Optimization of Replica Technology with Two-Stages Dynamic Factor in Cloud Environment

Jun Qin[1,2](✉), Ping Zong[2,3], and Yanyan Song[1]

[1] Communication University of China Nanjing, Nanjing 211172, China
[2] Nanjing University of Posts and Telecommunications, Nanjing 210003, China
{qjun,zong}@njupt.edu.cn
[3] Nanjing University of Science and Technology Zijin College, Nanjing 210003, China

Abstract. The replica technology in cloud storage can not only maintain the high availability of the system, but also improve the performance of the system as a whole. Based on the analysis of the shortcomings of the static copy mechanism of the existing Hadoop distributed file system, this paper dynamically adopts different adjustment strategies for files with different heat, and completes the copy factor adjustment through two stages of filtering and adjustment, which can improve the access performance and avoid the waste of storage resources. The experimental results show that the improved replica factor adjustment strategy, reduced the average response time of system jobs and advance the performance of data access.

Keywords: Cloud Environment · Big Data · Replica · Adjustment Factor

1 Introduction

In the distributed cloud storage environment with largescale nodes, the system availability problems caused by natural disasters and improper human operation cannot be ignored [1]. In addition, the cluster is often composed of a large number of cheap machines, and software and hardware failures become the normal behavior [2]. Therefore, the cloud storage system must provide a series of high availability mechanisms to ensure that the availability of the whole system is not affected, High availability (HA) means that the whole system can ensure more than 99% of the available time, which there are certain differences in service availability between different service providers and their services at different levels, so that the unavailable time of services and its adverse effects can be controlled within a reasonable range. The system can continue to provide services in case of node failure. High availability and performance are a very important factor for the further development of cloud storage. The rational use of replica technology can not only maintain high data availability, but also have important application value for reducing access delay and improving system integrity.

Replica technology is an important way to ensure the high availability mechanism of the system. The redundant storage is setting a certain replica factor for data blocks in the cloud storage system. The redundant replicas is distributed on different nodes in the cluster. So the impact of node failure on system services can be reduced. In dynamic replica technology, three main problems need to be solved intensively: replica factor adjustment timing, replica factor adjustment scheme and replica placement strategy.

2 Hadoop Default Replica Policy

Distributed file system HDFS [3] is mainly based on general distributed machine clusters to provide distributed storage services with high reliability, high performance, scalability and strong fault tolerance for the whole system. HDFS adopts a typical master/slave architecture [4], which is mainly composed of NameNode, DataNode and Secondary NameNode is showed in Fig. 1.

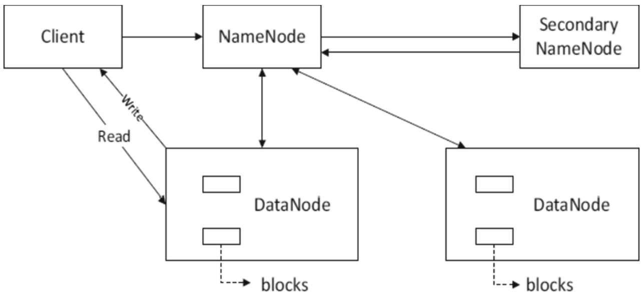


Fig. 1. Main structure diagram of HDFS.

2.1 Default Replica Management Mechanism

At present, the mainstream replica factor management strategies are mainly divided into two kinds: static strategy and dynamic strategy.

The static replica management strategy is a relatively simple replica factor implementation mechanism through the pre-configured replica factor, which cannot adapt to the changes of the system environment due to the lack of flexibility. Too low replica factor will affect the reliability and performance of the system. Too high replica factor will greatly increase the consumption of storage space. Dynamic replica factor strategy can better adapt to the changes of user access frequency, storage space, system bandwidth, system response time and network topology. Dynamically adjusting the replica factor at runtime can adaptively increase, reduce or maintain the number of replicas according to different evaluation indicators, which can often better meet the data access needs of multi-users and heterogeneous storage environment in cloud computing.

2.2 Main Problems and Analysis

The existing distributed file system HDFS adopts the static copy mechanism [5] by default. Especially in the multi-user environment [6] in cloud computing, there are great differences in the access frequency of different files by different users [7]. That is, there are great differences in the access heat of files. If a unified copy factor mechanism is adopted for different files with great differences in access heat, the files with high heat cannot meet the needs of high frequency access because the copy factor is too small [8]. The files with low heat waste storage space because they retain too many copies [9].

Dynamic copy mechanism is an effective solution to the impact of file access response time and network load caused by unequal file access popularity [10]. However, while adopting the dynamic replica factor strategy, it also needs to effectively filter the files to be adjusted. If a unified replica factor dynamic adjustment strategy is adopted for all files, it will bring a large consumption of time and space. At the same time, the replica adjustment strategy also needs to be able to effectively respond to the sudden needs of file access and maintain high data access performance in the case of sudden increase in file heat.

3 Dynamic Replica Factor Adjustment Strategy

Data access in the cloud storage system has the principle of the temporal and spatial locality [11]. The temporal locality refers to the higher probability that data with high access heat in the current time will be re accessed in the future. It reflects that the access of some data in the system has a certain correlation in a certain time period. That is, some files may continue to become hot files in a time period until their access heat gradually decreases.

In the process of cloud storage system performance optimization, due to the large-scale nature of the amount of data stored in the cluster in the cloud storage environment, if a high copy factor is adopted for all files, the system availability and response time can be improved, but the system storage resources will cause a serious waste and bring additional management and maintenance costs to the service provider. Therefore, it is necessary to filter out the hot data in the system according to the locality principle of data access, and then increase the copy factor for this small part of hot data to avoid unified copy factor adjustment. At the same time, for low heat files, the copy factor can be maintained at a low number without affecting the availability, so as to reduce the waste of storage resources.

3.1 Basic Idea of Improved Algorithm

Aiming at the shortcomings of HDFS default static replica strategy in the case of uneven distribution of file access heat, and the problem of unified decision-making and adjustment in the existing dynamic replica strategy, this paper proposes an improved adjustment strategy with two-stages dynamic factor. This strategy not only adjusts the replica factor according to the file access heat, but also considers the priority of different file heat. The adjustment decision of copy factor is made according to two different length time intervals, which can well adapt to the sudden increase of file access heat.

3.2 Improved Dynamic Replica Factor Adjustment Algorithm

The improved dynamic replica factor adjustment algorithm first obtains the set of replica factor files to be adjusted according to the file access heat and replica decision factor value, and then adopts different replica factor adjustment strategies for different files.

Description of Copy Factor File Filtering Algorithm to Be Adjusted
According to the file set F input by the cluster and the sum of the decision-making time interval Δt_1 and Δt_2, the filtering algorithm finally outputs the corresponding high heat file set F_h^1, F_h^2 and low heat file set F_c^1 in the two time intervals respectively. The specific steps of the algorithm are as follows:

Algorithm input: the file set $F = \{f_1, f_2 \ldots f_n\}$ in the cluster and the sum of two decision intervals Δt_1 and Δt_2.

Algorithm output: high heat file set F_h^1, F_h^2 and low heat file set F_c^1.

File f_K in t_{now} time access heat FH_k:

$$FH_k = \sum\nolimits_{t_i = t_{now} - \Delta t}^{t_{now}} (a_k(t_i, t_{i+1}) * decay(t_i, t_{now})) \tag{1}$$

Replica decision factors RD_k for documents f_k:

$$RD_k = \frac{\sum\nolimits_{t_i = t_{now} - \Delta t}^{t_{now}} (a_k(t_i, t_{i+1}) * decay(t_i, t_{now}))}{br_k * \sum\nolimits_{j=1}^{n_k} bs_j} \tag{2}$$

Cluster replica decision factor $RD_{cluster}$:

$$RD_{cluster} = \frac{\sum\nolimits_{k=1}^{n} \left(\sum\nolimits_{t_i = t_{now} - \Delta t}^{t_{now}} (a_k(t_i, t_{i+1}) * decay(t_i, t_{now})) \right)}{\sum\nolimits_{k=1}^{n} (br_k * \sum\nolimits_{j=1}^{n_k} bs_j)} \tag{3}$$

High heat file f_K copy factor dynamic adjustment value DV_k:

$$DV_k = \left\lceil \frac{RD_k - RD_{\min}}{RD_{\max} - RD_{\min}} * \lambda \right\rceil \tag{4}$$

Step 1: According to formula (1), (2) and (3), calculate all files in file set F in sequence Δ t1 and Δ t2 replica decision factor in two decision time intervals RD_k^1 and RD_k^2 (where $\Delta t_1 > \Delta t_2$), and the replica decision factor $RD_{cluster}^1$ and $RD_{cluster}^2$ of the cluster in the two decision time intervals.
Step 2: According to the standards of high heat file and low heat file defined in formula (4), obtain high heat file set F_h^1, F_h^2 in two decision intervals Δt_1 and Δt_2 and low heat file set F_c^1 corresponding to time interval Δt_1,Then from F_h^1 and F_h^2 the intersection F_h^I is obtained.
Step 3: If intersection F_h^I is $\varnothing$, then F_h^I in a certain proportion γ ($\gamma \in (0,1)$, which can be adjusted according to the actual situation. RD_k^1 is filtered to a certain extent from high to low priority. Some documents with low copy decision-making factor in F_h^1 shall be eliminated. If intersection F_h^I is not $\varnothing$, the documents in F_h^1 will not be filtered in the further.

Step 4: Output the two decision intervals Δt_1 and Δt_2 with the corresponding high heat file set F_h^1, F_h^2 and low heat file set F_c^1, and the algorithm ends.

Description of File Replica Factor Adjustment Algorithm

The adjustment algorithm obtains two decision intervals Δt_1 and Δt_2 with the corresponding high heat file according to the output of the above filtering algorithm.

Set F_h^1, F_h^2 and the set F_c^1 of low heat files in the time interval Δt_1 is dynamically increased arming to the corresponding replica factor for high heat files, meanwhile is dynamically decreased the arming to the corresponding replica factor for low heat files. In order to ensure the reliability of file access in the system, the minimum replica factor is set as $\lambda - 1$. That is the minimum replica factor is 2. The algorithm processing steps of file copy factor adjustment are described as follows:

Step 1: For the high heat file set F_h^1, F_h^2 output by the above filtering algorithm, if the intersection F_h^I of F_h^1 and F_h^2 is $\varnothing$, According to formula (4), the dynamic adjustment value of the replica factor of the files in the file set within the decision-making time interval Δt_1 is calculated successively. For the high heat file set F_h^2, the dynamic adjustment value DV_k^2 of the replica factor will take the dynamic adjustment value λ of the value with the maximum replica decision factor. That is, according to the calculated dynamic adjustment value of the replica factor, adjust the number of existing replica factors to the sum of HDFS default static replica factor λ and dynamic adjustment value of replica factor;

Step 2: If the intersection F_h^I of F_h^1 and F_h^2 is not $\varnothing$, for the files in the set $F_h^1 - F_h^I$, the dynamic adjustment value of the replica factor DV_k^1 of the files in the file set within the decision-making time interval Δt_1 is calculated in turn, and for the files in the set F_h^2, the dynamic adjustment value λ of the replica factor corresponding to the maximum replica decision factor is taken, and then the dynamic adjustment value of the replica factor is calculated according to the dynamic adjustment value of the replica factor to adjust the number of existing replica factors to the sum of HDFS default static replica factor λ and replica factor dynamic adjustment value.

Step 3: For the low heat file set F_c^1 output by the above filtering algorithm, adjust its replica factor as $\lambda - 1$, that is 2.

4 Simulation Verification and Analysis

In order to verify the improvement of the dynamic replica factor adjustment algorithm on the system performance, this paper builds a Hadoop distributed experimental environment in Alibaba cloud environment for simulation experiment verification, and compares and analyzes the impact of the default replica mechanism and the dynamic replica factor adjustment algorithm with two-stages on the average response time of jobs.

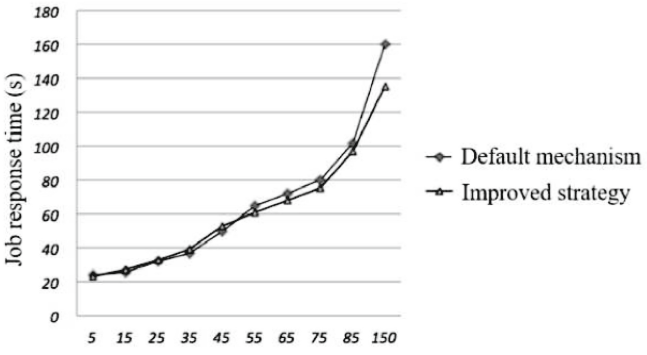
Based on the master/slave architecture of Hadoop, a distributed simulation environment is built with the help of Alibaba cloud ECs.

In order to simulate the difference of users' access heat to different files in the cluster, set the access times of files in the cluster to 5, 15, 25, 35, 45, 55, 65, 75, 85 and 100 groups per minute respectively, so as to reflect the change of users' access heat to files. In this experiment, for the determination of high heat files and low heat files, set the parameters is set as (a = 1.2, B = 0.8, Y = 0.8). The decision-making time interval is adjusted as $\Delta t_1 = 45$ s, $\Delta t_2 = 5$ s. Four groups of files with different sizes (32 MB, 64 MB, 128 MB and 256 MB are set. The comparison of the average response time of system jobs under different access heat is showed in Fig. 2.

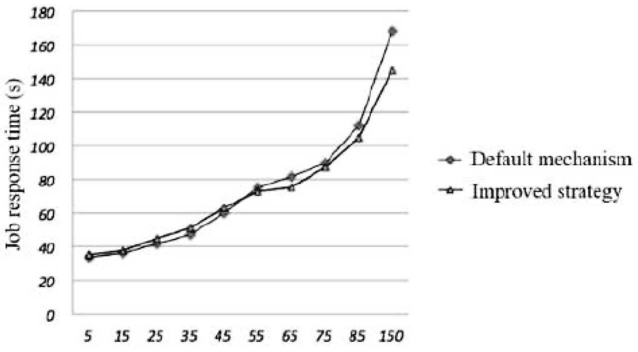
By analyzing the relevant results in Fig. 2, it can be seen that when the file access heat is low, the heat of the file has little impact on the dynamic adjustment of the replica factor, and it will not even trigger the dynamic increase or decrease of the replica factor. At the same time, because the algorithm itself needs to consume certain resources and time in the dynamic calculation process, the average job response time of the improved replica factor adjustment mechanism will be longer than that of the default static replica mechanism. That is, the dynamic replica factor adjustment algorithm cannot effectively improve its performance.

With the increasing popularity of file access, the dynamic replica factor adjustment algorithm begins to show some performance advantages. By observing the data results in Fig. 2, it can be seen that when the file access frequency per minute reaches 50–60 times, the dynamic increase of replica factor is triggered. Therefore, for high-popularity files, there will be multiple replicas to provide access services at the same time. So it can effectively reduce the file access competition under high heat concurrent access and shorten the response time of the job. For high heat files, it can timely and dynamically increase the replica factor to meet the continuous or sudden high heat access requirements. As a result it can effectively shorten the average response time of the system job.

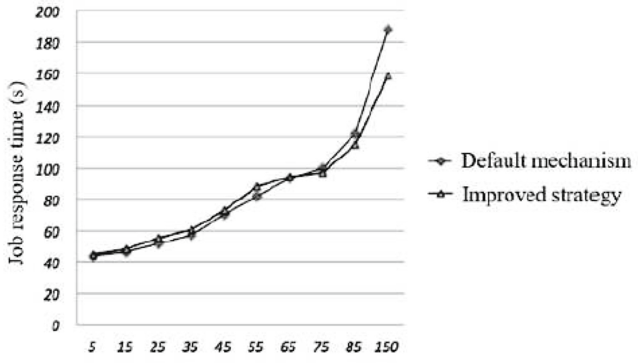
Since the access recognition of file D may decline over time, the dynamic replica factor adjustment strategy can also maintain the minimum number of replica factors on the premise of ensuring the basic availability of replicas, reduce the corresponding replica factors according to the attenuation of access recognition, and return to the normal or default number, so as to avoid unnecessary waste of storage space caused by excessive replica factors. At the same time, it also reduces the cost of system consistency maintenance. This improved strategy in this paper can take into account the availability, service performance and rational use of resources as much as possible, and is suitable for the improvement of service performance of multi-users file access in cloud environment.



(a) 32MB file job response time changed with access heat

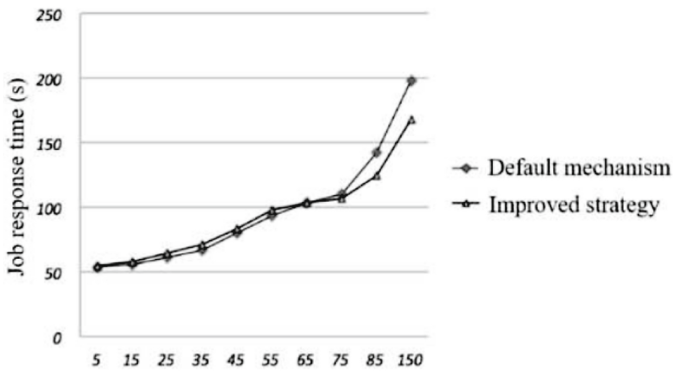


(b) 64MB file job response time changed with access heat



(c) 128MB file job response time changed with access heat

Fig. 2. Comparison of job response time with access heat



(d) 256MB file job response time changed with access heat

Fig. 2. (*continued*)

5 Conclusion

In view of the limitations of HDFS default static replica strategy in the case of uneven distribution of file access heat, and the problem of unified decision-making and adjustment in the existing dynamic copy strategy when adjusting the replica factor, this paper proposes an improved dynamic replica factor adjustment strategy with two-stages, which not only adjusts the replica factor according to the file access heat, but also considers the priority of different file heat, And the adjustment decision of replica factor is made according to two different length time intervals, which can well adapt to the sudden increase of file access heat, and has good application value. The simulation results also verify the effectiveness of the improved strategy mentioned in this paper.

References

1. Prakash, M., Manikandan, S., Sambit, S., Sanchali, D.: An efficient technique for cloud storage using secured deduplication algorithm. J. Intell. Fuzzy Syst. **41**(02), 2969–2980 (2021)
2. Yanxia, S.U., Qingsheng, W.A.N.G., Yongle, C.H.E.N.: Data sharing of cloud storage based on data segmentation. Comput. Eng. Des. **42**(10), 2742–2747 (2021)
3. Dong, C., Zhang, X., Cheng, W., Shi, J.: Performance optimization of distributed file system based on new type storage devices. J. Comput. Appl. **40**(12), 3594–3603 (2020)
4. Jun, L.I.U., Fangling, L.E.N.G., Shiqi, L.I., Yubin, B.A.O.: A distributed file system based on HDFS. J. Northeastern. Univ. (Nat. Sci.). **40**(06), 795–800 (2019)
5. Dhaya, R., Kanthavel, R., Venusamy, K.: Cloud computing security protocol analysis with parity-based distributed file system. Ann. Oper. Res. **11**(30), 1–20 (2021)
6. Uma, K., Kumar, P., Srinivasu, S.V.N., Nachappa, M.N.: Sqoop usage in Hadoop Distributed File System and Observations to Handle Common Errors. Int. J. Recent Technol. Eng. **9**(04), 452–454 (2020)
7. Elkawkagy, M., Elbeh, H.: High performance hadoop distributed file system. Int. J. Networked Distrib. Comput. (08)3, 119–123 (2020)
8. Wang, B., Wang, K., He, Z., Gao, W., Wei, X.: Optimizing file temperature prediction of Markov based on cuckoo search. Comput Eng. Des. **42**(11), 3121–3127 (2021)

9. Cheng, Z., Wang, L., Cheng, Y., Chen, G., Hu, Q., Li, H.: File access popularity prediction for hierarchical storage for high-energy physic. Comput. Eng. **47**(02), 126–132 (2021)
10. Wang, J., Liu, Y., Yu, C., Wang, M., Liu, X.: Construction of group repairable codes for non-uniform fault protection. J. Beijing Univ. Posts Telecommun. **42**(05), 75–82 (2019)
11. Li, J., Hou, R.: Simulation of information transmission redundancy elimination in big data access. Comput. Simul. **37**(03), 148–151+177 (2020)