ISSN: 2664-9640

A Multithreaded Erasure Code Distributed Storage System Based on Binary Code

Changhao Wu[1,2] , Sujing Li[1, *]

[1] School of Information Science and Technology, Yunnan Normal University, Yunnan, 650500, China.

[2] GIS Technology Research Center of Resource and Environment in Western China, Ministry of Education, Yunnan Normal University, Yunnan, 650500, China.

Corresponding author: Sujing Li (lisujingchn@gmail.com)

Abstract

With the rapid development of big data and cloud computing on a global scale, the rise of Google, Amazon and other companies, especially the increasing demand for business cloudification under the background of in-depth integration of "Internet plus", the huge amount of data is reliable for the storage system Performance and scalability present serious challenges. The low flexibility and high maintenance cost of traditional centralized storage can sometimes not meet the current storage needs. In terms of ensuring data reliability, the storage methods of raid and triple copy, which have always been in a dominant position, also have the defect of low storage efficiency, making Research on high-performance storage systems came into being. The research in this paper is a multi-threaded erasure code distributed storage system based on binary coding. Compared with traditional data storage systems, it has the advantages of small load on a single node, low construction cost, low maintenance cost, low storage cost, strong reliability, and high storage utilization.

Keywords

Erasure Code, Distributed Storage, Multithreading.

1. Introduction

In the era of big data, the huge amount of data poses a severe challenge to the performance of the storage system. Compared with the traditional centralized storage model, the distributed storage system has strong service capabilities, low cost and extremely easy expansion[1], and has been extremely widely used in the storage management of big data. But at the same time, how to ensure the correct switching and recovery of the system when a single node is abnormal, and to ensure the overall reliability of the system is also an important issue. In the storage field in the past, raid[2], 3 copies [3]have always been the main means of data storage to ensure reliability. However, its high storage cost has discouraged many merchants of distributed cloud storage. With the continuous development of storage technology, the application of erasure codes has attracted more and more attention. Erasure codes can achieve reliability close to the three-copy storage mode under the premise of greatly reducing storage costs[4], reducing data redundancy[5], to reduce storage pressure and improve access efficiency, thereby greatly reducing enterprise storage costs.

H Dau [6]has designed p-reconstructible μ-secure n, k erasure coding schemes ($0 \le \mu < k$, $1 \le p \le k - \mu$, $p \mid (k - \mu)$), which encode k-μ information symbols into n coded symbols. A study [7] proposed an improved decoding algorithm which is an erasure code decoding algorithm based on matrix that can reconstruct data elements and redundant elements at the same time. Research by J Zhang [8] showed that: due to the advantages of high storage efficiency, erasure

ISSN: 2664-9640

codes have been widely used in storage systems, and access efficiency has become the main disadvantage due to the additional data retrieval and decoding of unavailable data.

The research in this paper is a multi-threaded erasure code distributed storage system based on binary coding. Compared with traditional data storage systems, it has the advantages of small load on a single node, low construction cost, low maintenance cost, low storage cost, strong reliability, and high storage utilization.

2. Principle of Erasure CodeSection Headings

At present, there are three main types of erasure coding technology in distributed storage systems, array erasure codes (Array Code: RAID5, RAID6, etc.), RS Reed-Solomon erasure codes, and LDPC (Low Density Parity Check Code). Take the most widely used RS code as an example: it is a coding algorithm based on a finite field. The finite field is also called Galois Field, named after the famous French mathematician Galois. In RS code, GF(2w) is used, where 2w >= n + m. The encoding and decoding of RS code is defined as follows:

Coding: Given n data blocks D1, D2...Dn, and a positive integer m, RS generates m code blocks according to n data blocks, C1, C2...Cm.

Decoding: For any n and m, any n blocks from n original data blocks and m coded blocks can be used to decode the original data, that is, the RS can tolerate at most m data blocks or coded blocks to be lost at the same time.

3. The system design

This system is mainly divided into five parts, namely file blocker, encoder, I/O transmitter, distributed storage node, and decoder. The working principle of these four parts is shown in Figure 1:

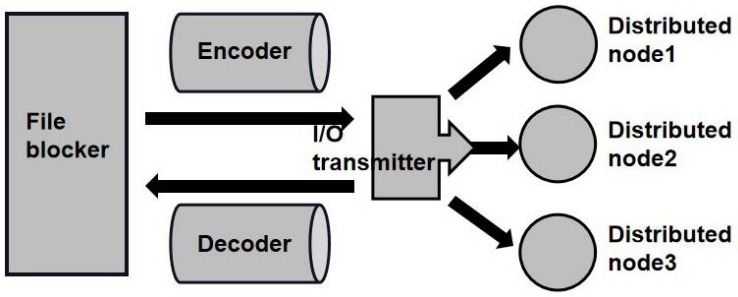

Figure 1: Architecture of erasure code distributed storage system

Local file blocker: In (n, k) erasure coding, the original data object is divided into k data blocks of the same size before being encoded. This is an important step before encoding. Since the size of the file is not necessarily divisible by k, the local file blocker also plays an important role in padded zeros at the end of the file. The specific zero-padding rule is that if the size of the original data object cannot be divisible by k, the minimum number of 0s is added to the end of the original data object until the original data object can be divisible by k. After dividing the file into k data blocks of equal size, it does not end. In order to save memory, the divided data cannot be directly encoded, but a smaller data strip is read from each data block each time. In this way, k data strips of equal size are generated, and then delivered to the encoder to generate encoded data by the encoder.

Encoder: The encoder is the main part of the system. Its function is to divide the strips by the blocker and encode them through the encoding matrix to generate k data strips and m check strips.

Decoder: In the erasure code where data is divided into k blocks, the work of the decoder first needs to detect whether there is a damaged data block. If a data block is detected to be damaged,

ISSN: 2664-9640

the decoder skips the data block and continues to read the following data block , Or check block, until the k-th block is read, that is, the first k undamaged data blocks are read to form a decoding matrix, and the corresponding k is obtained by diagonalization and Gaussian elimination in a finite field The inverse matrix of the order encoding matrix is left multiplied by the decoding matrix by the inverse matrix, and the result is the matrix composed of the original data block.

I/O transmitter: The encoded data is k data blocks and m check blocks. In the distributed storage architecture, they are stored in k+m storage nodes, through byte stream I/O. Transfer the encoded file to the other end of the storage. The I/O transmitter used in this system architecture is the open source Seafile distributed storage server. Seafile is an open source file cloud storage platform that solves the problems of centralized file storage, synchronization, and multi-platform access, focusing on security and performance. Configure n servers with seafilesever, record their IP port numbers in the I/O transmitter, the data blocks that are divided into blocks on the client side can be synchronized with the cloud server as long as they are stored in the local virtual cloud disk, and have Automatic deduplication function.

Distributed storage node: The distributed storage node of the system is a cloud server based on Tencent Cloud and has the following configurations:

Table 1: Configuration of distributed storage nodes

Host type	Standard S2
operating system	Ubuntu Server 16.04.1 LTS 64-bit
Mirror id	img-pyqx34y1
CPU	1
RAM	2
bandwidth	1
Public IP	58.87.122.103
Intranet IP	172.21.0.2
System disk type	CLOUD_BASIC
System disk size	50
Seafile version	seafile-server_6.2.5_x86-64

The erasure code engine used in this experiment is the open source Jerasure engine, which uses the c/c++ language to develop an open source erasure code library that can be directly applied to distributed storage systems. Jerasure implements classic erasure codes such as RS code and CRS code by serial encoding. And supports other erasure code extensions. Among the common open source libraries, Jerasure is an erasure code library with higher coding efficiency. It has been widely tested and applied in distributed storage systems.

Erasure codes often use logarithmic look-up tables in the encoding process. Among them, the operation in the finite field in the Jerasure engine uses the look-up table method to calculate addition and multiplication. This method has low efficiency and is a key factor affecting the encoding and decoding time. Therefore, the optimized method is to use a binary matrix to directly convert addition and multiplication into XOR. In addition, using multiple threads in the encoding process can save encoding time. This study sets up two sets of experiments for comparison, group one is a single-threaded coding method using logarithmic look-up table method, and group two uses a 4-threaded coding method using binary matrix method. The settings are as follows:

ISSN: 2664-9640

Table 2: Experiment group one setting

Threads	1
Encoding type	Rs
Number of blocks	4
Check the number of blocks	2
Number of failed bands	1
Finite field calculation method	Log look-up table

Table 3: Experiment group two settings

Threads	4
Encoding type	Rs
Number of blocks	4
Check the number of blocks	2
Number of failed bands	1
Finite field calculation method	Binary matrix method

4. Results

It can be seen from Figure 2 that the encoding method that uses a multi-threaded binary array has much lower encoding and decoding time than the original encoding method that uses a look-up table to perform operations within a finite field. Since erasure codes are not affected by each group of stripe codes in encoding, we use 4 threads to encode at the same time. Both encoding and decoding rates are greatly optimized. When the block size is between 128k-256k, the encoding and decoding time is the shortest and the rate is the highest. We can select the appropriate block size according to the actual situation. When the block is larger, the encoding and decoding time tends to be more stable.

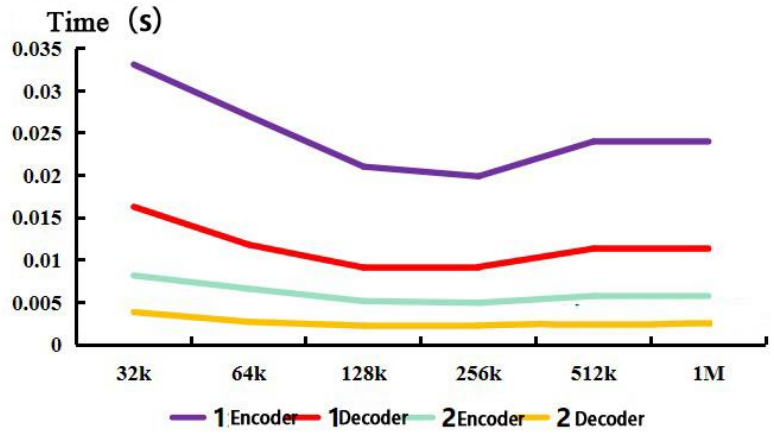

Figure 2: Codec processing time of two groups

ISSN: 2664-9640

5. Conclusion

The multi-threaded erasure code distributed storage system based on binary coding uses distributed storage to improve the scalability of the system. The optimization method of multi-thread and binary matrix makes the coding and decoding time of erasure codes greatly reduced. Although the representation space is increased by using the binary matrix method, it saves time. By using the multi-threaded method, most of the performance of the cpu is sacrificed in exchange for time. The combination of the two methods greatly optimizes the encoding and decoding time and greatly improves the overall system performance.

References

[1] S. Gao, J. Liang, S. Wu, and Y. Xu, "A ROTATED DEPLOYMENT-BASED EXPANSION SCHEME FOR RAID6 DISTRIBUTED STORAGE SYSTEM," Computer Applications and Software, 2016.

[2] L. Mei, F. Dan, L. Zeng, J. Chen, and J. Liu, "A Stripe-Oriented Write Performance Optimization for RAID-Structured Storage Systems," in IEEE International Conference on Networking, 2016.

[3] N. Maki, T. Imazu, and H. Yamamoto, "Storage System and Copy Method," 2009.

[4] L. Zheng and L. I. Xiaodong, "Low-cost Multi-node Failure Repair Method for Erasure Codes," Computer Engineering, 2017.

[5] H. Y. Lin, L. P. Tung, and B. Lin, "An Empirical Study on Data Retrievability in Decentralized Erasure Code Based Distributed Storage Systems," 2013.

[6] H. Dau, W. Song, A. Sprintson, and C. Yuen, "Secure Erasure Codes With Partial Reconstructibility," IEEE Transactions on Information Theory, vol. PP, no. 99, pp. 1-1, 2020.

[7] D. Fan, F. Xiao, and D. Tang, "A New Erasure Code Decoding Algorithm," International Journal of Network Security, vol. 21, no. 3, pp. 522-529, 2019.

[8] J. Zhang, L. I. Shanshan, X. K. Liao, S. L. Peng, X. D. Liu, and Z. Y. Jia, "HeMatch: A redundancy layout placement scheme for erasure-coded storages in practical heterogeneous failure patterns," Science China, no. 06, pp. 67101-067101, 2015.