A Survey of Data Repair Methods for Erasure Codes in Heterogeneous Environments (Post-print)

Authors: Zhong Fengyan, Wang Yan, Li Nianshuang

Date: 2018-07-23T00:00:00+00:00

Abstract

In large-scale cloud storage systems, storage node failures caused by disk or network faults occur frequently, necessitating data redundancy techniques to ensure data reliability and availability. Currently, most research on erasure code-based redundant data repair treats each storage node indiscriminately; however, in practical distributed storage systems, nodes typically exhibit heterogeneity in bandwidth resources, computational resources, storage capacity resources, and other aspects, and such resource heterogeneity significantly impacts redundant data repair performance. This work identifies the key factors affecting repair performance, selecting bandwidth overhead, disk access overhead, repair time, number of nodes participating in repair, and repair cost as evaluation metrics for repair performance; analyzes how existing research methods reduce these five types of overhead, with focused discussion on the advantages and disadvantages of these methods; expounds upon the current research status of erasure code repair techniques in heterogeneous distributed storage systems; and finally points out some unresolved challenges in erasure code data repair technology and potential future development directions for erasure code repair techniques.

Full Text

A Survey of Heterogeneous-based Data Repair Strategies for Erasure Codes

Zhong Fengyan, Wang Yan, Li Nianshuang
School of Software, East China Jiaotong University, Nanchang 330013, China

Abstract: In large-scale cloud storage systems, storage node failures due to disk or network faults occur frequently, necessitating data redundancy techniques to ensure data reliability and availability. Most existing research on erasure code-based redundant data repair treats all storage nodes indiscriminately. However,

in practical distributed storage systems, nodes typically exhibit heterogeneity in bandwidth resources, computing resources, and storage capacity, which significantly impacts redundant data recovery performance. This paper identifies the key factors affecting repair performance and selects bandwidth overhead, disk access overhead, repair time, the number of nodes participating in repair, and repair cost as evaluation criteria for repair performance. It analyzes how existing research methods reduce these five types of overhead, focusing on the advantages and disadvantages of these methods. The paper also elaborates on the current research status of erasure code repair technology in heterogeneous distributed storage systems. Finally, it points out some unresolved challenges in erasure code data repair technology and possible future development directions for erasure code repair methods.

Keywords: storage systems; erasure code; heterogeneity; data recovery; performance improvement

0 Introduction

As storage systems increase in scale and nodes become more diverse and complex, node failures occur frequently. To combat data loss caused by node failures, distributed storage systems must maintain a certain amount of redundant data to ensure reliability and availability. Two primary techniques generate redundant data: replication and erasure codes. Replication technology stores multiple copies of data across different nodes. When a storage node containing a replica fails, the distributed storage system automatically switches service to other replicas, achieving high reliability and availability. This approach does not involve specialized encoding and reconstruction algorithms and offers good fault tolerance, but its storage utilization is extremely low. As data continues to grow, multi-replication technology introduces enormous storage overhead at the PB level in data centers. For example, existing distributed storage systems such as HDFS (Hadoop Distributed File System) and Ceph typically employ triple replication, consuming three times the original file size in storage space.

In response to this situation, erasure code technology has been widely adopted due to its low storage overhead advantage. Among these, RS (Reed-Solomon) codes are currently the most widely used erasure code scheme. The encoding process generates m parity blocks from k data blocks according to specific coding rules. For these k+m encoded blocks, the coding property guarantees that the original file can be reconstructed from any k encoded blocks. Taking RS(4,2) coding as an example, the original file is divided into k=2 parts, and m=2 parity blocks are generated according to RS coding rules, providing fault tolerance of 2. The data collector can reconstruct the original file by selecting any 2 nodes, and this coding method only consumes 1.5 times the original file size in storage space while providing the same fault tolerance as triple replication.

Although erasure code technology offers low storage overhead, it suffers from high repair costs. When a node fails, the system must repair the data blocks

on the failed node and place them on other normal nodes to maintain system redundancy. Repairing a data block through replication simply copies the corresponding data from other normal nodes, whereas erasure code methods require downloading data blocks from multiple supplier nodes to repair lost data. This results in high network bandwidth overhead and long repair times. In large-scale cluster environments, disk, server, and network failures have become the norm. Rashmi et al. monitored node failures in a Facebook data center cluster using (10,4) RS erasure codes. The monitored cluster contained over 3,000 nodes, each storing 15 TB of data, for a total of 30 PB. According to their monitoring results, in such a cluster, an average of more than 20 storage nodes fail daily, with the number reaching as high as 100 in a single day. Under these circumstances, to ensure high reliability and availability, storage systems must perform frequent repair operations, increasing system pressure.

In response to the performance deficiencies of erasure code data repair, much theoretical design and engineering implementation work has been done in homogeneous scenarios (treating each storage node indiscriminately), assuming consistent bandwidth resources, computing resources, and storage capacity for each node. Currently, there are a few research surveys on erasure code data repair technology. Reference [8] mainly introduces the development status of typical and common erasure code technologies. Reference [9] focuses on the latest research progress in coding schemes, data repair, and data updates. Reference [10] discusses key technologies for optimizing erasure code repair performance in homogeneous environments from three aspects: computation, read/write, and transmission. This paper focuses exclusively on the problem of optimizing erasure code data repair performance in heterogeneous distributed storage systems. We believe that heterogeneous distributed storage systems affect erasure code data repair performance mainly in three aspects: bandwidth heterogeneity between storage nodes, computing capability heterogeneity of storage nodes, and storage capacity heterogeneity of storage nodes. Therefore, this paper categorizes data repair methods in heterogeneous environments into three types: bandwidth-heterogeneity-oriented, computing-capability-heterogeneity-oriented, and storage-capacity-heterogeneity-oriented repair methods. The paper discusses existing erasure code data repair methods in terms of five metrics: repair bandwidth overhead, disk access overhead, repair time, number of nodes participating in repair, and repair cost. Finally, it points out possible future research directions for erasure code data repair methods.

1.1 Basic Concepts

For ease of understanding, a server is often referred to as a node. The relevant concepts appearing in this paper are explained as follows:

a) **Data block:** The smallest encoding unit formed by partitioning original user data.

b) **Parity block:** The result obtained from encoding operations on original

data blocks.

c) **Stripe:** A redundant set composed of multiple data blocks and their corresponding parity blocks. If a certain number of encoded blocks are lost, they can be regenerated through operations on the remaining encoded blocks in the stripe.

d) **MDS codes:** MDS (Maximum Distance Separable) codes are space-optimal codes. An (n,k) MDS code can divide the original file into k blocks and encode them into n encoded blocks, each of size 1/k of the original file, where any k encoded blocks can reconstruct the original file. This property is called the MDS property. Any code satisfying the MDS property can be called an MDS code.

e) **Supplier node:** A node that participates in data repair. Supplier nodes read local data and transmit it to other nodes via the network to participate in data reconstruction.

f) **Newcomer node:** A node that reconstructs lost data. It needs to collect required data from supplier nodes and compute the lost data.

g) **Bottleneck bandwidth:** The link with the smallest bandwidth in the repair topology, which determines the time required for the repair process.

1.2 Factors Affecting Repair Performance

The data repair process generally consists of four steps:
a) Supplier nodes read required data from local disks.
b) To reduce data transmission volume, supplier nodes perform local random linear combinations on the data to generate transmission data.
c) Supplier nodes transmit the transmission data through the network to the newcomer node.
d) After receiving all data from supplier nodes, the newcomer node decodes and recovers the lost data.

By analyzing the data repair process, in heterogeneous distributed storage systems, the main factors affecting erasure code data repair performance are threefold: bandwidth heterogeneity between storage nodes, computing capability heterogeneity of storage nodes, and storage capacity heterogeneity of storage nodes. Data repair time is jointly affected by node read/write capability, computing capability, and transmission capability. Since most distributed systems are built on inexpensive servers interconnected via networks, and both server nodes and networks are unreliable, the CPU computing capability, disk performance, and network card speed of server nodes can all become bottlenecks limiting repair time.

Based on the above analysis, we believe that bandwidth heterogeneity, com-

puting capability heterogeneity, and storage capacity heterogeneity between storage nodes are the main factors affecting erasure code repair performance. Therefore, existing data repair methods are categorized into bandwidth-heterogeneity-oriented, computing-capability-heterogeneity-oriented, and storage-capacity-heterogeneity-oriented repair methods. Repair bandwidth overhead, disk I/O overhead, number of nodes participating in repair, repair time, and repair cost are selected as evaluation criteria for erasure code repair performance.

2 Bandwidth-Heterogeneity-Oriented Repair Methods

Bandwidth heterogeneity refers to the fact that link bandwidth between storage nodes is not always equal. Reference [11] points out that in practical distributed storage systems, link bandwidth between storage nodes exhibits heterogeneity, with some links having high bandwidth and others low bandwidth. Since each supplier node transmits equal amounts of data to the newcomer node, when the link bandwidth between each supplier node and the newcomer node differs, the time to complete one round of data repair is determined by the link with the smallest bandwidth (data transmission between nodes can be parallel). Therefore, a natural idea is to dynamically adjust the amount of data each supplier node transmits to the newcomer node according to link bandwidth, allowing high-bandwidth links to transmit more data and low-bandwidth links to transmit less. Existing research work follows this approach to optimize data repair performance. This section introduces three types of data repair strategies under bandwidth heterogeneity, including elastic repair strategies under star topology, repair strategies under tree topology, and XOR-based erasure code repair technology, and analyzes the performance of these methods in terms of repair bandwidth overhead, repair time, number of nodes participating in repair, and repair cost.

2.1 Star Topology-based Elastic Repair Methods

Star Structure Based Serial Repair Strategy (SSR) [12] refers to the situation where when multiple nodes fail simultaneously, the system repairs the failed nodes serially, reconstructing multiple redundant data nodes to restore the original redundancy level. When constructing each redundant data node, the system builds a star structure centered on the newcomer node with supplier nodes at the boundaries, where all supplier nodes directly transmit data to the newcomer node. In this structure, the regeneration time is determined by the slowest bandwidth link between the newcomer node and supplier nodes. Four star topology-based repair methods are introduced below.

2.1.1 Maximum Elasticity Selection for Elastic Repair Elastic repair dynamically determines the corresponding data transmission volume based on available bandwidth size in each round of repair. Specifically, high-bandwidth links transmit more data, while low-bandwidth links transmit less. Dimakis et

al. [13,14] introduced a strong constraint in the regenerating code data repair process: each supplier node transmits equal amounts of data to the newcomer node for data repair, and the data collector (DC node) only connects to k nodes, downloading data volume from each node. In this case, newcomer nodes or DC nodes that could utilize more links can only use a portion of them, potentially wasting some high-bandwidth link resources and possibly leading to inefficient repair. Shah et al. [15] proposed an elastic repair strategy that allows supplier nodes and DC nodes to fully utilize available link resources to transmit unequal amounts of data to the newcomer node. Specifically, the DC node downloads _i data from nodes i=1,⋯,n, satisfying the condition that the total download volume is not less than M (the size of the original file). Similarly, the newcomer node downloads _i data from supplier nodes i=1,⋯,n, satisfying the condition that the total received volume is greater than or equal to a set parameter M, which can be expressed by the following two inequalities:

kM) 1(nii) 0(ii()nii 1()max 0 ii ,01nMiii ,01niii kMkM

When a single node fails, the newcomer node selects any k nodes and downloads data volume from each, with repair bandwidth equal to the size of the original file. This repair mode may not utilize some high-bandwidth links. Using an elastic strategy, the newcomer node can download unequal amounts of data from different nodes based on available bandwidth size to reduce regeneration time.

2.1.2 Optimal Node Selection Repair Method Existing literature on reducing regeneration time typically employs two approaches: one is reducing data transmission volume, and the other is transforming the topology of the regeneration process. However, reference [16] argues that different supplier nodes participating in the data repair process also affect repair performance. Therefore, for two scenarios: (a) supplier nodes are determined while the newcomer node is undetermined, and (b) both supplier nodes and newcomer nodes are undetermined, they propose FPSN and SPSN algorithms to select optimal supplier nodes. The FPSN algorithm fixes d supplier nodes and selects an optimal newcomer node to form a repair topology with maximum bottleneck bandwidth. The SPSN algorithm traverses all link bandwidths to find the repair topology with maximum minimum link bandwidth among all possible repair topologies composed of d supplier nodes and one newcomer node. Additionally, the literature designs a FLEX algorithm to calculate the amount of data each supplier node transmits to the newcomer node for the second scenario. Practice has proven that using the node selection scheme proposed in the literature can reduce average repair time by 58.56%.

2.1.3 Trade-off between Download Cost and Repair Bandwidth in Distributed Storage Systems Due to bandwidth heterogeneity among different types of storage nodes in distributed storage systems, the cost of downloading a data block may vary when repairing failed data, i.e., download cost hetero-

geneity. Akhlaghi et al. [17] assume that the system has two types of download costs, C1 and C2, where nodes of the same download cost type form a group, and nodes in different groups have different download costs. Using information flow graphs, they propose Generalized Regenerating Codes (GRC) and theoretically compare the repair cost and download cost of Generalized Minimum Storage Regenerating codes (GMSR) and Minimum Storage Regenerating codes (MSR), as well as Generalized Minimum Bandwidth Regenerating codes (GMBR) and Minimum Bandwidth Regenerating codes (MBR). Under certain specific conditions, GRC codes are superior to RC codes. However, the literature only addresses the trade-off relationship between download cost and repair bandwidth, without providing a data repair scheme for heterogeneous download cost scenarios.

2.1.4 Low-cost Multi-node Failure Repair Method for Erasure Codes

The star topology-based repair methods introduced above only discuss single-node failure repair problems, without mentioning multi-node failure repair issues. In practical systems, multi-node failure situations occur frequently [18,19]. Zheng et al. [20] propose a low-cost multi-node failure repair method for erasure codes, which uses a serial repair approach to complete the repair of multiple failed nodes sequentially, using network distance as the basis for node selection. They argue that nodes with shorter network distances have higher bandwidth between them, and vice versa. The specific repair process is as follows: assuming r nodes fail, during repair operations, r newcomer nodes are first selected, then k nodes are chosen from the n-r remaining nodes. These k nodes must satisfy the condition that their total network distance to the r newcomer nodes is the shortest. After determining the supplier nodes, one of these newcomer nodes is selected as the central node, which communicates simultaneously with both other newcomer nodes and supplier nodes. After determining the central node, it receives data volume from each of the k supplier nodes. Supplier nodes only need to transmit data once, and the central node can complete the construction of r failed blocks, finally storing one corresponding data block locally and sending the remaining r-1 data blocks to the other r-1 newcomer nodes respectively. From this repair process, it can be analyzed that the selection of the central node needs to consider both its network distance to supplier nodes and to other newcomer nodes. This method of selecting supplier nodes and central nodes based on network distance can improve available bandwidth between nodes and reduce the burden of measuring available bandwidth between nodes for the system. Additionally, using multi-threaded computation and pipelined data transmission to organize data computation and transmission, and using a central node-based data repair method to simultaneously repair multiple failed data blocks, greatly reduces bandwidth overhead.

2.2 Tree Topology-based Repair Methods

The star topology-based repair method is relatively simple, but its disadvantage is that the central node simultaneously bears computation and transmission tasks. To improve data transmission efficiency, Li et al. [21,22] designed a

new transmission topology—the tree topology—to replace the star topology used in current repair processes, thereby improving data transmission speed during data repair. To maximize network link resource utilization, they modeled the link selection for tree topology as a bottleneck spanning tree problem in graph theory and proposed corresponding optimal algorithms to solve it. They first considered the case where the number of supplier nodes is k, i.e., typical MDS codes. Subsequently, they continued to consider the tree repair problem for MSR codes and building multiple trees in parallel to address situations where bidirectional link bandwidths differ. The following three subsections introduce the work of Li et al. and discuss the advantages and disadvantages of their work.

2.2.1 Regeneration Process for Symmetric Link Single Tree In symmetric networks, where uplink and downlink bandwidths between nodes are equal, Li et al. constructed a regeneration process for symmetric link single trees. They used Prim' s algorithm to construct the optimal regeneration tree, with each link transmitting data volume, and allowed intermediate nodes to perform encoding on data blocks in advance. By establishing a tree-structured transmission path to increase bottleneck bandwidth, the goal of reducing repair time overhead was achieved. The tree repair method proposed by Li et al. can well adapt to scenarios with inconsistent available bandwidth in practical distributed storage systems, greatly saving repair time for storage systems. However, Wang et al. [23] discovered through experiments that the tree repair strategy for MSR codes proposed by Li et al. cannot well guarantee data integrity during repair. Because simultaneously changing the repair topology structure and allowing intermediate nodes to encode leads to insufficient information transmitted during repair operations to reconstruct the lost data. Therefore, Wang et al. analyzed the information flow graph of tree repair, obtained the minimum amount of information that needs to be transmitted on each edge during repair operations, not only well compensating for the insufficient information problem in Li et al.' s tree MSR codes but also proposing a new elastic repair strategy also applicable to distributed storage systems with bandwidth heterogeneity.

2.2.2 Regeneration Process for Asymmetric Link Single Tree In asymmetric networks, where uplink and downlink bandwidths between nodes are inconsistent, Lee et al. [24] pointed out that only 21.49% of edges can be considered to have symmetric bidirectional links. If all links are treated as symmetric, for example, the bottleneck bandwidth of the optimal regeneration tree in Figure 1Figure 1: see original paper is 30 Mbps. However, if links are treated as asymmetric, as shown in Figure 1(b), the bottleneck bandwidth can only reach 15 Mbps. Therefore, treating link bandwidth as asymmetric better reflects actual network conditions, and constructing optimal regeneration trees in asymmetric networks can more truly improve bottleneck bandwidth. The bottleneck bandwidth of the optimal regeneration tree in Figure 1(c) can reach 20 Mbps. The regeneration process for asymmetric link single trees is similar to the repair process under symmetric links, and the problems existing in this

method are similar to those under symmetric links.

2.2.3 Regeneration Process for Asymmetric Link Multi-tree Parallel Transmission In asymmetric links, constructing multiple trees for parallel transmission can further reduce regeneration time. Therefore, constructing multiple regeneration trees can utilize more link bandwidth, reduce regeneration time, and improve regeneration efficiency. For example, Figure 2Figure 2: see original paper shows a two-tree parallel regeneration process utilizing 5 links, which further reduces regeneration time compared to single-tree regeneration. Additionally, if edges in multiple regeneration trees are allowed to share network links, for example, in Figure 2(b) where two trees each handle half of the regeneration traffic, the bottleneck bandwidth can be increased to 30 Mbps. Although multiple trees can fully utilize available bandwidth, similar problems still exist.

2.2.4 Pipelined Repair Technology for Storage Systems Using Erasure Codes Although using erasure codes to generate redundant data improves storage efficiency, it has the disadvantage of high repair costs. Specifically, repairing an unavailable encoded block requires reading multiple available encoded blocks. Compared to normal reads, reading additional data blocks not only increases read time but also consumes bandwidth resources of other foreground servers. Therefore, in practice, erasure coding is mainly used for storing data that does not need to be read frequently (cold data), while frequently accessed data (hot data) is stored using replication. Replication only requires simply reading the corresponding replica from other available nodes, maintaining efficient access speed. To reduce erasure code repair time, many research efforts have proposed new coding schemes or designed new repair methods. Although these methods effectively reduce repair time, repair time remains higher than normal read time.

Based on this, Li et al. [25] proposed a new pipelined repair technology that can be applied in both homogeneous and heterogeneous environments. Their approach transforms the repair process of a failed block into the repair process of multiple slices. Specifically, the block is evenly divided into s slices of equal size. For example, in a (14,10) RS coding system, setting a block size of 64 MiB, when pipelining the repair of a failed block, the failed block is divided into 2,048 slices, each 32 KiB in size, with the repair processes of each slice organized in a pipeline manner. In homogeneous environments, using pipelined repair technology can quickly solve the degraded read problem for a single block within a stripe, with repair time being (s+1)/s time slots. From this expression, it can be seen that as s increases, repair time approaches one time slot, meaning degraded read time approaches normal read time. Moreover, pipelined repair technology also solves repair problems across multiple stripes. When repairing blocks in different stripes, the greedy scheduling method is used to add a timestamp symbol to each node to track the most recent time it was selected as a supplier node, aiming to balance the load of each supplier node. Figure 3[Figure 3: see

original paper] shows the process of repairing one block in an RS(6,4) coding system, where the failed block is divided into 6 slices denoted as S1,S2,⋯,S6, with every 3 slices forming a group, represented by Group1 and Group2. In the process of repairing one data block, the first stage consumes 0.5 time slots, and in the second stage, the last supplier node on each path simultaneously transmits slices to the requester (R), also consuming 0.5 time slots.

In heterogeneous environments, where link bandwidth between each pair of nodes differs, the literature proposes a weighted path optimal selection algorithm for path selection. This algorithm can find the path with maximum bandwidth in a very short time. For example, in a (14,10) MDS coding system, using enumeration to search all paths to find the optimal path takes an average of 27 seconds, while the algorithm proposed in the literature only needs 0.9 seconds to search for the optimal path.

2.2.5 Data Repair Method Based on Transmission Cost Heterogeneity Transmission cost refers to the cost of transmitting one element over a single link between adjacent nodes, which is not always equal across different links. Additionally, different types of network topology structures affect total repair cost, thereby influencing repair scheme selection. Based on this, Akhlaghi et al. [26] considered a relatively simple scenario, assuming the system has two types of nodes, S1 and S2, each corresponding to different communication costs C1 and C2. According to regenerating code requirements that a newcomer node must connect to d supplier nodes during repair, assuming d1 nodes are selected from the node set with communication cost C1, and d2 = d - d1 nodes are selected from the node set with communication cost C2, the total repair cost CT required to repair one newcomer node can be expressed as: CT = d1×C1× + d2×C2× . On this basis, they provide the trade-off relationship between repair cost and repair bandwidth. The limitation of this method is that the system only has two types of nodes with communication costs, whereas in actual systems, communication costs may be diverse. Additionally, it does not consider the impact of network topology structure on the repair process. Li et al. considered the impact of network structure on the repair process and proposed an optimal tree repair strategy for regenerating codes. Gerami et al. [27] considered four typical network models: serial networks, star networks, grids, and fully connected networks, proposing a cooperative regeneration scheme for available nodes (SNC) to reduce repair costs. They proposed joint and separate methods to optimize repair costs. The main idea of the joint method is to construct regenerating codes with minimum repair cost that satisfy MDS properties. The separate method utilizes MDS properties, analyzes information flow graphs to find feasible regions, and transforms the optimization of repair cost into a linear programming problem. Using the SNC scheme to repair failed data can fully utilize network topology structure to achieve the goal of reducing repair costs. Their research limitation lies in using the same assumption as typical regenerating codes, i.e., the newcomer node downloads equal amounts of data from each supplier node.

2.3 XOR-based Erasure Code Repair Methods

The low network load data repair technology for XOR-based erasure codes [28] was first proposed by Xiang Liping et al. to optimize the amount of data transmitted during RDP code [29] repair. Khan et al. generalized it to apply to any XOR-based erasure code [30]. Each data block in XOR-based erasure codes can be viewed as composed of multiple data slices of equal size, where parity slices in parity blocks are generated through XOR operations on certain data slices. Three data repair technologies based on XOR operations are introduced below: the EG algorithm repair strategy for improving degraded read performance, the PHR algorithm repair strategy based on RAID-6 codes, and the data repair technology based on download cost heterogeneity, with performance analysis.

2.3.1 EG Algorithm Repair Method for Improving Degraded Read Performance There are two types of node failures: permanent failure and temporary failure. Permanent failure refers to lost data on a node, while temporary failure refers to data that is not lost but temporarily unreadable. For the former, the system performs failure recovery; for the latter, the system performs degraded reads. Degraded read operations must read both available and unavailable data. When unavailable data needs to be read, the system must perform corresponding data recovery operations. The encoding process using erasure codes can be represented by matrices. When a single node fails, the encoded blocks stored on available nodes can still be represented by an encoding submatrix, and the data recovery process involves computing the corresponding decoding submatrix. Considering node bandwidth heterogeneity, Zhu et al. [31] proposed the Enumerated Greedy algorithm (EG algorithm). The EG algorithm traverses all possible combinations of d available nodes. Under each combination, there are C(d,k) CDREs (Candidate Decoding Recovery Equations). It calculates the degraded read time for each CDRE and updates the degraded read time to minimize the degraded read time for each block.

Assuming a (k,m,w) erasure code system with n (n=k+m) storage nodes, where k represents the number of data nodes, m represents the number of parity nodes, and w represents the number of encoded blocks in a stripe. If f nodes fail, according to MDS properties, selecting any k nodes from n-f available nodes yields C(n-f,k) decoding submatrices, so each data block has C(n-f,k) CDREs. If there are l degraded read requests for l data blocks, then l CDREs are needed to regenerate l data blocks, making the solution space domain C(n-f,k)^l. The EG algorithm proposed in the literature can find the optimal degraded read sequence within a reasonable time, with time complexity O(C(n-f,k)). Compared to the basic approach, the EG algorithm can reduce degraded read time by 32.70%.

2.3.2 PHR Algorithm Repair Method Based on RAID-6 Codes RAID (Redundant Array of Independent Disks) coding, or redundant disk array technology, has become an important industry standard. Various RAID technologies based on replication and erasure coding provide higher reliability guaran-

tees for massive data storage. This technology mainly uses striping technology (parallel I/O technology) and redundancy technology to enable parallel data access and recovery in storage systems, thereby achieving high performance, high reliability, and large capacity for disk storage systems. Initially, disk arrays mainly included RAID0 to RAID5. As large-scale storage systems demanded higher reliability, RAID-6 codes with fault tolerance of 2 were proposed, such as EVENODD codes, RS codes, and RDP codes.

RAID-6 uses dedicated dual parity disks (P+Q), namely row parity disks and diagonal parity disks. Elements in row parity disks are obtained through XOR operations on data elements in the same row, while elements in diagonal parity disks are obtained through XOR operations on all elements on the same diagonal. Figure 4[Figure 4: see original paper] shows an RDP coding system with p=5. Assuming Disk0 fails, the total amount of data elements read varies depending on the data recovery strategy adopted. Niu et al. [32] proposed a multi-stripe repair strategy for RAID-6 coding systems. They divided the repair process of a single stripe into three stages: data reading stage, data decoding stage, and data writing stage, as shown in Figure 5[Figure 5: see original paper] for the single-stripe repair process. Utilizing multi-threading technology proposed by Holland et al. [33] and considering node heterogeneity, they proposed the Parallel Heterogeneous Recovery (PHR) algorithm, which can return an optimal repair sequence in a timely manner. When using the PHR algorithm to repair failed data on a single stripe, the repair process is divided into three stages: reading corresponding data elements or parity elements from available disks according to the repair sequence returned by the PHR algorithm; decoding the read elements to obtain lost elements; and writing the decoded elements to other disks. These three stages proceed sequentially. Since repair processes for different stripes are independent, the PHR algorithm can be parallelized when repairing failed data on multiple stripes to further reduce regeneration time. Figure 6[Figure 6: see original paper] describes the multi-stripe repair process, where Ri, Di, Wi represent the read stage, decode stage, and write stage on stripe i, respectively. For the read stage on stripe i, the number of disks equals the number of threads, with each thread b independently performing read operations, while in the decode stage, the number of threads depends on CPU speed.

Compared to the enumeration method for searching optimal repair sequences, the PHR algorithm is more efficient. For example, when p=23, the enumeration method takes 17 seconds to search for the optimal algorithm, while the PHR algorithm only needs 3 seconds. When repairing multiple stripes, parallel execution of the PHR algorithm further reduces repair time.

2.3.3 Data Repair Technology Based on Download Cost Heterogeneity

Transmission cost refers to the cost of transmitting one element over a link, and bandwidth resources differ between nodes, resulting in different transmission costs. Based on this, Zhu et al. [34] linked transmission cost with bandwidth

resources, defining total repair cost as C_total = Σ_{i k} w_i × y_i, where node k fails, w_i represents the cost of reading one element from node i, and y_i represents reading y_i elements from node i. They proposed Cost-based Heterogeneous Recovery (CHR). The CHR algorithm enumerates all possible minimum-read recovery sequences, calculates the total repair cost of these sequences, and returns the recovery sequence corresponding to the minimum total repair cost. The CHR algorithm groups opposite or reverse minimum-read recovery sequences as one optimal recovery sequence, thereby reducing traversal space and computational overhead.

Figure 7[Figure 7: see original paper] shows an RDP coding system in a heterogeneous environment with p=7, where the bandwidth of each node is as shown. Assuming the cost of reading one element is 1 and node 0 fails, using traditional repair methods to repair node 0, the Proxy reads 6 elements from each of the first 6 nodes, with total download cost 6×6=36; using the hybrid repair method to repair node 0 yields one optimal repair sequence {1110000}, where the Proxy reads 4 elements from nodes 3, 4, and 7, 4 elements from nodes 2 and 5, and 5 elements from nodes 1 and 6, with total download cost 4×3 + 4×2 + 5×2 = 30; using the CHR algorithm to search yields one optimal recovery sequence {1010100}, with elements transmitted from each node as shown in Figure 8[Figure 8: see original paper], where the Proxy reads 3 elements from nodes 1, 6, and 7, 5 elements from nodes 2 and 5, and 4 elements from nodes 3 and 4, with total cost 3×3 + 5×2 + 4×2 = 27. This method reduces download cost by 40.91% compared to traditional repair methods and by 25.89% compared to hybrid repair methods.

3 Computation Capability Heterogeneity-oriented Repair Methods

In distributed storage systems, each storage node has different data processing speeds due to various factors, which we call node computing capability heterogeneity. During data repair, non-leaf nodes need to read local data, combine it with received data for encoding, and transmit the encoded results to their parent node. The processing speed of operations such as reading local data and encoding operations is limited by the node' s own processing speed. Therefore, computing capability heterogeneity is reflected in node encoding time; the stronger the node' s computing capability, the faster the data processing speed. Henry' s survey [35] shows that disk I/O has become a bottleneck for storage nodes reading and writing local data, so the impact of node computing capability heterogeneity on data repair cannot be ignored. However, few studies have addressed this issue. Only one literature on node computing capability heterogeneity is introduced below: the node selection scheme for distributed storage regenerating code data repair, and its performance is discussed in terms of repair bandwidth overhead, repair time overhead, and number of nodes participating in repair.

3.1 Node Selection Scheme for Distributed Storage Regenerating Code Data Repair

Li et al. considered the impact of heterogeneous bandwidth resources on the repair process, assuming nodes transmit data in parallel in a pipelined manner during repair, ignoring data processing time at nodes, and proposed a tree repair strategy to increase bottleneck bandwidth and reduce repair time. However, in practical distributed storage systems, node processing time has a significant impact on the repair process. Merely increasing bottleneck bandwidth without considering node processing time may not reduce repair time. Qi et al. [36] simultaneously considered the impact of node computing capability heterogeneity and bandwidth resource heterogeneity on the data repair process, establishing star and tree repair topologies. For the supplier node selection problem, they proposed the S-SPA-C algorithm and T-SPA-C algorithm to solve it.

For star topology repair structures, supplier nodes directly transmit data to the newcomer node, which encodes the received data and saves it. Therefore, the entire repair time is constrained by the node with the longest delay among all supplier nodes. Repair time can be expressed as T = max_{i B} {T_i^c + T_i^o}, where T_i^c represents the processing delay of node i, and T_i^o represents the transmission time from node i to the newcomer node (transmission delay). The approach to constructing a star repair structure is to determine supplier nodes based on the sum of node computing delay and transmission delay. The basic approach calculates the computing capability c of all n-1 nodes (excluding the failed node), combines it with transmission volume, calculates the transmission delay from these nodes to the newcomer node, and obtains the sum of processing time T_i^c and corresponding transmission delay T_i^o for n-1 nodes to transmit data to the newcomer node. T is sorted in ascending order, and nodes numbered 1,2,⋯,d are selected as supplier nodes. Taking Figure 9[Figure 9: see original paper] as an example, there are 4 supplier nodes P1,P2,P3,P4 and one newcomer node P0, where the data in circles represents node processing time, and data on edges represents transmission delay between nodes. Assuming the newcomer node connects to 3 nodes to repair failed data, the S-SPA-C algorithm determines P1,P3,P4 as supplier nodes, with the repair process shown in Figure 10[Figure 10: see original paper], where this round of repair time is 7.4t.

For tree topology repair structures, leaf nodes transmit data to their parent nodes, non-leaf nodes receive data from their child nodes, combine it with their own encoded blocks for encoding, and transmit the results upward to their parent nodes, 逐级上传至根节点（新生节点）, the root node receives all data, encodes and generates new encoded blocks, and saves them. The approach to constructing a tree repair structure is to use the minimum spanning tree principle to determine supplier nodes. First, the time required for each node to transmit data to the newcomer node through different links is calculated one by one, then sorted in ascending order to determine the path for each node to transmit data to the newcomer node, ultimately determining the repair structure. Still taking Fig-

ure 9[Figure 9: see original paper] as an example, assuming 3 nodes are selected as supplier nodes, the T-SPA-C algorithm selects P1,P3,P4 as supplier nodes, where P3 chooses the path P3→P2→P0 to transmit to P0. The repair process is shown in Figure 11[Figure 11: see original paper], where repair time is reduced from the original total time of 5.9t to 5.4t.

This supplier node selection scheme can accelerate the repair speed of failed data to a certain extent, reduce the time of the entire data repair process, and improve the performance of the entire storage network.

4 Storage Heterogeneity-oriented Repair Methods

Storage capacity heterogeneity refers to the fact that the amount of data stored on each storage node is not always equal. Current research on node storage capacity mainly focuses on examining how variations in node storage capacity affect the probability of users successfully decoding the original file, such as in references [37-48]. Leong et al. [44] first studied how to distribute files across storage nodes to maximize the probability of users successfully obtaining the file. Li et al. [39] designed a data distribution method for scenarios where each node in a distributed storage system is successfully accessed with different probabilities, aiming to improve the probability of users successfully obtaining the file. They proposed a hierarchical uniform distribution method, which achieves better performance compared to completely non-uniform data distribution methods [37]. Li et al. [49] considered node storage performance heterogeneity and proposed a data placement method for cloud file systems based on erasure codes, achieving system load balancing and improving data write and repair speeds based on real-time node load conditions. However, they did not further study the impact of storage heterogeneity on erasure code data repair performance. Currently, few studies have discussed the impact of node storage capacity variations on erasure code data repair. Only one storage capacity heterogeneity-based data repair technology is introduced below, and its performance in terms of repair bandwidth overhead is analyzed. Additionally, several papers on storage capacity variations are introduced.

Reference [50] describes a storage system where there exists a super node with larger storage capacity, reliability, and availability than other nodes. This literature proposes three storage allocation schemes for (k+2,k) MDS codes and (k+2,k) non-MDS codes, where the super node stores 1 blocks and other nodes store 2 blocks each, with different storage contents under each scheme. Under each storage scheme, the literature considers all possible node failure scenarios, describes the repair process for each failure scenario, and finally analyzes data reliability under the three storage schemes. From the perspectives of repair bandwidth and data reliability, repairing a single failed node achieves the minimum bandwidth of M/k, while repairing two nodes only requires 2M/k. Compared to traditional allocation methods (where all nodes have the same storage capacity), the data reliability is improved by 10%.

Reference [13] obtained the trade-off relationship between storage capacity and repair bandwidth by analyzing information flow graphs. They proved that if the minimum cut (min-cut) in the information flow graph is greater than the size of the original file, there exists a linear code that allows each DC node to recover the original file. If random linear coding is used in a sufficiently large finite field, DC nodes can recover the original file with probability approaching 1. Figure 12[Figure 12: see original paper] shows the trade-off curve between storage and bandwidth overhead for parameters n=10, k=5, d=9. The curve indicates that the larger the storage capacity per node, the smaller the bandwidth overhead for repairing a single failed node. Codes satisfying this trade-off relationship are called regenerating codes. The two extreme points on the curve correspond to two special types of codes: Minimum Storage Regenerating codes (MSR codes) and Minimum Bandwidth Regenerating codes (MBR codes), corresponding to different (,) values:

_MSR = M/k, _MSR = M×d/(k×(d-k+1))
_MBR = 2M×d/(k×(2d-k+1)), _MBR = 2M×d/(k×(2d-k+1))

Dimakis et al. derived the theoretical bound between node storage and repair bandwidth under the single-node repair model. For multi-node repair models, Shum et al. [51] proposed a model where newcomer nodes cooperate with each other and provided the theoretical bound for storage and bandwidth under this model. Zhang et al. [52] proposed a model where newcomer nodes no longer transmit data to each other, reducing design and computational complexity compared to cooperative repair and better meeting actual system needs. Wang et al. [53] used the cut principle to find the minimum cut for this new repair model and used linear programming to provide the theoretical bound for storage and bandwidth, with a simpler process. On this basis, they provided encoding construction methods for some special parameters. Li et al. [54] proposed a new scheme called Cache Size Adaptive Determination (CAROM) for data storage costs and bandwidth costs. This scheme combines traditional cache-based methods and erasure code methods to improve system repair efficiency. To achieve balance between cache size and its benefits, they proposed an elastic method based on the convex function characteristics of total cost to achieve elastic selection of cache size. The CAROM scheme reduces storage costs and bandwidth costs by 60% and 43% respectively compared to replication and erasure code strategies, offering both low bandwidth cost and low storage cost.

Reference [55] considered a relatively simple scenario, assuming the system has two types of nodes, S1 and S2, each corresponding to different storage costs C1 and C2, with storage capacities 1 and 2 respectively. Assuming there are n1 nodes with storage cost C1 and n2 nodes with storage cost C2, the total storage cost Cs can be expressed as: Cs = n1×C1× 1 + n2×C2× 2. During repair of failed data, the newcomer node connects to any d supplier nodes, downloading data volume from each node, and repair bandwidth can be expressed as: = d× . On this basis, they provide the trade-off relationship between storage cost and repair bandwidth. The limitation of this method is that the system only

has two types of nodes with storage costs, whereas in actual systems, storage costs may be diverse.

Figure 13[Figure 13: see original paper] shows an information flow graph based on MDS codes. The figure displays a distributed storage system with parameters n=4, k=2, d=3, where the storage cost of the first two nodes is 1 with storage capacity 1, and the storage cost of the last two nodes is 2 with storage capacity 2. Assuming V2_in→V2_out is a failed storage node and Vin→Vout is a newly added storage node (newcomer). To complete data repair, the newly added storage node needs to read data volume from each of the other d remaining storage nodes, i.e., the regeneration traffic marked in the figure.

Reference [56] generalized the optimization of storage cost to more general scenarios. They assumed that storage node v has storage capacity _v (v=1,2,⋯ ,n) and storage cost s_v (v=1,2,..,n). The average cost of storing one block (system cost) can be expressed as: C_s = Σ_{v=1}^n s_v × _v, subject to Σ_{v=1}^n _v B. They also considered the impact of communication cost on data repair, proposing a method to construct IFR codes and optimizing IFR codes to optimize data allocation. They proposed a non-uniform distribution method, which achieves better performance compared to completely uniform distribution methods [57]. The limitation of this method is that the constructed IFR coding system does not satisfy MDS properties. The MDS property in erasure codes is a desirable property that ensures users can recover the original file with maximum probability. If the system uses IFR codes to generate redundant data, DC nodes can only recover the original file through specified node sets and cannot satisfy the property of recovering from any k nodes.

Figure 14[Figure 14: see original paper] shows examples of constructing FR and IFR codes, both with parameters n=4, k=2, d=2. If MDS-FR codes are used, assuming the original file M=4, after (6,4) MDS encoding, 6 encoded blocks F1,F2,⋯,F6 are generated and allocated to 6 nodes. In this 6-node ring, the encoded blocks on edges between adjacent nodes represent the common blocks stored by these two nodes, and the numbers on edges represent the communication costs between adjacent nodes. As shown in Figure 14(a), each node has storage capacity 2. Node 1 stores F1 and F6, node 2 stores F1 and F2, node 3 stores F2 and F3, node 4 stores F3 and F4, node 5 stores F4 and F5, and node 6 stores F5 and F6, with total storage cost 34. Any node failure can be repaired by connecting to its two adjacent nodes, with total repair cost 36. If MDS-IFR codes are used, assuming the original file M=4, after (7,4) MDS encoding, 7 encoded blocks F1,F2,⋯,F7 are generated, with each node' s storage capacity as shown in Figure 14(b). The total storage cost is 33, and total repair cost is 22. It can be seen that the latter allocation method achieves better performance, but the limitation is that it requires higher storage costs than triple replication to provide very low repair costs.

5 Discussion

To comprehensively compare existing erasure code repair performance, Table 1 compares the data repair performance of 6 typical erasure codes using space utilization rate, single-block repair cost, and total repair cost as evaluation criteria. For comparison, the table includes common triple replication technology.

Table 1. Comparison of Data Repair Performance Among Several Typical Erasure Codes and Multi-replication Technology

Erasure Code Type	Space Utilization Rate	Single-block Repair Cost	Total Repair Cost
RS(14,10) Traditional MDS code	71.4%	10M	10M
LRCs(10,2,4)	62.5%	5M	6M
SHEC(10,6,5)	58.8%	6M	6M
(9,5,8)-MSR	55.6%	5M	5M
(14,10,13)-MBR	50.0%	5M	5M
(14,10)-Hitchhiker	71.4%	7M	7M
Triple Replication	33.3%	1M	1M

As shown in Table 1, no single coding scheme can well satisfy all three metrics. Traditional MDS code RS(14,10) has the highest space utilization rate, but its single-block repair cost and total repair cost are also the largest, even several times higher than other types of erasure codes. Compared to traditional MDS codes, group codes LRCs(10,2,4) and SHEC(10,6,5) can significantly reduce single-block repair cost and total repair cost at the expense of relatively small additional storage space overhead. Regenerating code (14,10,13)-MBR achieves good performance in both single-block repair cost and total repair cost, but regenerating codes have significantly lower storage space utilization than other categories of erasure codes, with a maximum storage space utilization of only about 50%. Therefore, regenerating codes are more suitable for systems with high network bandwidth costs and storage costs.

Most erasure code data repair schemes optimize repair bandwidth, repair time, disk access, number of nodes participating in repair, and repair cost based on fixed parameters (n,k) and (,). These research works cover functional repair and exact repair of erasure codes but do not consider the impact of system bandwidth resources, computing resources, and storage resources on data repair. However, the reality is that in large-scale data centers, equipment replacement and hardware failures not only cause data loss but also lead to hardware differences among storage nodes in data centers, such as differences in available

bandwidth, computing capability, and storage capacity between storage nodes. Therefore, studying and optimizing the redundant data repair performance of distributed storage coding in heterogeneous environments has important theoretical significance and practical value.

Currently, most research on storage coding in distributed storage systems considers homogeneous environments (treating each storage node indiscriminately), assuming consistent bandwidth resources, computing resources, and storage resources among nodes in the distributed system. However, the actual situation is that geographical differences and disk performance variations lead to hardware differences among nodes. Even the few research works on erasure code data repair in heterogeneous environments focus on the impact of bandwidth resource heterogeneity on erasure code data repair, rarely considering the impact of node computing capability and storage capacity heterogeneity on erasure code data repair. Additionally, current erasure code data repair technology has significant defects in bandwidth overhead and time overhead, making it difficult to achieve ideal states for all these objectives simultaneously. Therefore, heterogeneous distributed systems become very meaningful in practical deployment, but research on data repair technology for heterogeneous distributed systems remains theoretical, with practical applications still being a blank slate. In the future, we will apply some repair methods from optimization theory and some properties from graph theory to actual distributed systems and study repair performance gain issues after coding in some special scenarios. How to design erasure code data repair technology that is excellent in all aspects remains a long-term challenge for future research.

6 Conclusion

This paper analyzes the factors affecting erasure code data repair and discusses methods for optimizing erasure code repair performance from three aspects: bandwidth resources, computing resources, and storage capacity resources. Most existing repair methods do not consider the heterogeneity of bandwidth resources, computing resources, and storage capacity resources among storage nodes. The challenges facing erasure code data repair technology are mainly manifested in three aspects: computation, read/write, and transmission. Among them, SIMD technology can perform the same operation on multiple data units simultaneously, accelerating encoding computation speed based on finite field operations. Adjusting computation order and avoiding repeated calculations can effectively reduce computation 量 to address computational challenges. By reasonably selecting stripes for repair, the data required for repair can have more overlaps, allowing read data blocks to be used for repairing multiple data blocks. Additionally, without reducing the total amount of data read, introducing more disks into the data repair process can reduce the read volume on individual disks to address read/write challenges. Through reasonable coding design, regenerating codes and various other derivative codes can improve data repair performance in terms of bandwidth overhead and

supplier node overhead. Some codes [58-74] support exact repair, making it possible for systematic codes (codes where encoded blocks contain original data) to be applied in distributed systems, providing a technical foundation for improving data access performance to address coding challenges.

References

[1] Lakshmi N. Bairavasundaram, Garth R, et al. An analysis of latent sector errors in disk drives [J]. ACM SIGMETRICS Performance Evaluation Review, 2007, 35 (1): 289-300.

[2] Honnutagi P S. The Hadoop distributed file system [J]. International Journal of Computer Science & Information Technolo, 2014, 5 (5): 6238-6242.

[3] Weil S A, Brandt S A, Miller E L, et al. Ceph: a scalable, high-performance distributed file system [C]// Proc of the 7th Conference on USENIX Symposium on Operating Systems Design and Implementation. Berkeley: USENIX Association, 2006: 307-320.

[4] Reed I S, Solomon G. Polynomial codes over certain finite fields [J]. Journal of the Society for Industrial & Applied Mathematics, 1960, 8 (2): 300-304.

[5] Rashmi K V, Shah N B, Gu Dikang, et al. A solution to the network challenges of data recovery in erasure-coded distributed storage systems: a study on the Facebook warehouse cluster [C]// Proc of the 5th Workshop on Hot Topics in Storage and File Systems. Berkeley: USENIX Association, 2013: 1-5.

[6] Dimakis A G, Ramchandran K, Wu Yunnan, et al. A survey on network codes for distributed storage [J]. Proceedings of the IEEE, 2011, 99 (3): 476-489.

[7] Li Jun, Li Baochun. Erasure coding for cloud storage systems: a survey [J]. Tsinghua Science and Technology, 2013, 18 (3): 259-272.

[8] 罗象宏, 舒继武. 存储系统中的纠删码研究综述 [J]. 计算机研究与发展, 2012, 49 (1): 1-11. (Luo Xianghong, Shu Jiwu. Summary of research for erasure code in storage system [J]. Journal of Computer Research and Development, 2012, 49 (1): 1-11.)

[9] 王意洁, 许方亮, 裴晓强. 分布式存储中的纠删码容错技术研究 [J]. 计算机学报, 2017, 40 (1): 236-255. (Wang Yijie, Xu Fangliang, Pei Xiaoqiang. Research on erasure code-based fault-tolerant technology for distributed storage [J]. Chinese Journal of Computers, 2017, 40 (1): 236-255.)

[10] 杨松霖, 张广艳. 纠删码存储系统中数据修复方法综述 [J]. 计算机科学与探索, 2017, 11 (10): 1531-1544. (Yang Songlin, Zhang Guangyan. Review of data recovery in storage systems based on erasure codes [J]. Journal of Frontiers of Computer Science and Technology, 2017, 11 (10): 1531-1544.)

[11] Ernvall T, El Rouayheb S, Hollanti C, et al. Capacity and security of heterogeneous distributed storage systems [J]. IEEE Journal on Selected Areas in Communications, 2013, 31 (12): 2701-2709.

[12] Pei Xiaoqiang, Wang Yijie, Ma Xingkong, et al. Cooperative repair based on tree structure for multiple failures in distributed storage systems with regenerating codes [C]// Proc of the 12th ACM International Conference on Computing Frontiers. New York: ACM Press, 2015: 784-792.

[13] Dimakis A G, Godfrey P B, Wu Yunnan, et al. Network coding for distributed storage systems [J]. IEEE Trans on Information Theory, 2010, 56 (9): 4539-4551.

[14] Wu Yunnan, Dimakis A, Ramchandran K. Deterministic regenerating codes for distributed storage [C]// Proc of Allerton Conference on Control, Computing, and Communication. 2007.

[15] Shah N B, Rashmi K V, Kumar P V. A flexible class of regenerating codes for distributed storage [C]// Proc of IEEE International Symposium on Information Theory. 2010: 1943-1947.

[16] Gong Qingyuan, Wang Jiaqi, Wei Dongsheng, et al. Optimal node selection for data regeneration in heterogeneous distributed storage systems [C]// Proc of International Conference on Parallel Processing. 2015: 390-399.

[17] Akhlaghi S, Kiani A, Ghanavati M R. A fundamental trade-off between the download cost and repair bandwidth in distributed storage systems [C]// Proc of IEEE International Symposium on Network Coding. 2010: 1-6.

[18] 贾亚男, 岳殿武. 博弈论框架下认知小蜂窝网络的动态资源配算法 [J]. 电子学报, 2015, 43 (10): 1911-1917. (Jia Yanan, Yue Dianwu. Dynamic resource allocation algorithm based on game theory in cognitive small cell networks [J]. Acta Electronica Sinica, 2015, 43 (10): 1911-1917.)

[19] 洪浩, 张焱, 肖立民, 等. 认知双向中继网络的功率分配优化算法研究 [J]. 电波科学学报, 2014, 29 (2): 201-206+226. (Hong Hao, Zhang Yan, Xiao Limin, et al. Optimal power allocation for cognitive two-way relaying networks with underlay spectrum sharing [J]. Chinese Journal of Radio Science, 2014, 29 (2): 201-206+226.)

[20] 郑力明, 李晓冬. 面向纠删码的低成本多节点失效修复方法 [J]. 计算机工程, 2017, 43 (7): 110-118, 123. (Zheng Liming, Li Xiaodong. Low-cost multi-node failure repair method for erasure codes [J]. Computer Engineering, 2017, 43 (7): 110-118, 123.)

[21] Li Jun, Yang Shuang, Wang Xin, et al. Tree-structured data regeneration with network coding in distributed storage systems [C]// Proc of International Workshop on Quality of Service. 2009: 2892-2900.

[22] Li Jun, Yang Shuang, Wang Xin, et al. Tree-structured data regeneration in distributed storage systems with regenerating codes [C]// Proc of INFOCOM. 2010: 1-9.

[23] Wang Yan, Wei Dongsheng, Yin Xunrui, et al. Heterogeneity-aware data regeneration in distributed storage systems [C]// Proc of INFOCOM. 2014.

[24] Lee S J, Sharma P, Banerjee S, et al. Measuring bandwidth between planetlab nodes [C]// Proc of International Conference on Passive and Active Network Measurement. Springer-Verlag, 2005: 292-305.

[25] Li Runhui, Li Xiaolu, Lee P P C, et al. Repair pipelining for erasure-coded storage [C]// Proc of Usenix Technical Conference. Berkeley: USENIX Association, 2017.

[26] Akhlaghi S, Kiani A, Ghanavati M R. Cost-bandwidth tradeoff in distributed storage systems [J]. Computer Communications, 2010, 33 (17): 2105-2115.

[27] Gerami M, Xiao Ming, Skoglund M. Optimal-cost repair in multihop distributed storage systems [C]// Proc of IEEE International Symposium on Information Theory. 2012: 1437-1441.

[28] Xiang Liping, Xu Yinlong, Lui J C S, et al. Optimal recovery of single disk failure in RDP code storage systems [J]. ACM SIGMETRICS Performance Evaluation Review, 2010, 38 (1): 119-130.

[29] Corbett P, English B, Goel A, et al. Row-diagonal parity for double disk failure correction [C]// Proc of Usenix Conference on File and Storage Technologies. Berkeley: USENIX Association, 2004: 1-1.

[30] Khan O, Burns R, Plank J, et al. Rethinking erasure codes for cloud file systems: minimizing I/O for recovery and degraded reads [C]// Proc of Usenix Conference on File and Storage Technologies. Berkeley: USENIX Association, 2012: 20.

[31] Zhu Yunfeng, Lin Jian, Lee P P C, et al. Boosting degraded reads in heterogeneous erasure-coded storage systems [J]. IEEE Trans on Computers, 2015, 64 (8): 2145-2157.

[32] Niu Fang, Xu Yinlong, Zhu Yunfeng, et al. PHR: a pipelined heterogeneous recovery for raid6-coded storage systems [C]// Proc of International Conference on Parallel and Distributed Computing, Applications and Technologies. 2014: 325-331.

[33] Holland M, Gibson G A, Siewiorek D P. Architectures and algorithms for on-line failure recovery in redundant disk arrays [J]. Distributed & Parallel Databases, 1994, 2 (3): 295-335.

[34] Zhu Yunfeng, Lee P P C, Xiang Liping, et al. A cost-based heterogeneous recovery scheme for distributed storage systems with RAID-6 codes [C]// Proc of IEEE/IFIP International Conference on Dependable Systems and Networks. 2012: 1-12.

[35] Henry. I/O bottlenecks: biggest threat to data storage. [2009-12-31]. http://www.enterprisestorageforum.com/technology/features/article.php/3856121/IO-Bottlenecks-Biggest-Threat-to-Data-Storage.htm.

[36] 齐凤林, 宫庆媛, 周扬帆, 等. 分布式存储再生码数据修复的节点选择方案 [J]. 计算机研究与发展, 2015, 52 (Suppl): 68-74. (Qi Fenglin, Gong Qingyuan, Zhou Yangfan, et al. Heterogeneity-aware node selection of data repair in distributed storage systems [J]. Journal of Computer Research and Development, 2015, 52 (Suppl.): 68-74.)

[37] Li Zhao, Ho T, Leong D, et al. Distributed storage allocation for heterogeneous systems [C]// Communication, Control, and Computing. 2013: 320-326.

[38] Huang Zhen, Yuan Yuan, Peng Yuxing. Storage allocation for redundancy scheme in reliability-aware cloud systems [C]// Proc of IEEE International Conference on Communication Software and Networks. 2011: 275-279.

[39] Ntranos V, Caire G, Dimakis A G. Allocations for heterogeneous distributed storage [C]// Proc of IEEE International Symposium on Information Theory Proceedings. 2012: 2761-2765.

[40] Kao Y H, Dimakis A G, Leong D, et al. Distributed storage allocations and a hypergraph conjecture of Erdős [C]// Proc of IEEE International Symposium on Information Theory Proceedings. 2013: 902-906.

[41] Xu Guangping, Lin Sheng, Wang Gang, et al. HERO: heterogeneity-aware erasure coded redundancy optimal allocation for reliable storage in distributed networks [C]// Proc of IEEE International Performance Computing and Communications Conference. 2012: 246-255.

[42] Leong D, Dimakis A G, Ho T. Distributed storage allocation problems [C]// Proc of Workshop on Network Coding, Theory, and Applications. 2009: 199-205.

[43] Derek Leong, Alexandros G. Dimakis, Tracey Ho. Distributed Storage Allocation for High Reliability [C]// IEEE, International Conference on Communications. IEEE, 2010: 1-6.

[44] Leong D, Dimakis A G, Ho T. Distributed storage allocations [J]. IEEE Trans on Information Theory, 2012, 58 (7): 4733-4752.

[45] Leong D, Dimakis A G, Ho T. Symmetric allocations for distributed storage [C]// Proc of Global Telecommunications Conference. 2010: 1-6.

[46] Sardari M, Restrepo R, Fekri F, et al. Memory allocation in distributed storage networks [C]// Proc of IEEE International Symposium on Information Theory. 2010: 1958-1962.

[47] Hong Tao, Wu Yating, Cao Bingyao, et al. A dynamic data allocation method with improved load-balancing for cloud storage system [C]// Proc of IET International Conference on Smart and Sustainable City. 2013: 183-187.

[48] 李君, 侯孟书. 基于萤火虫优化的副本放置方法 [J/OL]. 计算机应用研究, 2019, 36 (3). [2018-02-02]. http://www.arocmag.com/article/02-2019-03-045.html. (Li Jun, Hou Mengshu. Replica placement strategy based on glowworm swarm

optimization [J/OL]. Application Research of Computers, 2019, 36 (3). [2018-02-02]. http://www.arocmag.com/article/02-2019-03-045.html.)

[49] 李佳, 陈海涛, 芦伟. 基于纠删码的云文件系统数据放置方法 [J]. 北京信息科技大学学报: 自然科学版, 2014, 29 (6): 1-6. (Li Jia, Chen Haitao, Lu Wei. A Novel way of data placement of cloud file system based on erasure code [J]. Journal of Beijing Information Science and Technology University: Natural Science, 2014, 29 (6): 1-6.)

[50] Van Vo T, Chau Yuen, Li Jing. Non-homogeneous distributed storage systems [C]// Communication, Control, and Computing. 2012: 1133-1140.

[51] Shum K W, Hu Yuchong. Cooperative regenerating codes [J]. IEEE Trans on Information Theory, 2013, 59 (11): 7229-7258.

[52] Zhang Huayu, Li Hui, Hou Hanxu, et al. Concurrent regenerating codes and scalable application in network storage [J]. arXiv preprint arXiv: 1604.06567, 2016.

[53] 王丽莎, 唐小虎. 新多节点修复模型下的再生码 [J]. 计算机应用研究, 2018, 35 (2): 527-531. (Wang Lisha, Tang Xiaohu. regenerating codes for new multi-node repair model [J]. Application Research of Computers, 2018, 35 (2): 527-531.)

[54] 李松涛, 金欣. 基于混合策略的低成本云存储方案 [J]. 计算机应用, 2014, 34 (10): 2800-2805+2811. (Li Songtao, Jin Xin. Low-cost cloud storage scheme based on hybrid strategy [J]. Journal of Computer Applications, 2014, 34 (10): 2800-2805+2811.)

[55] Yu Quan, Shum K W, Sung C W. Minimization of storage cost in distributed storage systems with repair consideration [C]// Proc of Global Telecommunications Conference. 2012: 1-5.

[56] Yu Quan, Sung C W, Chan T H. Irregular fractional repetition code optimization for heterogeneous cloud storage [J]. IEEE Journal on Selected Areas in Communications, 2014, 32 (5): 1048-1060.

[57] El Rouayheb S, Ramchandran K. Fractional repetition codes for repair in distributed storage systems [C]// Communication, Control, and Computing. 2010: 1510-1517.

[58] Papailiopoulos D S, Luo Jianqiang, Dimakis A G, et al. Simple regenerating codes: Network coding for cloud storage [C]// Proc of INFOCOM. 2015: 2801-2805.

[59] 徐志强, 袁德砦, 陈亮. 基于稀疏随机矩阵的再生码构造方法 [J]. 计算机应用, 2017, 37 (7): 1948-1952+1959. (Xu Zhiqiang, Yuan Dezhai, Chen Liang. Regenerating codes construction method based on sparse random matrix [J]. Journal of Computer Applications, 2017, 37 (7): 1948-1952+1959.)

[60] 宋海龙, 王伟平, 肖亚龙. 基于柯西矩阵的最小带宽再生码研究 [J]. 湖南大学学报: 自然科学版, 2017, 44 (8): 152-160. (Song Hailong, Wang Weiping, Xiao Yalong. Study

of minimum bandwidth regeneration codes based on cauchy matrix [J]. Journal of Hunan University: Natural Science, 2017, 44 (08): 152-160.)

[61] 曹凯, 文捷. 基于 (k+2, k) MSR 的多容错低修复带宽编码 [J]. 计算机工程, 2018, 44 (2): 84-87, 91. (Cao Kai, Wen Jie. Multiple Fault tolerant and low repairing bandwidth coding based on (k+2, 2) MSR [J]. Computer Engineering, 2018, 44 (2): 84-87, 91.)

[62] Qu Shan, Zhang Jinbei, Wang Xinbing. Asymmetric regenerating codes for heterogeneous distributed storage systems [C]// Proc of IEEE International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. 2018: 1-8.

[63] 万武南, 杨威, 陈运. 一种新的 3 容错扩展 RAID 码 [J]. 北京邮电大学学报, 2014, 37 (5): 75-79. (Wan Wunan, Yang Wei, Chen Yun. A toleration based extended raid code triple failures [J]. Journal of Beijing University of Posts and Telecommunications, 2014, 37 (05): 75-79.)

[64] 李琛, 李琦, 高军萍, 等. 基于 Hadamard 向量的新型 (k+2, k) MSR 码 [J]. 河北工业大学学报, 2018, 47 (02): 9-13. (Li Chen, Li Qi, Gao Junping, et al. New (k+2, k) MSR codes based on Hadamard vectors [J]. Journal of Hebel University of Technology, 2018, 47 (02): 9-13.)

[65] 马良荔, 柳青. 减少重建数据量的冗余编码技术研究 [J]. 计算机科学, 2017, 44 (S1): 463-469. (Ma Liangli, Liu Qing. Researches of redundancy coding technologies on reducing reconstruction data amount [J]. Computer Science, 2017, 44 (S1): 463-469.)

[66] 李杰. 面向分布式存储系统的具有最优存取//更新性质的最小存储再生码的设计与分析 [D]. 成都: 西南交通大学, 2017. (Li Jie. Design and analysis of minimum storage regenerating codes with the optimal access//update property for distributed storage systems [D]. Chengdu: Southwest Jiaotong University, 2017.)

[67] 李晨卉. 应用于分布式存储系统的准循环再生码构造方案 [J]. 计算机工程, 2015, 41 (3): 81-87. (Li Chenhui. Construction Scheme of Quasi-cyclic Regenerating Code for Distributed Storage System [J]. Computer Engineering, 2015, 41 (3): 81-87.)

[68] Chen Bin, Xia Shutao, Hao Jie, et al. Constructions of optimal cyclic (r,) locally repairable codes [J]. IEEE Trans on Information Theory, 2016, PP (99): 1-1.

[69] Park H, Lee D, Moon J. LDPC code design for distributed storage: balancing repair bandwidth, reliability and storage overhead [J]. IEEE Trans on Communications, 2018, PP (99): 1-1.

[70] 肖宜龙. 随机化数据冗余方法及其在存储系统中的应用 [D]. 成都: 电子科技大学, 2013. (Xiao Yilong. random data redundancy method and its application in distributed storage systems [D]. Chengdu: University of Electronic Science and Technology of China, 2013.)

[71] 王禹, 赵跃龙, 侯昉. 基于矩阵运算的最小冗余存储再生码 MSRRC 研究 [J]. 计算机科学, 2014, 41 (S2): 191-194+207. (Wang Yu, Zhong Yuelong, Hou Fang. Minimun Redundancy storage regeneration code research msrrc based on matrix operation [J]. Computer Science, 2014, 41 (S2): 191-194+207.)

[72] 谢显中, 黄倩, 王柳苏, 等. 一种云存储中基于干扰对齐的多节点精确修复方法 [J]. 电子学报, 2014, 42 (10): 1873-1881. (Xie Xianzhong, Huang Qian, Wang Liusu, et al. A multi-node exact repair method in cloud storage based on interference alignment [J]. Acta Electronica Sinica, 2014, 42 (10): 1873-1881.)

[73] 李小兵, 许胤龙, 林一施, 等. X 再生码: 一类适用于云存储的准确修复编码 [J]. 计算机应用与软件, 2014, 31 (08): 241-244, 248. (Li Xiaobing, Xu Yinlong, Lin Yishi, et al. X regenerating codes: a calss of accurate repair codes for cloud storage [J]. Computer Applications and Software, 2014, 31 (08): 241-244+248.)

[74] 王静, 张崇, 梁伟, 等. 分布式存储系统中基于 Pyramid 码的局部性修复编码 [J]. 电子测量与仪器学报, 2017, 31 (9): 1481-1487. (Wang Jing, Zhang Chong, Liang Wei, et al. Locally repairable codes based on Pyramid codes in distributed storage systems [J]. Journal of Electronic Measurement and Instrumentation, 2017, 31 (9): 1481-1487.)

Note: Figure translations are in progress. See original paper for figures.