

Dynamic Replica Creation Strategy Based on File Heat and Node Load in Hybrid Cloud

YaHui Zhao, ChunLin Li, LaYuan Li, Peng Zhang

Computer Science and Technology

Wuhan University of Technology

No.1186, Heping Boulevard, Wuchang District, Wuhan, Hubei

CHINA

1140614849@qq.com, chunlin74@aliyun.com, lilayuan26@163.com

Abstract—Replica creation strategy is one of the important research directions of the distributed file system in the hybrid cloud environment. However, traditional replica creation strategy just simply calculated the file heat based on the number of accesses to the file within a period of time. Besides, creating too many copies will seriously affect the performance of the node without considering the node load. In order to solve this problem, the improved dynamic replica creation strategy based on file heat and node load is presented in this paper combined with the characteristics of the hybrid cloud environment. File heat of history and current access frequency of three cycles and change rate of file are considered comprehensively in the calculation of the heat based on LRFU (Least Recently Frequently Used). Combined with the node load, the average heat and the average load are used to adjust the number of copies in this paper, which can adapt to the changes of the environment dynamically. Experiments show that with changes of file access and traffic intensity, the improved strategy is sensitive to access frequency, which can adaptively adjust the number of copies, reduce the average response time, and achieve better load balance of cluster.

Keywords—Hybrid cloud, Replicas, File heat, Node load, Load balance

I. INTRODUCTION

Hybrid cloud, which is the combination of public cloud and private cloud in the target architecture, has the characteristics of low cost and high scalability [1] [2] [3]. The node load capacity of hybrid cloud is very different. There are a large number of data files in hybrid cloud and each file has different amount of access. In the cloud storage system, the replica technique [4] is the key technology to improve the reliability. For example, it is used in HDFS [5] and Dynamo [6] to ensure the data reliability of the cloud storage system. However, when there are more file with highly-visited gathered in some nodes with poor storage capacity, it will cause a hot issue which may reduce the overall performance of the system.

Researching on creation strategy of the number of copies can provide users with more quality and efficient data storage services, so it has great practical significance and application prospects.

Traditional research shows that data access is unexpected, and the maintenance time of partial burst access requests is very short [7]. In this case, considering that the file heat is too high and creating a new copy not only occupy a large amount of system resource, but also make the system load unbalance which will result in a high latency [8]. In the study of file heat, a lot of storage systems are only a simple calculation of the number of files in a period of time for the statistics of the heat [9] [10]. Domestic and foreign scholars try to predict the file heat by using multiple frequencies, but it is difficult to take into account the heat of files and node load balancing [11] [12].

In order to solve these problems, this paper proposes a dynamic replica creation strategy based on file heat and node load. Our contributions in this paper are: (1) Considering the limitation of the default strategy in HDFS, this paper proposes a initialization strategy by introducing the concept of file reliability, so that it can dynamically initialize the number of copies according to the user needs. (2) Considering file heat of history and current access frequency of three cycles comprehensively based on the LRFU, this paper introduce the change rate of the file into the heat calculation formula. Besides, this paper set different impact factors for file heat and node load, and it can adjust the number of copies dynamically, reduce the average response time and improve the overall performance of the cluster.

The remainder of this paper is organized as follows. We discuss related work in Sec. II. Sec. III introduces the definition of the relevant parameters and gives the formula of the number of copies. The implementation of the dynamic replica creation strategy is introduced in

Sec. IV. Sec. V introduces comparative experiments to verify the effectiveness of the proposed algorithm. Sec. VI concludes the paper and highlights the direction for our future work.

II. RELATED WORK

Replica creation consists of static and dynamic replica creation strategies. Dynamic replica creation strategy is the most practical application in the related research and the existing system. Although the dynamic replica creation strategy is more complex than the static replica creation strategy, it is more suitable for the changing data access characteristics.

Some researchers have studied the strategy of replica creation based on the frequency of access. A hierarchical dynamic replication strategy was proposed [8]. When a file is frequently accessed in the past period of time, a copy of the file is deployed at a particular node; the policy has a better job execution time, which is particularly evident when the storage space of the node is small. The concept of acceleration in physics was applied to the dynamic replica creation strategy [9]. The access acceleration shows a change in the file access frequency over a period of time, which can predict the hot file in the next cycle and find the best node to create a new copy. The dynamic replica management strategy can focus on the next cycle of forecasting without the need to concentrate on the current copy of the file. REN et al. [10] proposed a method of calculating the file heat based on file access timing and LRFU algorithm, and the strategy dynamically adjust the copy according to the file heat. A similarity prediction function was proposed [11], and a novel method of data copy creation is constructed by comparing the similarity of the file transfer path between the history and the current.

In addition, Some scholars have researched replica creation strategy based on the node load. The load balancing status and reliability of data nodes were analyzed [12], and a dynamic replica management mechanism is proposed. Kyungbaek Kim [13] proposed a time-related replication strategy which use the node interaction information to prevent the loss of replica. Xin Sun [14] proposed a dynamic replication strategy based on access costs for the purpose of reducing delays. They are designed for distributed environments and cannot adapt well to the cloud storage system. Those strategy neither reduce the network latency nor provide user access efficiency.

Although scholars have carried out some researches on the replica creation technology, however, they just have made a statistic about file accessing time when calculate the file heat. Even some strategies use multiple

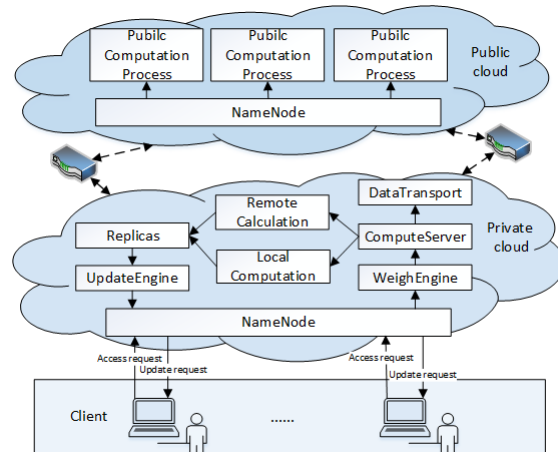


Figure. 1. The model of dynamic replica creation strategy based on file heat and node load

frequencies to predict the file heat, it is difficult to take into account the file heat and node load balancing. Therefore, this paper studies the dynamic replica creation strategy based on file heat and node load.

III. DYNAMIC REPLICA CREATION STRATEGY BASED ON FILE HEAT AND NODE LOAD

When a new file is generated in the cloud storage system (such as uploading files), the first need to determine is the number of file copies. The replicas also needs to be changed dynamically according to the changes in the environment. The initialization strategy and dynamic creation strategy of the copy are proposed in this section, and the model of dynamic replica creation based on file heat and node load is established.

Figure 1 gives an overview of the model of dynamic replica creation based on file heat and node load. The user submits an access request through the client and NameNode receives the access request and distributes the request through Server Compute. Private cloud and public cloud communicate each other through Transport Data. Finally, adjustment results of the copy number are fed back to NameNode. Local Computation and Remote Calculation part in the private cloud is the main research contents of this paper. According to the client's access request, those module can do the statistics of the amount of access, calculate file heat, and combined with the node load, adaptively adjust the number of copies, thereby improving the performance of the cluster and reducing the average response time.

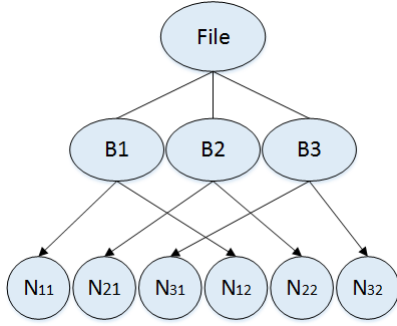


Figure. 2. File block layout

A. The Initialization of the Number of Replicas

Assuming that there are n copies of a file, each copy is designated as m of the same size of the data block using $\{B_1, B_2, \dots, B_m\}$ to express. In order to improve the reliability of the data, it is assumed that each of the data block B_i is placed on a different node N_{ij} , $1 \leq j \leq n$. The copy of the data block is shown in Figure 2, the file is divided into 3 data blocks, B_1, B_2, B_3 , each data block has 2 copies, which are distributed in different data nodes, thus improving the reliability of the data.

Assuming that the reliability of the data block B_i is $P(B_i)$, the online rate of node N_{ij} is $P(N_{ij})$, then:

$$\begin{aligned} P(B_i) &= 1 - P(\overline{N_{i1}}) * P(\overline{N_{i2}}) * \dots * P(\overline{N_{in}}) \\ &= 1 - \prod_{j=1}^n (1 - P(N_{ij})) \end{aligned} \quad (1)$$

The reliability of the file is $P(F)$, and $P(F) = P(B_1) * P(B_2) * \dots * P(B_m)$. In order to simplify the calculation, it is assumed that each node has the same online rate p , then:

$$P(F) = \prod_{i=1}^m (1 - \prod_{j=1}^n (1 - P(N_{ij}))) = (1 - (1 - p)^n)^m \quad (2)$$

When the file is uploaded to the cluster, it was set up different reliability parameters $para$ (The range of the value is 0~1. The bigger the value, the more important the file.) by user according to its importance. The original number of copies can be calculated by the following formula:

$$P(F) = (1 - (1 - p)^n)^m \geq para \quad (3)$$

$$n \geq \log_{1-p}(1 - \sqrt[m]{para}) \quad (4)$$

The value n calculated by the above formula is the initial value of the number of copies to ensure the reliability of the file. Initially, the default file placement policy of HDFS is used to store the file.

B. The Design of Dynamic Replica Creation Strategy

1) *Calculation Method of File Heat Based on the Change Rate of File*: Different from the traditional calculation method, this paper statistics three different periods T_1, T_2, T_3 of file access to calculate the file heat based on LRFU. For a file, a five tuple $(hot_0, k_0, v_1, v_2, v_3)$ is used to describe the file heat. Where hot_0 represents file heat on the last T_1 cycle, k_0 indicates the size of the file at the end of the last T_1 cycle, n_1, n_2, n_3 , respectively, indicates the number of visits of the file in the last statistical period T_1, T_2, T_3 .

f_i are used to represent the frequency of access in the cycle T_i , that is, the number of visits per unit of time. So, $f_i = (V - v_i) / \Delta t_i$ ($i = 1, 2, 3$). V represents the total number of times that the file is accessed. Δt_i represents the difference between the current time and the corresponding recent statistics. Such as $T_1 = 24 \text{ hours}$, the recent statistical time is 0, Δt_1 is the difference between the current time and the 0. So, Δt_1 is less than or equal to T_1 , Δt_2 is less than or equal to T_2 , Δt_3 is less than or equal to T_3 .

In order to reduce the communication flow between nodes, this paper introduces the file change rate σ into the file heat calculation formula. The calculation method is as follows, Where v_1 represents the size of the file at the current time:

$$\sigma = \left| \frac{v_1 - v_0}{v_0} \right| \times 100\% \quad (5)$$

First, we calculate the value V_1, V_2, V_3 of the current time, which indicate file access times of the last statistics time to the current time, and then give the calculation formula of the current file heat considering three cycles of file access frequency and file change rate. μ is the ratio of the difference between the current time and the most recent statistical time T_1 to the T_1 cycle.

$$\begin{cases} V_i = V - v_i & i = 1, 2, 3 \\ \mu = \frac{\Delta t_1}{T_1} \\ hot = (1 - \mu)hot_0 + \mu * \sigma + \mu(f_1 + \frac{(V_2 * \frac{f_2}{T_2} + V_3 * \frac{f_3}{T_3})}{T_1}) \end{cases} \quad (6)$$

The calculation method of average access heat for the file is as follows, Where n represents the number of copies, $Total_f$ represents the total number of files on the cluster.

$$h_{avg} = \frac{\sum_{i=1}^n hot_i}{Total_f} \quad (7)$$

F represents a collection of all the files in a distributed file system. According to the different of heat, it can be divided into three subsets L_H, L_M, L_L , respectively for the hot files set and common set of files and unpopular set of files, then $F = L_H \cup L_M \cup L_L$. Files are sorted based on the value hot_i in descending. Those top files that its heat is greater than the average access heat α are selected as hot files and those that its heat is less than the average access heat α are selected as unpopular files. For any file F_i in the distributed file system, the standard of its heat is as follows:

$$F_i \in \begin{cases} L_H & hot_{F_i} > h_{avg}(1 + \alpha) \\ L_M & h_{avg}(1 - \alpha) \leq hot_{F_i} \leq h_{avg}(1 + \alpha) \\ L_L & hot_{F_i} < h_{avg}(1 - \alpha) \end{cases} \quad (8)$$

2) *Calculation Method of Replicas Based on File Heat and Node Load*: The number of copies of is not only related to the amount of accessing to the file, but also related to the load of the node where the copy is located. The storage node sends the load status information to the master control node periodically. The load of the storage node i is ld_i and the total number of cluster nodes is num , so the average load of the whole cluster node can be expressed as:

$$l_{avg} = \frac{\sum_{i=1}^{num} ld_i}{num} \quad (9)$$

Based on the load of nodes, the nodes are divided into three categories, heavy load nodes, moderate load nodes and light load nodes. For any node $node_i$, the load partitioning criteria are as follows:

$$node_i \in \begin{cases} Lev_H & ld_i > l_{avg}(1 + \beta) \\ Lev_M & l_{avg}(1 - \beta) \leq ld_i \leq l_{avg}(1 + \beta) \\ Lev_L & ld_i < l_{avg}(1 - \beta) \end{cases} \quad (10)$$

Assuming that a file currently has M copies stored on the server $\{S_1, S_2, \dots, S_M\}$ respectively. In the current statistical period, the corresponding load of each server

is $\{ld_1, ld_2, \dots, ld_M\}$. Considering file heat and node load in an integrated manner, the dynamic adjustment formula for the number of copies is as follows:

$$n = \begin{cases} \eta_1 \frac{M * hot_i}{h_{avg}(1 - \alpha)} + \eta_2 \frac{\sum_{i=1}^M ld_i}{l_{avg}(1 - \beta)} & (if F_i \in L_L \cap node_i \in Lev_L) \\ \eta_1 \frac{M * hot_i}{h_{avg}(1 + \alpha)} + \eta_2 \frac{\sum_{i=1}^M ld_i}{l_{avg}(1 + \beta)} & (if F_i \in L_H \cap node_i \in Lev_H) \\ M & (otherwise) \end{cases} \quad (11)$$

η_1 and η_2 are the influence factors of file heat and node load. Users can set different values according to their own preferences. besides, $\eta_1 + \eta_2 = 1$.

IV. THE IMPLEMENTATION OF DYNAMIC REPLICA CREATION STRATEGY BASED ON FILE HEAT AND NODE LOAD

The adaptive dynamic replica creation strategy in this paper considers the heat of files and the load of nodes that is different with the traditional replica creation strategy based on threshold. First, it can avoid the impact of sudden access to the value of file heat; Second, when the change occurs in the contents of the file and the two file is same, the update of small file are superior to those large files, which effectively reduce network traffic and make sure that you get the latest data on each visit as far as possible; Finally, the calculation of the heat and the node load is based on the average value of the current three cycles, which can adapt to the changes of the environment dynamically.

The following is a detailed description of the algorithm mentioned in this paper: initialize the number of copies of files and calculate the heat of all the files in the cluster(line 1~7); then, sort hot-set Hot based on the value hot_i in descending (line 8); moreover, calculate the average access heat of the file and divide files into hot files, common files and unpopular files(line 9~10); furthermore, calculate the load of the nodes in the current three cycles, the total number of nodes and the average load of nodes, and divide the nodes into heavy load nodes, moderate load nodes and light load nodes(line 11~12); and then, according to the heat of the file and the node load, calculate the number of copies of the current file(line 13~21). When file is unpopular and node which the file is placed is light load nodes, if $M > n$, it means that copy number of the current file is too much, and need to delete $M - n$ copies in order to improve the utilization of resources. If $M \leq n$, it indicates that the number of copies of the file is appropriate, do not need to delete. When the file is hot and the host node is heavy

load nodes, if $M < n$, it implies that the file belongs to hot files and the average load of nodes is heavy, need to increase the $n - M$ copies in order to reduce the node load. If $M \geq n$, it signifies that the number of copies is appropriate, do not need to increase. In addition to the above, the number of copies is the current default copy number M ; finally, dynamically adjusts the number of copies of each file according to the resulting set (line 22).

algorithm 1 Dynamic replica creation Algorithm Based on File Heat and Node Load

Input: (G, F, M, Hot) //Node-set G , filesets F , the sets of copy number M , hot-set Hot

Output: n //the set of the replicas

```

1: Begin
2: for each  $(F_i \in F)$ 
3:   According to the reliability parameters  $para$  of the
   file  $F_i$ , initialize the number of copies  $M_i$ 
4:   Calculate the value of file heat  $hot_0$  in the last
   cycle and file access  $v_1, v_2, v_3$ , file size  $k_0$ 
5:   Calculate the value of file heat  $hot_i$ 
6: end for
7: Get a set  $M$  of the file copy
8: Sort  $Hot$  based on the value  $hot_i$  in descending
9: Calculate the average heat  $h_{avg}$ 
10: Divide files into three categories  $L_H, L_M, L_L$  ac-
   cording to the formula (10)
11: Calculate the average load of nodes  $l_{avg}$ 
12: Divide nodes into three categories
    $Lev_H, Lev_M, Lev_L$  according to the formula
   (12)
13: while  $(i < Len)$  //  $Len$  is the length of filesets
14:   if  $(F_i \in L_L \cap node_i \in Lev_L)$ 
15:      $n_i = \eta_1 \frac{M * hot_i}{h_{avg}(1-\alpha)} + \eta_2 \frac{\sum_{i=1}^M ld_i}{l_{avg}(1-\beta)}$ 
16:   else if  $(F_i \in L_H \cap node_i \in Lev_H)$ 
17:      $n_i = \eta_1 \frac{M * hot_i}{h_{avg}(1+\alpha)} + \eta_2 \frac{\sum_{i=1}^M ld_i}{l_{avg}(1+\beta)}$ 
18:   else
19:      $n_i = M_i$ 
20:   end if
21: end while
22: Dynamically adjust the number of copies according
   to the set  $n$ 

```

The algorithm complexity depends on the size of the node file set due to that it needs to traverse the node file, so the time complexity of the algorithm is $O(n)$.

V. EXPERIMENT DESIGNS AND RESULTS ANALYSIS

A. Experimental Environment

In this experiment, the hybrid cloud environment is composed of public cloud and private cloud. The

private cloud consists of eleven local machines, and rent Microsoft Azure as a public cloud. The experiment is conducted on the open source Hadoop distributed platform. Operating system is the Ubuntu14.04 Linux system, Hadoop version for the hadoop-2.7.1. Details configuration and settings of the private cloud are as shown in TABLE I. NameNode is the core of the system, using a high configuration. The other DataNodes are made up of 3 different types of computers to represent the differences in the performance of the nodes. At the same time, the DataNode also serves as the client of the system.

Detailed configuration of the public cloud is as shown in TABLE II.

TABLE I: Configuration and Setting of the Private Cloud

HostName	Cluster Configuration	IP
Master (NameNode)	Intel Pentium 4 CPU@2.53GHz RAM:4G DISK:500G	192.168.203.10
Slave1-Slave3 (DataNode) (Client)	Intel Core Duo T7500 CPU@2.20GHz RAM:2G DISK:500G	192.168.203.11 ~ 192.168.203.13
Slave4-Slave6 (DataNode) (Client)	Intel Core Duo T8100 CPU@2.10GHz RAM:2G DISK:320G	192.168.203.14 ~ 192.168.203.16
Slave7-Slave10 (DataNode) (Client)	Intel Core Duo E2160 CPU@1.80GHz RAM:2G DISK:250G	192.168.203.17 ~ 192.168.203.20

TABLE II: Detailed Configuration of the Public Cloud

Public Cloud	Instance Type	CPU Type
Microsoft Azure	Core:2 RAM:3.5G Bandwidth:10Mbps	Intel(R) Xeon(R) E5-2650 v2 2.60GHZ
	Core:4 RAM:7G Bandwidth:20Mbps	
	Core:8 RAM:14G Bandwidth:30Mbps	

1) *Test Case:* Experiment of the validity of the file heat takes 4 files as the object of assessment. File sizes are 100M, 200M, 400M, 800M, respectively. 10 files are used as the experiment object in the dynamic adjustment of the number of copies. File A, B, C are 100M, 200M, 400M, respectively. The size of D~J is 800M. In the last comparative experiment, 60 files are uploaded to the cluster in Hybrid cloud. Each client simulates 2 to 3 users that randomly read these 60 documents. All the test time is one day.

2) *Parameter Setting:* The parameters involved in the experiments are set as shown in TABLE III.

TABLE III: Experimental Parameter

Parameter	Value	Parameter	Value
$para$	0.8	T_3	0.5
p	0.8	α	0.3
m	2	β	0.3
T_1	24	η_1	0.6
T_2	2	η_2	0.4

Among them, $para$ is the file reliability parameter; p represents the node online rate; m is the number of data block; T_1, T_2 and T_3 is the statistical period and base unit is hour; α and β indicate the partition parameter for files and nodes; η_1 and η_2 are the influence factors of file heat and node load.

3) *Evaluation Criteria*: The validity of the file heat considers the file heat as the evaluation index, and verifies the change of the file heat by using different access frequency to access the same file. The number of copies is taken as the evaluation index in the dynamic replica creation strategy which can verify the effectiveness of the proposed strategy to adjust the number of file copies with different heat and size. The load balance effect of dynamic replica creation strategy based on file heat and node load is measured by the variance formula In the last experiment, so that the effect of the experiment is more obvious. Measure index is the number of times that each data node is accessed. The formula is as follows:

$$S^2 = \sum_{i=1}^{num} \frac{(avcount_i - avcount')^2}{num} \quad (12)$$

Among them, num is the total number of data nodes; $avcount'$ represents the average number of times that the data nodes are accessed; $avcount_i$ is the number of times that the data node i is accessed.

B. Experimental Results Analysis

Before the start of the experiment, the number of copies were initialized. The reliability parameter $para$ of the file is set to 0.8. The node's online rate p is set to 0.8, and the number of data block m is set to 2. According to the formula (4), it can be calculated that the initial value of the number of copies of the file is 2.

1) *The Validity of the File Heat*: Initially, for five tuple $(hot_0, k_0, v_1, v_2, v_3)$, the average access frequency of the test scenario is 100 times / h, at this time $hot_0 = 1.75$.

Figure 3 describes the change trend of file heat(hot) with time. The access frequency of file A is gradually increasing; the access frequency of file B remains stable; in contrast to file A, the access frequency of the file C is

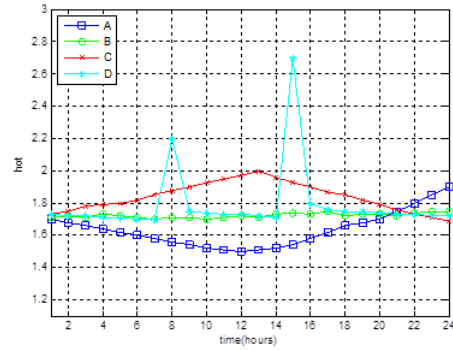


Figure. 3. Hot over time

gradually reduced; file D is used to simulate unexpected access. Under the premise of ensuring the average access frequency, the access frequency of the file A is continuously improved, and its final heat value is higher than B. The access frequency of file C is declining, and its final heat value is lower than B, which shows that the paper's algorithm is sensitive to access frequency. At the same time, the file D has a surge of more than 10 times of the frequency of access in the three statistical periods, but the change of its heat value is very low and can quickly return to the lower heat value. The burst access will only affect the statistics of short period of access frequency, however, the absolute number of visits with the short cycle access frequency is low, which can have a slight on the heat. Considering the short period and long period of access frequency comprehensively, this algorithm can effectively solve the problem of accessing to the file suddenly.

2) *The Dynamic Creation of the Number of Copies*: The access of file is similar to the Zipf distribution, and the amount of access to the first three files is approximately 70% of the total amount of access. Ensure that other factors are the same, the first three files are written to the same content, and the content of the D is unchanged. Figure 4 indicates the number of copies of the 10 test files, the cluster starts a copy of the hot file. File A creates 6 copies, file B creates 5 copies, and 4 copies are created in the file C. Common files and unpopular files do not create copies and keep the number of copies in the original. This adjustment strategy can adjust the number of copies according to the environment. The higher the file change rate is, the higher the heat becomes, the urgent need to quickly update, indicating that this algorithm is sensitive to real-time changes in the file, which reduces the network

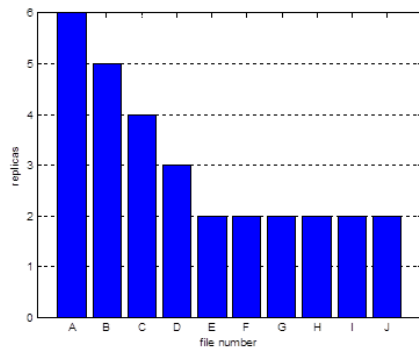


Figure. 4. Replicas over file number

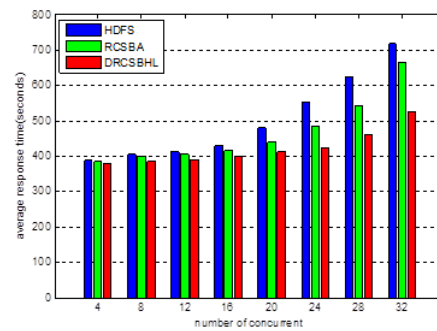


Figure. 5. Average response time over concurrent

traffic flow and abates the impact of hot events on the performance of the system.

3) *The Contrast of Average Response Time and Load Variance*: In order to facilitate the description, dynamic replica creation strategy based on file heat and node load is abbreviated as DRCSBHL. In order to verify the performance, the DRCSBHL and replica creation strategy based on acceleration (abbreviated as: RCSBA) in the literature [9] and default policy of HDFS are compared. Test results are as shown in Figure 5 and Figure 6.

Figure 5 depicts the average response time of the three strategies under different number of concurrent. At the beginning, the number of concurrent is relatively few, and the change of file heat is small, so the average response time of the three strategies is not quite different. But with the increase of the number of concurrent, DRCSBHL creates a corresponding number of copies according to file access heat and node load, so that the average response time does not change much. In contrast to the default copy policy of HDFS, the average response time are increased dramatically. RCSBA is between the two. Figure 6 describes the comparison of the load balancing of the three strategies under different number of concurrent. Corresponding to the results shown in Figure 5, DRCSBHL strategy can create a certain amount of copies dynamically, and make it evenly distributed in each node. Therefore, the load variance is kept at a relatively stable level, just up and down slightly. However, the number of copies of the HDFS policy can not be adjusted according to the load conditions, the fluctuation of load variance is obvious and the phenomenon of load imbalance began to increase. RCSBA also has a larger fluctuation, and its load balancing effect is between HDFS and DRCSBHL. It can be concluded from the experimental results that the DRCSBHL has

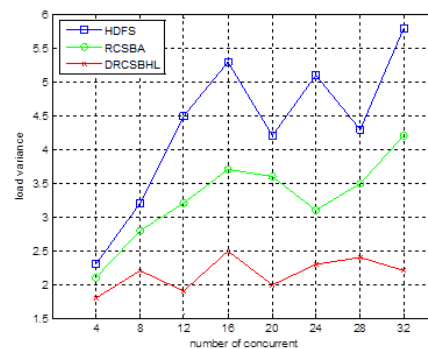


Figure. 6. Load variance over concurrent

better performance than RCSBA and the HDFS default strategy.

VI. CONCLUSIONS

In this paper, the dynamic replica creation strategy is deeply studied in hybrid cloud. The purpose is to adjust the number of copies dynamically, reduce the traffic flow of the hybrid cloud and the average response time of the users, and achieve the load balance of the cluster. In order to achieve this goal, this paper proposes a dynamic replica creation strategy based on file heat and node load. Considering file heat of history and current access frequency of three cycles comprehensively in the calculation of the heat based on LRFU and taking into account the influence of the change rate of the file, then combining with the node load to achieve this strategy. Experimental results shows that compared with the default HDFS replica strategy, the proposed strategy can adaptively adjust the number of replicas, and has obvious advantages in reducing the average response time and achieving the cluster load balancing. Compared

with the replica creation strategy based on acceleration, it also has a great improvement in the overall performance of the cluster in the hybrid cloud environment.

ACKNOWLEDGMENT

The work was supported by the National Natural Science Foundation (NSF) under grants (No.61472294, No.61672397), Key Natural Science Foundation of Hubei Province (No.2014CFA050).

REFERENCES

- [1] Mu S, Chen K, Gao P, et al. LibCloud: Providing High Available and Uniform Accessing to Multiple Cloud Storages[C]. 13th International Conference on IEEE, 2012: 201-208.
- [2] Zhang H, Jiang G, Yoshihira K, et al. Proactive workload management in hybrid cloud computing [J]. IEEE Trans on Network and Service Management, 2014,11(1): 90-100.
- [3] Duan R, Prodan R, Li X. Multi-Objective Game Theoretic Scheduling of Bag-of-Tasks Workflows on Hybrid Clouds [J]. IEEE Trans on Cloud Computing, 2014, 2(1): 29-42.
- [4] Mistarihi H, Yong C. Replica management in distributed file systems[J]. International Journal of Computer Science and Network Security, 2008, 22(8):22-32.
- [5] SHVACHKO K, RADIA S, et al. The hadoop distributed file system[C]. 26th IEEE Symposium on Mass Storage Systems and Technologies, 2010:1-10.
- [6] De CANDLA G, HASTORUN D, et al. Dynamo: amazon's highly available key-value store[C]. 21st ACM SIGOPS Symposium on Operating System Principles, New York, 2007:205-220.
- [7] Anderson E. Capture, conversion, and analysis of an intense NFS workload[C]. Proceedings of Conference on File and Storage Technologies, 2009: 139-152.
- [8] Mansouri N, Dastghaibafard G H. A dynamic replica management strategy in data grid[J]. Journal of Network and Computer Applications, 2012, 35(4): 1297-1303.
- [9] H. Mengxing, Y. Xianglong, et al. A Strategy of Dynamic Replica Creation in Cloud Storage[C]. 1st International Workshop on Cloud Computing and Information Security. Atlantis Press, 2013.
- [10] REN Tao, CHEN Shuyu, ZHANG Yuye, et al. Research on replica adaptive consistency in cloud storage[J]. Computer Engineering and Applications, 2015, 51(4): 83-86.
- [11] Scpahvand R, Horri A, Dastghaibafard G. Replication and scheduling methods based on prediction in data grid[J]. Journal of Applied Sciences Research, 2011, 7(11): 1485-1496.
- [12] G. B. Bao, H. Zhao, et al. The model of data replica adjust to the need based on HDFS cluster[C]. 5th International Conference on Business Intelligence and Financial Engineering, 2012: 532-535.
- [13] Kyungbaek Kim. Time-Related Replication for P2P Storage System[C]. 7th International Conference on Networking, 2008: 351-356.
- [14] Xin Sun, Jun Zheng, et al. Dynamic Data Replication Based on Access Cost in Distributed Systems[C]. 4th International Conference on Computing Sciences and Convergence Information Technology, 2009: 829-837.



Yahui Zhao, born in Zhoukou City, Henan Province, China, 1990, Master candidate. This author's major is computer science and technology, Wuhan University of Technology, China. This author's current research interests include cloud computing and big data.



Chunlin Li (b. December 07, 1974) is a Professor of Computer Science in Wuhan University of Technology. She received the M.S. in Computer Science from Wuhan Transportation University in 2000, and PhD in Computer Software and Theory from Huazhong University of Science and Technology in 2003. Her research interests include cloud computing and distributed computing.



LaYuan Li, born in Wuhan City, Hubei Province, China, 1946. Professor and PhD supervisor. Mr Li served as executive director of the International Federation of High and New Technology Research (IAHT), IEEE, ICCSA, ICCST and other international conferences of the chairman / vice chairman or member. His research interests include Wireless network and Grid computing.



Peng Zhang, born in NanYang City, Henan Province, China, 1992, Master candidate. Ms. Zhang's major is computer science and technology, Wuhan University of Technology, China. Ms. Zhang's current research interests mainly focus on cloud computing.