

# Comparison and Performance Analysis of EVENODD and RDP Fault-Tolerant Coding Methods

Haitao Zhan \*

Department of Computer Science, Beijing University of Technology, Beijing, 100124, China

\* Corresponding Author Email: zhanhaitao@emails.bjut.edu.cn

**Abstract.** Fault-tolerant coding schemes are techniques designed to protect original data from errors and data loss by introducing redundant data. This study compares the performance of two commonly used fault-tolerant coding methods, EVENODD and RDP. These methods are widely employed in data storage to ensure data integrity and fault tolerance. The objective of this research is to determine which method is better suited for specific application scenarios. Through experiments and performance evaluations, this research found that RDP outperforms EVENODD in terms of encoding and decoding, requiring fewer computational resources, especially excelling in encoding and decoding times. EVENODD exhibits better storage and I/O efficiency, typically requiring fewer data disks but demanding more computational resources for encoding and decoding. The choice between EVENODD and RDP should be based on specific application requirements and the balance of performance, storage, and computational resources. Furthermore, this research discusses their respective advantages and disadvantages and proposes future research directions, including hybrid fault-tolerant coding schemes, integration with cryptography, and combining them with artificial intelligence and deep learning technologies. These studies will contribute to enhancing the security, integrity, and reliability of data storage.

**Keywords:** RDP; EVENODD; fault-tolerant coding.

## 1. Introduction

In today's information age, the storage and integrity of data are of paramount importance. In large-scale data centers, cloud computing environments, and various application scenarios, data reliability and fault tolerance have become critical elements in system design. To address the challenges of data storage, RAID (Redundant Array of Independent Disks) systems and fault-tolerant coding schemes are widely employed. RAID systems aim to enhance data storage reliability and performance, while fault-tolerant coding schemes ensure data integrity through the distribution of redundant data and coding strategies.

RAID, or Redundant Array of Independent Disks, is a storage technology designed to improve data availability, fault tolerance, and performance by combining multiple hard drives. RAID systems employ various levels (such as RAID0, RAID1, RAID5, RAID6, etc.), each with its unique data distribution and redundancy strategies. These strategies may include data mirroring, striping, and parity checking, all aimed at preventing data loss and mitigating the impact of hard drive failures on the system. RAID systems have become a core component of many data centers and enterprise-level storage solutions, ensuring high data availability and fault tolerance[1].

Fault-tolerant coding schemes are techniques that protect original data from errors and data loss by introducing redundant data. They are widely used in data communication, storage, and transmission to ensure data integrity. EVENODD and RDP (row-diagonal parity) are two common fault-tolerant coding schemes[2]. EVENODD encodes data blocks to correct errors caused by hard drive failures. RDP, on the other hand, employs different coding strategies to enhance data integrity and fault tolerance. This study chooses to compare EVENODD and RDP to determine which fault-tolerant coding scheme is better suited for specific application scenarios, ensuring data security.

The primary motivation for this research lies in the fact that although RAID systems and fault-tolerant coding schemes are widely adopted, performance and reliability requirements vary across different application scenarios. Therefore, this research aims to compare EVENODD and RDP, seeking to find the most suitable solution for specific needs. Our research objectives include

evaluating the performance and reliability of these two schemes in data storage to provide better decision support for system designers and meet the demands of data integrity and availability. Through an in-depth examination and comparison, this research aims to offer valuable insights and guidance for the further development of the data storage domain.

In the following sections of this paper, this research will introduce the fundamental principles of EVENODD and RDP fault-tolerant coding schemes, conduct a comprehensive performance comparison, and explore the practical applications and advantages of EVENODD and RDP in real storage systems through experimental assessments. This research aspires to provide decision-makers in the data storage field with valuable information and support to ensure data security and integrity.

## 2. An Overview of RDP and EVENODD Coding Methods

### 2.1. Related Research

RDP is a coding method related to disk fault tolerance. Numerous studies have been dedicated to optimizing RDP coding to enhance the fault tolerance and performance of disk arrays. These studies focus on various aspects of RDP coding, including coding strategies, disk recovery, and performance optimization[3]. The application of RDP coding in disk arrays has attracted widespread research interest in addressing failures and performance challenges in data storage systems.

EVENODD is another coding method related to disk fault tolerance, designed to provide highly reliable data storage.[4] Multiple research efforts have been aimed at improving EVENODD coding to strengthen the fault tolerance capability of disk arrays. These studies typically concentrate on aspects such as the performance of EVENODD coding, recovery mechanisms, and data reliability. EVENODD coding holds potential applications in addressing disk failures and data integrity, thus garnering significant research interest[5], [6].

### 2.2. Introduction to the RDP Encoding Method

RDP encoding achieves fault tolerance in storage systems by employing a set of data disks and a diagonal parity disk. Data blocks are distributed on data disks following a specific layout, and both row parity blocks and diagonal parity blocks are computed for each data block. The presence of these two types of parity blocks allows the system to recover data in case of disk failures[7].

In RDP encoding, the parameter 'p' represents a prime number that defines the size and structure of the RDP array. The choice of 'p' has a significant impact on the performance and fault tolerance of the entire storage system. Typically, 'p' is chosen in one of the following ways:

- 1). Choosing  $p = 2n + 1$ , where 'n' is a positive integer. This choice creates an RDP array consisting of 2n stripes with diagonal parity between them. This selection is common in practical applications because it effectively defines the block size used in RDP.

- 2). Choosing another prime number 'p' that meets specific requirements. The specific choice may depend on system performance, storage capacity, and available hardware resources.

The selection of the 'p' parameter should strike a balance between storage capacity and data protection requirements. Larger 'p' values can provide more data protection but may require more storage space and computational resources. Therefore, choosing an appropriate 'p' value requires careful consideration of system needs and resource constraints.

### 2.3. Introduction to EVENODD Encoding Method

The core idea behind EVENODD encoding is to achieve efficient disk fault tolerance by minimizing redundancy. It employs a clever approach that requires only two redundant disks to tolerate two disk failures. This is a highly efficient design because it minimizes the need for additional storage while providing dual-disk fault tolerance, which is crucial in RAID architectures[8].

In EVENODD encoding, the main parameter is the number of information disks (denoted as 'm'). Here are some discussions regarding parameter settings and their effects:

1). Number of Information Disks ('m'). The performance and fault tolerance of EVENODD is influenced by the number of information disks ('m'). A larger 'm' value provides more fault tolerance but may increase the complexity of encoding and decoding. Choosing the optimal 'm' value typically requires a trade-off between performance and cost.

2). Symbol Size. EVENODD allows flexibility in the size of symbols, but the symbol size impacts storage capacity and the performance of encoding/decoding. Larger symbol sizes can enhance encoding/decoding efficiency but increase storage overhead.

3). Number of Redundant Disks. EVENODD requires only two redundant disks, making it an efficient design. Increasing the number of redundant disks can enhance fault tolerance but also raises storage costs and complexity.

4). Diagonal Position. The diagonal position determines when modifications are needed for all redundant symbols. For specific applications, special configurations may be necessary to ensure that the choice of diagonal position does not lead to performance degradation.

The fundamental principle of EVENODD encoding is to achieve efficient disk fault tolerance through minimal redundancy, using XOR operations, and maintaining independence between parity bits. Parameter settings include the number of information disks, symbol size, the number of redundant disks, and diagonal position, and these settings need to be balanced and configured based on specific application requirements.

### 3. EVENODD VS RDP

#### 3.1. Disk Requirements

EVENODD requires  $n$  data disks, where  $n$  is odd, to store actual data blocks. RDP, on the other hand, requires  $p + 1$  data disks, where  $p$  is a prime number greater than 2, to store the actual data block. EVENODD typically requires two parity disks to store parity information, whereas RDP requires two parity disks—one for storing row parity information and the other for storing diagonal parity information. The total disk requirements for both EVENODD and RDP are determined by the sum of data disks and parity disks.

#### 3.2. Fault Tolerance

Both RDP and EVENODD possess double-disk fault tolerance, meaning they can continue to operate and recover data even in the event of two simultaneous disk failures. This fault tolerance is designed to ensure high availability and data reliability, safeguarding data integrity in the presence of multiple disk issues.

#### 3.3. Computational Efficiency

The primary measure of computational efficiency revolves around the count of XOR operations necessary for establishing parity. In the case of EVENODD, when dealing with an array featuring  $n$  data disks, each containing  $n - 1$  data blocks, the computation of row parity entails  $(n - 1)(n - 1)$  XOR operations, and calculating the parity for  $n$  diagonals requires  $(n - 2)n$  XOR operations. Moreover, EVENODD demands an additional  $(n - 1)$  XOR operation to amalgamate the parity of a specific diagonal with the parity of the other  $(n - 1)$  diagonals to complete the computation of stored diagonal parity. This results in an overall computational expense of  $2n^2 - 3n$  XOR operations for constructing parity in EVENODD for an array with  $n(n - 1)$  blocks. Consequently, EVENODD necessitates  $2 - 1/(n - 1)$  XOR operations per block.

In contrast, RDP safeguards  $(p - 1)^2$  data blocks using  $2p^2 - 6p + 4$  XOR operations. To ensure the protection of  $n^2$  data blocks with  $n = p - 1$ ,  $2n^2 - 2n$  XOR operations can be achieved, adhering to the optimal ratio of  $2 - 2/n$ . Given that  $n$  exceeds 2, the XOR operation count in RDP is notably fewer than that in EVENODD. Consequently, RDP exhibits enhanced computational efficiency when compared to EVENODD.

### 3.4. I/O Efficiency

In the case of full stripe writes:

In RDP, performing a full stripe write operation involves writing two parity blocks along with  $p-1$  data blocks, requiring two additional disk I/O operations to write these two parity blocks.

Compared to RDP, EVENODD typically only requires writing one parity block. In EVENODD, each stripe usually contains only one parity block, which is used to check the content of all data blocks. Therefore, during a full stripe write operation, only this single parity block needs to be updated, reducing disk I/O operations.

In the case of partial stripe writes:

RDP can perform partial stripe writes by updating parity blocks using either addition or subtraction methods. This means that during partial stripe writes, RDP can partially update based on changes in data blocks without rewriting the entire stripe.

EVENODD may be less efficient in the case of partial stripe writes, as it typically requires rewriting the entire parity block.

EVENODD and RDP have their advantages and disadvantages in different aspects. The choice of which method to use should be based on specific storage requirements and performance considerations. If disk capacity and the computational efficiency of parity calculations are key factors, then RDP may be more suitable. If there is a higher demand for I/O efficiency, then EVENODD might be the better choice.

## 4. Applications

### 4.1. Application Scenarios in Storage Systems

In practical storage system applications, EVENODD and RDP offer unique advantages tailored to specific scenarios. To provide a more comprehensive understanding of their distinct strengths, this research will discuss them separately, emphasizing their applicability in various real-world scenarios.

EVENODD finds its niche in storage environments characterized by a high demand for I/O efficiency. This includes scenarios such as online transaction processing (OLTP) systems and large-scale database storage. In these contexts, the frequent occurrence of read-and-write operations necessitates swift response times. EVENODD excels by minimizing disk I/O operations, thus enhancing overall system performance. Furthermore, EVENODD is well-suited for virtualized environments, where efficient disk access by virtual machines is pivotal for maintaining system performance.

RDP, on the other hand, shines in scenarios where computational efficiency and robust data protection are paramount. Examples of such scenarios include scientific computing clusters and large-scale data analytics platforms. In these environments, workloads are often compute-intensive, making efficient data protection crucial. RDP excels by offering effective parity calculations with fewer XOR operations[9]. Additionally, for systems dealing with substantial data volumes, RDP's double-disk fault tolerance ensures a high level of data reliability, even in the face of hardware failures.

The selection between EVENODD and RDP should be guided by the specific requirements and application scenarios of the storage system in question. If the application demands superior I/O efficiency, such as OLTP systems or virtualized environments, EVENODD emerges as the preferred choice. Conversely, if computational efficiency and data protection are the top priorities, particularly in scientific computing or big data analytics, then RDP stands out as the more suitable option.

### 4.2. Choosing a Data Protection Scheme

When deciding between EVENODD and RDP as data protection schemes, the choice should be based on specific requirements. Here are some recommendations that can guide your selection:

- 1). Disk Capacity and Cost

If disk capacity is limited, and this research wants to minimize additional storage overhead, EVENODD may be the better choice as it typically requires fewer redundant disks.

2). Computational Efficiency

If computational efficiency in terms of encoding and decoding is a performance concern, especially for large datasets, RDP might be more suitable. It usually requires fewer XOR operations to construct parity.

3). I/O Efficiency

If you have high demands for I/O operations efficiency, especially during partial stripe writes, EVENODD may be a better fit. It is typically more efficient in partial stripe writes as it only needs to update one parity block.

4). Data Integrity

If data integrity is critical for your application, whether in data centers, cloud storage, or critical mission applications, double-disk fault tolerance is a must. In this case, both EVENODD and RDP are suitable choices[10].

5). Specific Application Requirements

Choose based on specific application needs. For example, in large-scale data analytics, RDP's computational efficiency might be more critical, while in archival storage, I/O efficiency and data integrity could be paramount.

6). Multi-Tiered Storage

If your storage environment follows a multi-tiered storage approach, you can selectively apply EVENODD and RDP based on the requirements of different tiers to balance performance and data protection at different levels.

7). Hybrid Storage Solutions:

For storage environments that require a balance between performance and capacity, consider designing hybrid storage solutions where EVENODD and RDP can complement each other, providing a balance between computational efficiency and I/O efficiency.

## 5. Experimental Design and Performance Evaluation

### 5.1. Objective and Methodology

The experiment aims to compare the performance of EVENODD and RDP in terms of encoding/decoding speed, computational efficiency, etc. This helps determine which method is better suited for specific application scenarios. Through the experiment, this research tests the performance of EVENODD and RDP in handling disk failures to verify whether they can protect data integrity.

This research uses Python to construct EVENODD and RDP code, ensuring an equal number of disks. The values of  $n$  and  $p$  are generated using random functions. Random 16-bit binary data is fed into the system, and time measurements are taken after encoding and decoding, repeated 1000 times. The average time for encoding and decoding is then compared. Furthermore, the XOR count for each block of EVENODD and RDP is calculated to assess changes in computational efficiency as the number of disks varies.

### 5.2. Data Collection

In the experiment, the collected times and average times are of float type, obtained using the `time` function. The XOR counts for EVENODD and RDP are of int type, and the disk failure rate is of float type.

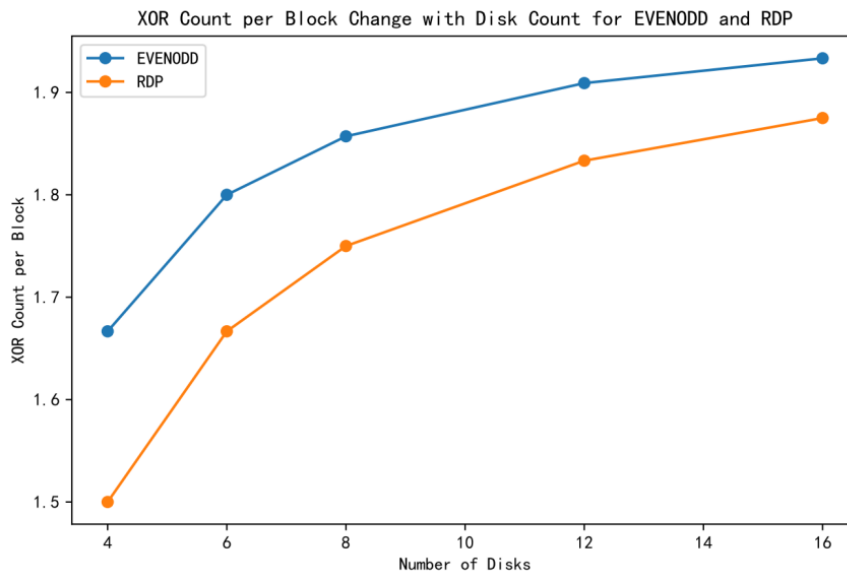
### 5.3. Performance Metrics Analysis

After conducting experiments, the following data and charts were obtained.

**Table 1.** Comparison of RDP and EVENODD encoding/decoding and error rates.

| Algorithm | Encoding Time          | Decoding Time          | Failure Rate |
|-----------|------------------------|------------------------|--------------|
| RDP       | 2.8930187225341797e-06 | 3.1590700149536132e-06 | 0.0          |
| EVENODD   | 4.9993991851806643e-06 | 4.9977302551269532e-06 | 0.0          |

From Table 1, it can be observed that the average encoding time of RDP is less than EVENODD, and the average decoding time of RDP is also less than EVENODD. This indicates that RDP outperforms EVENODD in terms of encoding and decoding efficiency, requiring fewer computational resources. Additionally, after multiple experiments, both algorithms showed an error rate of 0.0%, indicating that both RDP and EVENODD can effectively handle disk failures of up to two disks.



**Fig 1.** XOR Count per Block Change with Disk Count for EVENODD and RDP.

According to Figure 1, it is evident that as the number of disks increases, both RDP and EVENODD exhibit an increase in XOR operations. The XOR operations per block for both RDP and EVENODD increase, and they both show a rising trend with similar shapes. However, with the growth in the number of disks, the increase in XOR operations per block becomes slower, gradually approaching stability. As the number of disks increases, RDP's XOR operations per block remain consistently lower than EVENODD's, indicating that RDP is more computationally efficient than EVENODD, which aligns with theoretical expectations.

## 6. Conclusion and Outlook

### 6.1. Summary of Comparison Results

Taking the comparison results into consideration, RDP demonstrates superior performance and efficiency in encoding and decoding, requiring fewer computational resources, particularly excelling in encoding and decoding times. Additionally, RDP exhibits a lower number of XOR operations, signifying its computational efficiency. However, RDP typically demands more data disks, which might lead to increased storage space usage. On the other hand, EVENODD showcases better storage and I/O efficiency, as it usually requires fewer data disks. Nevertheless, it necessitates more computational resources for encoding and decoding. In summary, the choice between EVENODD and RDP should depend on specific application requirements and the trade-off considerations regarding performance, storage, and computational resources.

## 6.2. Summary of Advantages and Disadvantages of Each Encoding Method

### 6.2.1. Advantages and Disadvantages of EVENODD

EVENODD generally requires fewer data disks, reducing storage costs. During full stripe writes, EVENODD only needs to update a single parity block, reducing disk I/O operations. EVENODD employs a minimal redundancy design, providing dual-disk fault tolerance.

EVENODD demands more computational resources for encoding and decoding, which may impact performance. Special Configuration Required: The selection of diagonal positions may require special configurations to ensure optimal performance.

### 6.2.2. Advantages and Disadvantages of RDP

RDP typically outperforms in encoding and decoding, requiring fewer computational resources. RDP exhibits a lower number of XOR operations, enhancing computational efficiency.

Increased Storage Space Usage: RDP usually necessitates more data disks, potentially raising storage costs. During full stripe writes, RDP requires additional disk I/O operations to write parity blocks.

## 6.3. Future Research Directions

In the future, the potential exists for the exploration of combining the strengths of EVENODD and RDP to develop novel hybrid fault-tolerant coding schemes that can accommodate the demands of diverse application scenarios. Additionally, it is imperative to delve into more sophisticated fault-tolerant coding techniques to offer enhanced safeguards against disk failures, encompassing a wider spectrum of potential disk malfunction scenarios. Expanding on these fault-tolerant coding approaches, the incorporation of cryptography can be pursued to bolster the confidentiality and integrity of fault-tolerant coding, adding complexity to potential information compromise. Moreover, the integration of fault-tolerant coding with artificial intelligence can be explored to facilitate data sharing and collaboration within distributed AI systems. When certain nodes experience issues, fault-tolerant coding can maintain system consistency and availability. By harnessing deep learning technology, the encoding and decoding procedures of fault-tolerant coding can be improved, allowing for the utilization of neural networks in the creation of fresh error-correcting codes and the formulation of adaptive fault-tolerant approaches to elevate fault-tolerant efficiency.

## References

- [1] A. Thomasian, G. Fu, and C. Han, "Performance of Two-Disk Failure-Tolerant Disk Arrays," *IEEE Transactions on Computers*, vol. 56, no. 6, pp. 799–814, Jun. 2007, doi: 10.1109/TC.2007.1041.
- [2] J. Li, P. Li, R. J. Stones, G. Wang, Z. Li, and X. Liu, "Reliability Equations for Cloud Storage Systems with Proactive Fault Tolerance," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 4, pp. 782–794, Jul. 2020, doi: 10.1109/TDSC.2018.2882512.
- [3] G. Duchrau and M. Gössel, "A New Decoding Method for Double Error Correcting Cross Parity Codes," in *2022 IEEE 28th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, Sep. 2022, pp. 1–5. doi: 10.1109/IOLTS56730.2022.9897284.
- [4] S. Huang, H. Hou, and X. Yu, "A Lower Bound on Disk Reads for Single Information Disk Failure Recovery and One Recovery Scheme for EVENODD(p, 3)," in *2019 Ninth International Workshop on Signal Design and its Applications in Communications (IWSDA)*, Oct. 2019, pp. 1–5. doi: 10.1109/IWSDA46143.2019.8966126.
- [5] H. Hou and P. P. C. Lee, "A New Construction of EVENODD Codes With Lower Computational Complexity," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1120–1123, Jun. 2018, doi: 10.1109/LCOMM.2018.2820007.
- [6] H. Hou, Y. S. Han, K. W. Shum, and H. Li, "A Unified Form of EVENODD and RDP Codes and Their Efficient Decoding," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5053–5066, Nov. 2018, doi: 10.1109/TCOMM.2018.2859956.

- [7] “Row-diagonal parity for double disk failure correction | Proceedings of the 3rd USENIX conference on File and storage technologies.” Accessed: Oct. 13, 2023. [Online]. Available: <https://dl.acm.org/doi/10.5555/1973374.1973375>
- [8] M. Blaum, J. Brady, J. Bruck, and J. Menon, “EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures,” *IEEE Transactions on Computers*, vol. 44, no. 2, pp. 192–202, Feb. 1995, doi: 10.1109/12.364531.
- [9] G. Zhang, K. Li, J. Wang, and W. Zheng, “Accelerate RDP RAID-6 Scaling by Reducing Disk I/Os and XOR Operations,” *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 32–44, Jan. 2015, doi: 10.1109/TC.2013.210.
- [10] “Optimal recovery of single disk failure in RDP code storage systems | Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems.” Accessed: Oct. 02, 2023. [Online]. Available: <http://dl-acm-org-s.libziyuan.bjut.edu.cn:8118/doi/10.1145/1811039.1811054>