Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2021.020408
ISSN 2790-0916 Vol. 2, Issue 4: 61-68

Distributed System Optimization Relying on Simulated Annealing Algorithm and Data Storage Technology

Sahiilin Kavitane*

Saadah University, Yemen

**corresponding author*

Keywords: Simulated Annealing Algorithm, Distributed System, Data Storage Technology, Optimization Scheme

Abstract: The storage demand for large-scale data in modern society is increasing, and the traditional storage architecture has limitations in disaster tolerance, scalability, and maintenance management, resulting in additional costs for enterprises. In order to solve this problem, the distributed system(DS) and data storage technology can be fully combined to give full play to the advantages of the DS in processing massive data. However, DSs are faced with serious data storage overhead and high availability problems. For these two problems, this paper designs a corresponding optimization scheme and verifies the effectiveness of the optimization scheme on the performance of DSs. The results show that the local repair code based on the system MSR code is conducive to the rapid data repair, and is consistent with the data recovery of the simulated annealing algorithm, saving the time overhead of the data reading process and data repair process; and the optimized high availability scheme can Reduce system failure recovery time.

1. Introduction

In data-centric environments such as the Internet of Things, sensor networks, and data center networks, the security and efficiency of distributed data storage is a current research focus. The traditional data backup method is not suitable for a highly reliable distributed network storage system due to its high overhead and difficulty in data synchronization. For data storage problems, DSs based on data storage technology can achieve high reliability and low overhead, and have attracted wide attention in data processing [1-2].

The applied research on simulated annealing algorithm and data storage technology in DSs has achieved good results. For example, the uniform coding data block allocation model proposed by a certain scholar lacks consideration of the availability of the storage nodes themselves, or assumes that the node failure rates are the same, which makes the distributed storage system due to the online duration of the storage nodes themselves, external hacker attacks and other reasons. Data storage nodes cannot be accessed at any time [3]. Some studies expand NameNode into multiple DSs, use simulated annealing algorithm to achieve system consistency, and design a corresponding election mechanism, but these schemes are relatively complex, difficult to implement, and not universal [4]. A related study proposes a distributed cache scheme, which reduces the amount of metadata that needs to be managed by merging small files, and uses client cache and DataNode cache to reduce unnecessary access requests. The client cache stores metadata information of files accessed by users, and the DataNode cache stores small files frequently accessed by users. However, this scheme needs to design a relevant replacement strategy when managing the cache, which is difficult to implement [5]. Although DSs are widely used, the data storage problem they face has also received unanimous attention from scholars, so the overall performance of the system has to be improved.

This paper first briefly describes the two major problems existing in DSs, and proposes an optimization scheme for these two problems. In order to verify the effectiveness of the optimization scheme, this paper conducts a verification experiment analysis. Under the guidance of simulated annealing algorithm for data recovery, this paper uses coding technology to repair data, and verifies the effectiveness of the HA optimization scheme by comparing the recovery switching time of the DS under single-node and double-node failures.

2. DS Optimization and Simulated Annealing Algorithm

2.1. Analysis of Problems Existing in DSs

(1) Data storage problem

The traditional data storage method is not ideal due to its scalability, reliability and access delay. For a large amount of data, the distributed storage method must be deployed [6]. Although distributed storage systems have been widely used, many DSs currently use the replica mechanism as a fault tolerance mechanism to improve the fault tolerance of the system. The storage overhead of this method is very serious, but it has not achieved a very good repair effect [7]. To improve the overall performance of the system, erasure codes have been introduced. Compared with the copy mechanism, the erasure code improves the storage performance of the system, but each repair process needs to restore the original data and then obtain the data of the failed node [8].

(2) High availability problem (HA)

Although HDFS sets up various mechanisms to maintain the reliability of the system, due to its innate architecture, the cluster has only one central management node NameNode [9] at any time. Unlike DataNode, which has a replication strategy to ensure the reliability of data blocks, the metadata of the NameNode does not implement hot backup. Once the NameNode fails for some reason, the EditLog and FSImage on its local disk cannot be applied, which is equivalent to the entire namespace is unavailable. NameNode Unable to use metadata information to respond to requests from clients, which is a high availability problem of HDFS. In response to this problem, this paper proposes a multi-NameNode scheme based on the HDFS HA architecture, so that the system can still serve externally after the primary and secondary nodes of the NameNode fail to achieve high availability of HDFS [10].

In general, the definition of HA is expressed by the following formula:

$$MTTF/(MTTF+MTTR)/100\% \tag{1}$$

MTTF means mean time between failures and MTTR means mean time to repair.

2.2. Simulated Annealing Algorithm

Simulated Annealing is an efficient approximation algorithm for large-scale combinatorial optimization problems [11]. Applying the simulated annealing algorithm to the data recovery of distributed files, the data receiving node of the DS can successfully recover the original data, and the amount of data obtained by accessing the data storage node is not less than the original data [12]. The original data is the unit size, that is, the amount of data accessed by the data receiving node is not less than 1. Assuming that the data receiving node successfully restores the original data, it needs to access d data storage nodes to form a d subset, and use |d| to represent the number of elements in the d subset, that is, the number of storage nodes. Define K as the event of successful data recovery.

$$P(K)=\sum_{|d|=1}^{m}P(|d|)\cdot I\left[\sum_{i\in d}x_i\geq 1\right] \tag{2}$$

In the network model of DS based on data storage technology, the basic flow of data information is from the source data node to the data storage node, and the data receiving node obtains data information from the data storage node [13]. The amount of data stored in the data storage nodes in the network is different. Due to the dynamic characteristics of the network, node exit, failure and other reasons, the network topology is constantly changing, and the link between the data receiving node and the data storage node is also constantly changing. In the change of data receiving node, the success rate of data information acquisition is also constantly changing [14]. The allocation strategy of using encoded data blocks is the core key issue of distributed data storage based on network coding. The simulated annealing algorithm can be used to allocate the node availability of the DS, and the idea of combinatorial optimization can be introduced to convert the unstorable data state into a storable state [15-16].

3. DS Optimization Scheme Design

3.1. Design of Coding Repair Scheme

Erasure coding divides the original file into chunks, and then encodes the chunked data and distributes it in different data nodes. When some node data fails, some of the data can be completely restored to the original file, and then the data of the failed node can be obtained [17]. Erasure codes can maintain high reliability of the system while generating less data redundancy. Due to its strong fault tolerance, high space utilization, and low redundancy, erasure coding technology is increasingly used in DSs [18]. Commonly used erasure codes include RS coding, LDPC coding, and the like. Figure 1 shows the data encoding and decoding process.

For the storage overhead of various redundant storage methods, the most commonly used three-copy mechanism, RS mechanism and local repair coding mechanism of system MSR code are mainly discussed, and the three methods are respectively applied to the HDFS system to measure their storage overhead.

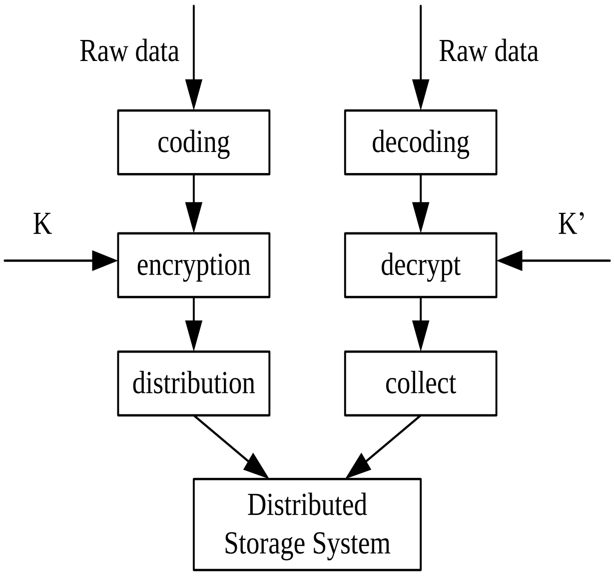

Figure 1. Encrypted encoded data flow

3.2. High Availability (HA) Optimization Scheme Design

In order to meet the high-performance reading and writing requirements in HDFS, all operations performed on metadata are recorded in EditLog. Therefore, the metadata that needs to be synchronized by the shared storage mechanism is EditLog file, which is not responsible for the storage of FSImage file. FSImage is still stored in On NameNode disk, it is periodically loaded into memory and merged with the EditLog that completes the synchronization. The shared storage system consists of the active and standby NameNode nodes and the JournalNode cluster. The active node writes the EditLog to the cluster, and the standby node periodically reads files from the cluster to complete synchronization. In the original HA scheme, the namespace metadata of the system is jointly managed by the active and standby nodes. Drawing on the idea of the three-copy strategy of data blocks, the standby node is artificially expanded from one to two to ensure the availability of metadata to the greatest extent. Once the master node fails, Zookeeper will use its internal election mechanism to elect a new node and take over the cluster in a warm-start fashion.

4. DS Optimization Scheme Verification

4.1. Code Repair Scheme Verification

The three-copy mechanism, RS code, MSR code and local repair code of system MSR code are respectively applied to HDFS system to verify the time overhead of system reading data and node repair.

(1) Data reading process

Figure 2. Time overhead during data reading

As can be seen from Figure 2, compared with the original HDFS, the time overhead of reading data in the HDFS system using the encoding technology is relatively large. In the reading process, the RS erasure code needs to decode the coding block, so it takes a long time, but the copy mechanism and the local repair coding based on the system MSR code can directly read the original data, so it takes less time and close.

(2) Node repair process

Table 1. Time overhead of small node repair process for different files (time/10000ms)

	100	200	300	400	500	600
RS offset correction code	2.32	4.75	7.34	8.25	10.08	13.42
Local Repair Coding for MSR Codes	1.64	2.58	3.61	4.33	5.83	6.65
MSR code	1.57	2.39	3.52	4.21	5.34	6.18

As shown in Table 1, the repair time of the local repair code of the system MSR code is similar to that of the MSR code, that is to say, the difference in bandwidth consumption between the two is not large. The computational overhead of the local repair coding of the systematic MSR code is relatively larger than that of the RS code, but the use of the local repair code based on the systematic MSR code is much smaller than the repair cost of the RS erasure code in the process of repairing the invalid data, and with As the number of invalid data blocks increases, the time gap also increases. Since the HDFS system mostly uses low-cost machines, the possibility of failure is relatively high. Therefore, it is more important to quickly repair the data under the condition of occupying less bandwidth. At this time, it can be repaired by simulated annealing algorithm.

4.2. Verification and Analysis of High Availability Optimization Scheme

In the test, the switchover time is defined as the time difference from the occurrence of the fault to the time when the switchover succeeds and the system returns to normal. In addition, since the optimization scheme in this paper is proposed to improve the dual-node failure of the HA scheme, in addition to testing the switching time under single-node failure, the switching time under dual-node failure is also tested.

In the single-node test, the optimized HA scheme was compared with the original HDFS HA scheme and the Avatar scheme, and a single point of failure was simulated by simulating the downtime of the master node. Three experiments were performed for each scheme, and the recording time was averaged.. When switching between the HA solution and the optimized HA solution, you need to download the latest EditLog from the Journal cluster, load it into memory and merge it with FSImage to serve externally. The scale of metadata that needs to be restored has a certain impact on the switching time. Therefore, each solution has a certain impact on the switching time. Different numbers of files were selected for testing, and the results are shown in Table 2.

Table 2. Comparison of switching time under single node failure

	10000	20000	30000	40000	50000	60000	70000	80000
Avatar solution	74	123	159	178	196	208	220	232
HDFS HA solution	36	42	45	49	52	55	60	64
Improved HA scheme	34	38	42	47	50	52	56	59

It can be seen from the test results that the switching time between the HA solution and the optimized HA solution is shorter than that of the Avatar solution. This is because the HA solution and the optimized HA solution use the same hot backup mechanism, so the switching time is not much different.

After the original HA solution is switched, there is only one node that provides services in the cluster, and it still faces a new single point of failure problem. In order to test the performance of the optimized solution under the failure of two nodes, after both NameNode nodes of the original HA solution are down, manually start the new node, and record the switching time after returning to normal. Compared with the optimized HA solution, the results are shown in Figure 3.

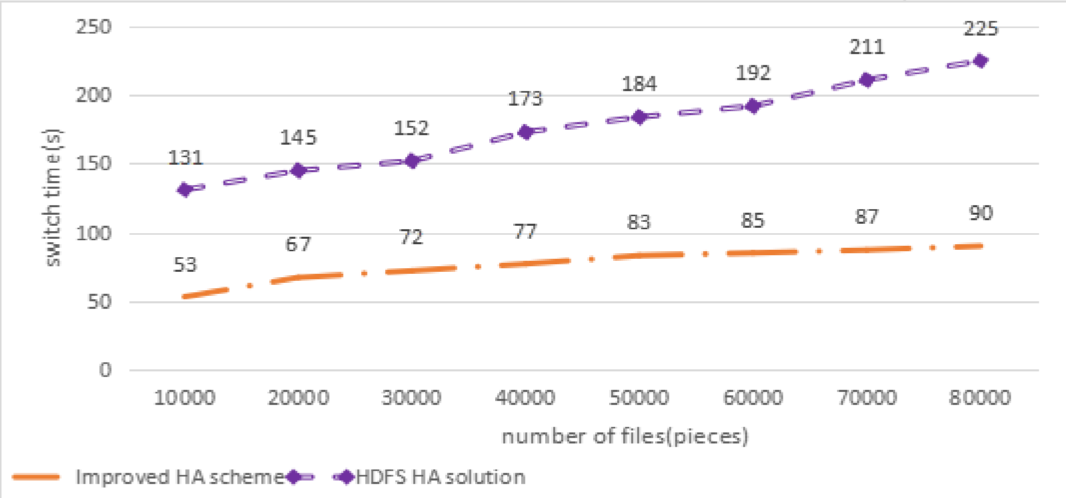

Figure 3. Comparison of switching time under dual node failure

It can be seen from the experimental results in Figure 3 that in the original HA scheme, when both the primary and secondary nodes fail, the administrator can only restore one of the nodes manually, and all metadata of the system needs to be reloaded when starting the node. to memory, it takes a certain amount of time. The solution in this article will automatically start the remaining hot standby nodes. Just go to the JournalNode cluster to download the metadata of the latest cycle, and the cluster can continue to provide external services. Compared with manual methods, it saves time wasted on fault discovery and greatly improves availability.

5. Conclusion

The application of data storage technology in DSs is a technology that synchronously and dispersedly stores various data and information generated in the Internet in multiple independent devices, and can use the network to connect thousands of storage nodes together to support Data continues to grow. At the same time, DSs have the advantages of high access, high performance and low cost, and are widely used in large enterprises. In this paper, the data storage and high availability problems of DSs are optimized, and erasure codes and HA schemes are introduced into DSs. Experiments show that both optimization schemes are conducive to promoting the efficient operation of DSs.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Umashankar M L. An efficient hybrid model for cluster head selection to optimize wireless sensor network using simulated annealing algorithm. Indian Journal of Science and Technology, 2021, 14(3):270-288.

[2] Baldovino J, Millan-Paramo C, Izzo R, et al. CO2 and cost optimization of reinforced concrete footings over a lime-treated soil using modified simulated annealing algorithm. Inge CUC, 2020, 16(1):1-16.

[3] Yildiz A R, Bureerat S, Kurtulu E, et al. A novel hybrid Harris hawks- simulated annealing algorithm and RBF-based metamodel for design optimization of highway guardrails. Materials Testing, 2020, 62(3):1-15.

[4] Attiya I, Elaziz M A, Xiong S. Job Scheduling in Cloud Computing Using a Modified Harris Hawks Optimization and Simulated Annealing Algorithm. Computational Intelligence and Neuroscience, 2020, 2020(16):1-17.

[5] Bouddou R, Benhamida F, Zeggai A, et al. The Dynamic Economic Dispatch in An Integrated Wind-Thermal Electricity Market Using Simulated Annealing Algorithm. Przeglad

Elektrotechniczny, 2020, 96(11):pp. 55-60.
[6] Pinheiro C C, Fernandes C, Taglialenha S. Algoritmo Simulated Annealing Para Roteirizao De Veculos Em Uma Empresa De Outsourcing Simulated Annealing Algorithm For Vehicle Routing In A Outsourcing Company. Colloquium Exactarum, 2020, 11(3):1-16.
[7] Redi A, Maula F R, Kumari F, et al. Simulated annealing algorithm for solving the capacitated vehicle routing problem: a case study of pharmaceutical distribution. Jurnal Sistem dan Manajemen Industri, 2020, 4(1):41-49.
[8] Hanine M, Benlahmar E H. A Load-Balancing Approach Using An Improved Simulated Annealing Algorithm. Journal of Information Processing Systems, 2020, Vol. 16(Feb. 2020):pp. 132-144.
[9] Eroglu D Y, Orbak Y. Simulated Annealing algorithm and implementation software for fabric cutting problem. Tekstil ve Konfeksiyon, 2020, 30(1):10-19.
[10] Shrestha L, Sheikh N J. Multiperspective Assessment of Enterprise Data Storage Systems: The Use of Expert Judgment Quantification and Constant Sum Pairwise Comparison in Finding Criteria Weights. Open Journal of Business and Management, 2021, 09(2):955-980.
[11] Bhagi A, Sarkar A, Sethuraman V, et al. Blockchain technology for immunisation data storage in India: opportunities for population health innovation. BMJ Innovations, 2021, 8(1):1-3.
[12] Yoneda N, Nobukawa T, Morimoto T, et al. Common-path angular-multiplexing holographic data storage based on computer-generated holography. Optics letters, 2021, 46(12):2920-2923.
[13] Hiranaga Y, Cho Y. Material Design Strategy for Enhancement of Readback Signal Intensity in Ferroelectric Probe Data Storage. IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control, 2021, 68(3):859-864.
[14] Satti F A, Hussain M, Hussain J, et al. Unsupervised Semantic Mapping for Healthcare Data Storage Schema. IEEE Access, 2021, PP(99):1-1.
[15] Nguyen T T, Cai K, Immink K, et al. Capacity-Approaching Constrained Codes With Error Correction for DNA-Based Data Storage. IEEE Transactions on Information Theory, 2021, 67(8):5602-5613.
[16] Samala R K, Kotapuri M R. Distributed Generation Allocation in Distribution System using Particle Swarm Optimization based Ant-Lion Optimization. International Journal of Control and Automation, 2020, 13(1):414-426.
[17] Tiendrebeogo T, Diarra M. Big Data Storage System Based on a Distributed Hash Tables System. International Journal of Database Management Systems, 2020, 12(5):1-9.
[18] Tebbi A, Chan T H, Sung C W. Multi-Rack Distributed Data Storage Networks. IEEE Transactions on Information Theory, 2019, 65(10):6072-6088.