage system fault recovery

Theoretically, the RDP code can correct the error of any two nodes. Since RDP is a classical MDS code, the computational complexity and efficiency of the codecs are theoretically optimal. RDP code is a two-dimensional array of size (m-1) * (m+1), where m is a prime number greater than 2, in which n=m+1, k=m-1. In this context, di,j is used to denote row i and column j of the disc. In RDP coding, the first m-2 columns are used to store the raw data, the m-1 columns serve as row checks, and the m-1 columns are utilized for diagonal checks. This can be expressed as follows:

$$\mathrm{d}_{\mathrm{i,m}-1} = \sum_{j=0}^{m-2} d_{i,j} \tag{1}$$

$$\mathrm{d}_{\mathrm{i,m}} = \sum_{j=0}^{m-1} d < i - j >_{m,j} \tag{2}$$

The sum is XOR sum. For example: m=5 (figure 5).

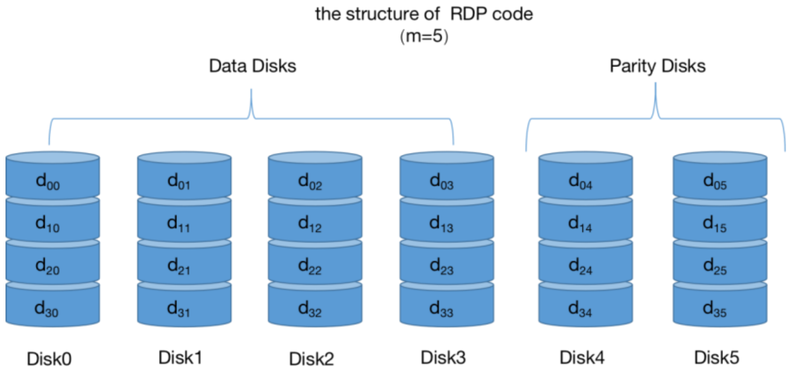

Figure 5. The structure of RDP coding (Photo/ Picture credit: Original).

The relation between parity disks and data disks are as follow:

$$\begin{gathered}
d_{0,4} = d_{0,0} \oplus d_{0,1} \oplus d_{0,2} \oplus d_{0,3} \\
d_{1,4} = d_{1,0} \oplus d_{1,1} \oplus d_{1,2} \oplus d_{1,3} \\
d_{2,4} = d_{2,0} \oplus d_{2,1} \oplus d_{2,2} \oplus d_{2,3} \\
d_{3,4} = d_{3,0} \oplus d_{3,1} \oplus d_{3,2} \oplus d_{3,3} \\
d_{0,5} = d_{0,0} \oplus d_{3,2} \oplus d_{2,3} \oplus d_{1,4} \\
d_{1,5} = d_{1,0} \oplus d_{0,1} \oplus d_{3,3} \oplus d_{2,4} \\
d_{2,5} = d_{2,0} \oplus d_{1,1} \oplus d_{0,2} \oplus d_{3,4} \\
d_{3,5} = d_{3,0} \oplus d_{2,1} \oplus d_{1,2} \oplus d_{0,3}
\end{gathered} \tag{3}$$

If a single disk failure occurs (figure 6).

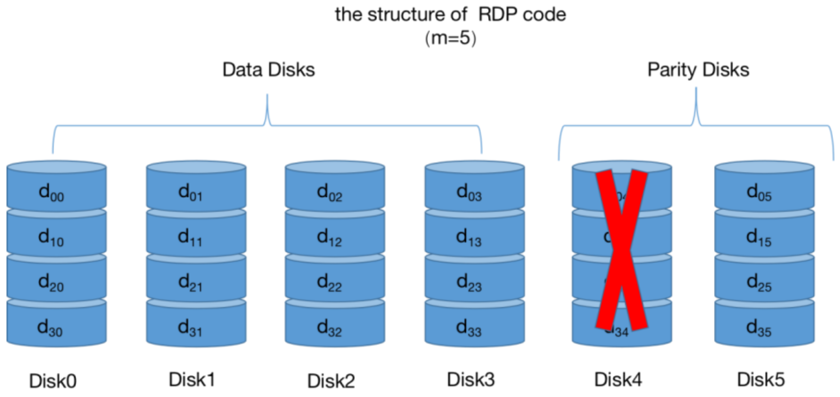

Figure 6. One parity disk failure in RDP coding (Photo/ Picture credit: Original).

In this case, if the parity disk fails, recover the data by a simple reconstruction via the above formula, yielding:

$$\begin{gathered}
d_{0,4} = d_{0,0} \oplus d_{0,1} \oplus d_{0,2} \oplus d_{0,3} \\
d_{1,4} = d_{1,0} \oplus d_{1,1} \oplus d_{1,2} \oplus d_{1,3} \\
d_{2,4} = d_{2,0} \oplus d_{2,1} \oplus d_{2,2} \oplus d_{2,3} \\
d_{3,4} = d_{3,0} \oplus d_{3,1} \oplus d_{3,2} \oplus d_{3,3}
\end{gathered} \tag{4}$$

When there is a single failure in the parity disk, the number of nodes to be visited is m-1 (figure 7).

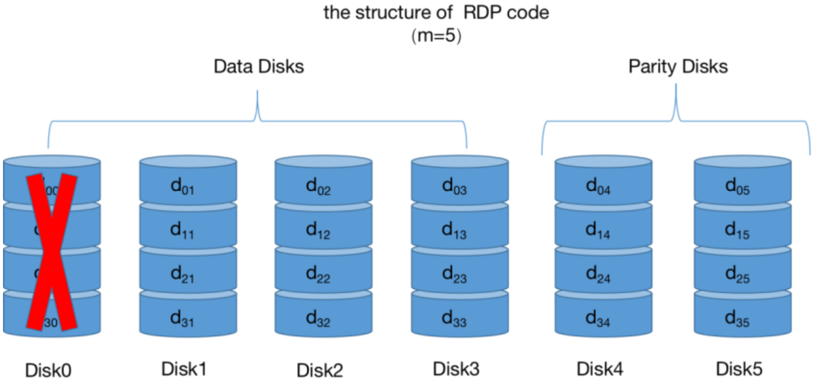

Figure 7. One data disk failure in RDP coding (Photo/ Picture credit: Original).

In this case, if a data disk fails, the data can be recovered by using a parity disk and three additional original data. Using the above formula:

$$\begin{aligned} d_{0,4} &= d_{0,0} \oplus d_{0,1} \oplus d_{0,2} \oplus d_{0,3} \\ d_{1,4} &= d_{1,0} \oplus d_{1,1} \oplus d_{1,2} \oplus d_{1,3} \\ d_{2,4} &= d_{2,0} \oplus d_{2,1} \oplus d_{2,2} \oplus d_{2,3} \\ d_{3,4} &= d_{3,0} \oplus d_{3,1} \oplus d_{3,2} \oplus d_{3,3} \end{aligned} \tag{5}$$

Or

$$\begin{aligned} d_{0,5} &= d_{0,0} \oplus d_{3,2} \oplus d_{2,3} \oplus d_{1,4} \\ d_{1,5} &= d_{1,0} \oplus d_{0,1} \oplus d_{3,3} \oplus d_{2,4} \\ d_{2,5} &= d_{2,0} \oplus d_{1,1} \oplus d_{0,2} \oplus d_{3,4} \\ d_{3,5} &= d_{3,0} \oplus d_{2,1} \oplus d_{1,2} \oplus d_{0,3} \end{aligned} \tag{6}$$

Yielding:

$$\begin{aligned} d_{0,0} &= d_{0,4} \oplus d_{0,1} \oplus d_{0,2} \oplus d_{0,3} \\ d_{1,0} &= d_{1,4} \oplus d_{1,1} \oplus d_{1,2} \oplus d_{1,3} \\ d_{2,0} &= d_{2,4} \oplus d_{2,1} \oplus d_{2,2} \oplus d_{2,3} \\ d_{3,0} &= d_{3,4} \oplus d_{3,1} \oplus d_{3,2} \oplus d_{3,3} \end{aligned} \tag{7}$$

Or

$$\begin{aligned} d_{0,0} &= d_{0,5} \oplus d_{3,2} \oplus d_{2,3} \oplus d_{1,4} \\ d_{1,0} &= d_{1,5} \oplus d_{0,1} \oplus d_{3,3} \oplus d_{2,4} \\ d_{2,0} &= d_{2,5} \oplus d_{1,1} \oplus d_{0,2} \oplus d_{3,4} \\ d_{3,0} &= d_{3,5} \oplus d_{2,1} \oplus d_{1,2} \oplus d_{0,3} \end{aligned} \tag{8}$$

When there is a single failure in the data disk, the number of nodes to be visited is still m-1. While using the improved RDP coding scheme (figure 3), the recovery would be considerably easier and faster. Using local parity1, apparently reduced disk access by 50% when recovering a disk failure.

$$\begin{aligned} \mathrm{d}_{0,6} &= \mathrm{d}_{0,0} \oplus \mathrm{d}_{0,1} \\ \mathrm{d}_{1,6} &= \mathrm{d}_{1,0} \oplus \mathrm{d}_{1,1} \\ \mathrm{d}_{2,6} &= \mathrm{d}_{2,0} \oplus \mathrm{d}_{2,1} \\ \mathrm{d}_{3,6} &= \mathrm{d}_{3,0} \oplus \mathrm{d}_{3,1} \end{aligned} \tag{9}$$

Gets:

$$\begin{aligned} \mathrm{d}_{0,0} &= \mathrm{d}_{0,6} \oplus \mathrm{d}_{0,1} \\ \mathrm{d}_{1,0} &= \mathrm{d}_{1,6} \oplus \mathrm{d}_{1,1} \\ \mathrm{d}_{2,0} &= \mathrm{d}_{2,6} \oplus \mathrm{d}_{2,1} \\ \mathrm{d}_{3,0} &= \mathrm{d}_{3,6} \oplus \mathrm{d}_{3,1} \end{aligned} \tag{10}$$

3.2. Storage system fault recovery of EVENODD code

Like RDP codes, EVENODD is a classical MDS code that can also correct errors at any two nodes. The EVENODD code is a two-dimensional array of size (m-1) * (m+2).

the relation between data disk and parity are as follow:

$$\mathrm{s} = \sum_{j=1}^{m-1} d_{m-1-j,j} \tag{11}$$

$$\mathrm{d}_{\mathrm{i,m}} = \sum_{j=0}^{m-1} d_{i,j} \tag{12}$$

$$\mathrm{d}_{\mathrm{i,m+1}} = \mathrm{s} \oplus \sum_{j=0}^{m-1} d_{<i-j>_m,j} \tag{13}$$

The sum is XOR sum. Where $\mathrm{d}_{\mathrm{i,j}}$, is used to represent row i and column j of the disk, <x>y=x mod y, s is the check factor (figure 8).

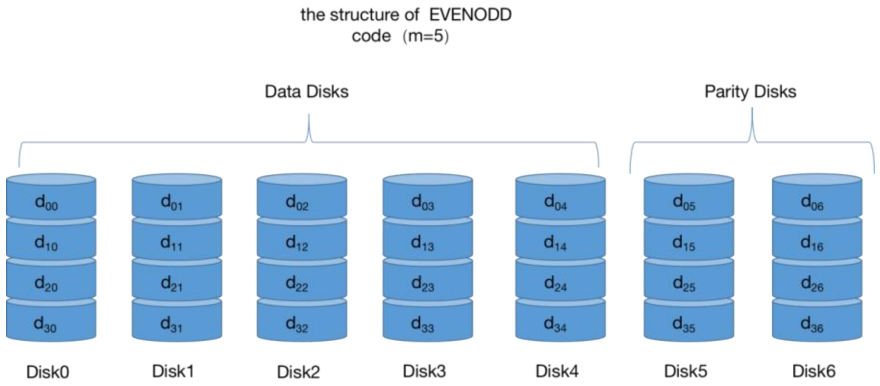

Figure 8. The structure of EVENODD coding (Photo/ Picture credit: Original).

The relation between parity disks and data disks are as follow:

$$\mathrm{s} = \mathrm{d}_{3,1} \oplus \mathrm{d}_{2,2} \oplus \mathrm{d}_{1,3} \oplus \mathrm{d}_{0,4}$$

$$d_{0,5} = d_{0,0} \oplus d_{0,1} \oplus d_{0,2} \oplus d_{0,3} \oplus d_{0,4}$$

$$
\begin{aligned}
d_{1,5} &= d_{1,0} \oplus d_{1,1} \oplus d_{1,2} \oplus d_{1,3} \oplus d_{1,4} \\
d_{2,5} &= d_{2,0} \oplus d_{2,1} \oplus d_{2,2} \oplus d_{2,3} \oplus d_{2,4} \\
d_{3,5} &= d_{3,0} \oplus d_{3,1} \oplus d_{3,2} \oplus d_{3,3} \oplus d_{3,4} \\
d_{0,6} &= s \oplus d_{0,0} \oplus d_{3,2} \oplus d_{2,3} \oplus d_{1,4} \\
d_{1,6} &= s \oplus d_{1,0} \oplus d_{0,1} \oplus d_{3,3} \oplus d_{2,4} \\
d_{2,6} &= s \oplus d_{2,0} \oplus d_{1,1} \oplus d_{0,2} \oplus d_{3,4} \\
d_{3,6} &= s \oplus d_{3,0} \oplus d_{2,1} \oplus d_{1,2} \oplus d_{0,3}
\end{aligned} \tag{14}
$$

When a single disk failure is encountered, it is clear that need at least five sets of disks to recover. While using the improved EVENODD coding scheme (figure 4), the recovery would be considerably easier and faster.

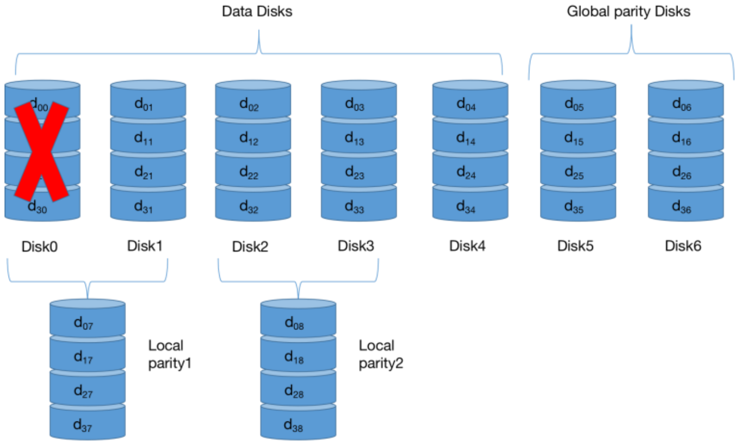

Figure 9. Single data disk failure in improved EVENODD coding scheme (Photo/ Picture credit: Original).

In this case need at least two sets of disks to recover the failure, Disk1 and local parity1. (The probability of this case is 80%)(figure 9).

$$
\begin{aligned}
d_{0,0} &= d_{0,1} \oplus d_{0,7} \\
d_{1,0} &= d_{1,1} \oplus d_{1,7} \\
d_{2,0} &= d_{2,1} \oplus d_{2,7} \\
d_{3,0} &= d_{3,1} \oplus d_{3,7}
\end{aligned} \tag{15}
$$

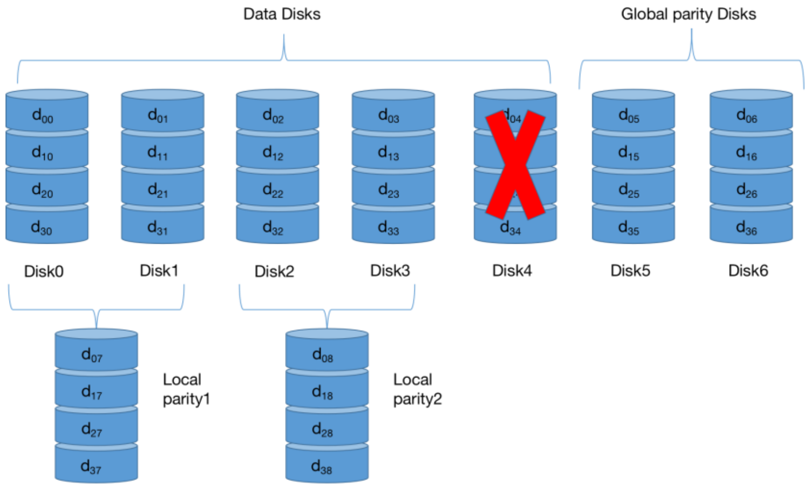

Figure10. Single data disk failure in improved EVENODD coding scheme (Photo/ Picture credit: Original).

While in this case need at least three sets of disks to recover the failure, Disk5,local parity1 and local parity2.(The probability of this case is 20%)(figure 10).

$$d_{0,4} = d_{0,5} \oplus d_{0,7} \oplus d_{0,8}$$
$$d_{1,4} = d_{1,5} \oplus d_{1,7} \oplus d_{1,8}$$
$$d_{2,4} = d_{2,5} \oplus d_{2,7} \oplus d_{2,8}$$
$$d_{3,4} = d_{3,5} \oplus d_{3,7} \oplus d_{3,8} \tag{16}$$

Apparently in this case, disk access has been reduced by 40% when recovering a disk failure. Therefore, the number of nodes that need to be accessed when recovering a single disk failure decrease by 0.80.5+0.20.4=48% on average.

4. Conclusion

In essence, this paper proposes a novel approach to enhancing data recovery and system stability in the event of single disk failure by leveraging local redundancy, an addition that aims to mitigate the limitations inherent in traditional erasure codes. The conventional erasure codes, while undoubtedly effective in their designated purpose, tend to visit an excessive number of nodes during the recovery process of a single disk failure. This considerable number of node visits can lead to a steep increase in data recovery time, raising the risk of potential system instability and multiple disk failures. Our proposed methodology offers a solution by introducing local redundancy into the system, effectively reducing the number of nodes visited during the data recovery process. This reduction can significantly improve data recovery times, thereby bolstering overall system stability. However, it's important to note that the addition of local redundancy into the system comes with its own set of trade-offs. While the system gains in terms of recovery speed and stability, this process concurrently consumes additional storage space, an aspect that can prove challenging in systems where storage efficiency is paramount. Furthermore, the introduction of local redundancy results in a decrease in the code rate. Despite these challenges, this approach can prove invaluable in scenarios where rapid recovery times and system stability are more crucial than storage efficiency and code rate optimization. Ultimately, this research provides a valuable foundation for future studies aiming to strike the perfect balance between system stability, recovery time efficiency, storage space optimization, and code rate in the field of data storage systems.

References

[1] Liu Z, Li Q, Chen X, et al. Point cloud video streaming: Challenges and solutions[J]. IEEE Network, 2021, 35(5): 202-209.

[2] Wang Y, Wang J, Zhang W, et al. A survey on deploying mobile deep learning applications: A systemic and technical perspective[J]. Digital Communications and Networks, 2022, 8(1): 1-17.

[3] Ji B, Wang Y, Song K, et al. A survey of computational intelligence for 6G: Key technologies, applications and trends[J]. IEEE Transactions on Industrial Informatics, 2021, 17(10): 7145-7154.

[4] Zhang D, Pee L G, Cui L. Artificial intelligence in E-commerce fulfillment: A case study of resource orchestration at Alibaba's Smart Warehouse[J]. International Journal of Information Management, 2021, 57: 102304.

[5] You X, Huang Y, Liu S, et al. Toward 6G Extreme Connectivity: Architecture, Key Technologies and Experiments[J]. IEEE Wireless Communications, 2023, 30(3): 86-95.

[6] Almalki M, Giannicchi A. Health apps for combating COVID-19: descriptive review and taxonomy[J]. JMIR mHealth and Health, 2021, 9(3): e24322.

[7] Nagarkar A A, Root S E, Fink M J, et al. Storing and reading information in mixtures of fluorescent molecules[J]. ACS Central Science, 2021, 7(10): 1728-1735.

[8] Wahab O F A, Khalaf A A M, Hussein A I, et al. Hiding data using efficient combination of RSA cryptography, and compression steganography techniques[J]. IEEE access, 2021, 9: 31805-31815.

[9] Usama M, Malluhi Q M, Zakaria N, et al. An efficient secure data compression technique based on chaos and adaptive Huffman coding[J]. Peer-to-Peer Networking and Applications, 2021, 14: 2651-2664.

[10] Wang D, Wang H, Fu Y. Blockchain-based IoT device identification and management in 5G smart grid[J]. EURASIP Journal on Wireless Communications and Networking, 2021, 2021(1): 125.