Data Replica Placement Policy Based on Load Balance in Cloud Storage System

Xiong Fu[1], Jian Li[1], Wenjie Liu[1], Song Deng[2], Junchang Wang[1]
1. School of Computer and Technology, Nanjing University of Posts and Telecommunications
2. Institute of Advanced Technology, Nanjing University of Posts and Telecommunications
Harbin, China
fux@njupt.edu.cn, 1169282043@qq.com, 632933653@qq.com, dengsong@njupt.edu.cn, wangjc@njupt.edu.cn

***Abstract*—Data replication technology has been widely used in cloud storage system. How to select the appropriate data center to place data replication to effectively improve the access performance of cloud storage system became a problem which is worthy of study. To solve this problem we presented a replica placement algorithm based on load balance in this paper. According to the capital value of the data center, the algorithm selects a new replica node and can ensure load balance between the data centers. Related simulation experiments show that the algorithm can quickly and effectively respond to user requests and it is reliable and efficient.**

***Keywords*—cloud storage; replica placement; load balance degree; capital value**

I. INTRODUCTION

Cloud storage is a new concept which extends and develops from cloud computing. It is a model of data storage where the digital data is stored in logical pools, the physical storage spans multiple servers. It is a system that can be used to provide data storage and business access through the application of cluster, network technology, distributed file system and other functions[1]. Cloud storage system is a cloud computing [2] system based on data storage and management. In the system data files have multiple replicas placed in different storage nodes, which improve the efficiency of data access and ensuring data reliability and availability and helps the cloud system be less vulnerable to data loss or other disasters. The replica placement and management technology is the most important problem in the cloud storage system[3].

Based on different constraints and optimization objectives, the researchers proposed some replica placement algorithms [4]. Through the replica technique, data center duplicates copies from the remote location to the local to reduce system's network bandwidth, shorten response time of the service request, And improve the performance of system data services. Some researchers consider the user's requirements for services performance by reducing access cost, which can directly reflect the performance improvement of the system.

In this paper, a data replica placement strategy based on load balancing is proposed. Through the process of load balancing in each data center, the data center's capital value is calculated by different parameters. The most appropriate replica placement location is based data center's capital value. The proposed strategy can adapt to the changes of the system and make reasonable choice of data center to place a replica, not only to meet the data availability and reliability, but also improve access efficiency and load balance between data centers to ensure the stability of the whole system. The results of the simulations can prove that the proposed replica placement policy can improve the system performance while ensuring the load balance between data centers. The contribution of this paper can be summarized as follows: (1) A replica placement algorithm is proposed. Choosing an appropriate location to place data replica and ensure the load balance of the data centers. (2) Using the variance to determine the degree of load balance between data centers. It is helpful to make comparison service performance of different data centers. (3) Through the process of selection, we make the normalization of the parameters of data centers. We define the concept of data center capital value as the evaluation of the merits of each data center.

In recent years, there have been a lot of researches on the replica placement in large scale cloud storage system. [6] proposes a strategy used in Wireless Multi-hop Mesh Networks (WMN) according to the size of the data object and the degree of popularity to adjust the number of copies in the system. Multiple Partitions per Delegate Node Assignment(MP_DNA) is proposed in [7]. It is defined as the NP complete problem for the optimization of P median problem. In the algorithm, router with limited storage capacity is regarded as a replica server. It places a replica in the location nearby users as far as possible, which reduce access delay and minimize the number of replicas. [5] proposes a replica selection and placement strategy, which includes two stages. The first stage uses the directory and index to determine the creation and placement of replicas, and the second stage determines whether the destination node has enough space to store a copy of the file. The strategy aims to improve the availability of resources in the process of copying data and reduce access costs and delay time. [8] proposes a strategy which uses Enhanced Interior Gateway Routing Protocol (EIGRP) to select transmission path of replica to ensure the reliability of data replicas, reduce the delay time and provide reliable data transmission policy.

Previous studies have promoted many replica placement strategies which take into account the optimization of the system performance from various aspects. There is not much studies related to load of the data centers. It is very rare to study the problem of load balancing of user's requests between data centers. This article emphatically starts from this aspect to do research.

The structure of this paper is as follows: The content of the first part has been described above. Section 2 presents the system model and the definition of the problem. The detailed steps of the replica placement algorithm are presented in the section 3. And we present experiment results in section 4. At last, conclusions and future work is shown in section 5.

II. Problem Model

In the cloud storage system, data replicas are distributed in different storage nodes to provide services for users. The target file that the users request for will change timely, users in remote place access the file will be waiting for a long time to get the response. In order to improve the service performance of the system and reduce the user's access waiting time, we will select the appropriate replica placement node and duplicate files place in the new storage node. How to select the appropriate location and ensure the load balancing of the data centers at the same time is the problem we will discuss.

TABLE I. *Notation Summary*

Symbol	*Meaning*
E	*Ideal load value*
D	*Number of data centers storage with data object n*
r_{cn}	*Client c request for replica n*
v	*Variance of load of data centers*
light	*Light load node*
heavy	*Heavy load node*
n	*A data replica*
$load_i$	*Load value of data center i*
$RD_{standard}$	*Standard distance for selection data center*
$dist_{jk}$	*Distance between replica j and k*
bw_i	*Bandwith of data center i*
f_i	*Request error rate of data center i*
RD_{lh}	*Distance of light and heavy node*
τ_i	*Average response time*

Data object O's replication set in the system is $S=\{R_1,R_2..R_j..R_k..R_m\}$, and each replica is placed in a data center. Each data center is only allowed to storage one replica of the same data file. R_j, R_k are the two replicas of data object O. We can calculate the standard distance as:

$$RD_{standard}=\frac{\sum_{j,k\in S} dist_{jk}}{C_m^2} \quad (1)$$

In the process of load balancing, if the distance between light nodes and heavy nodes $RD_{lh}>RD_{standard}$, it is considered that the cost of forwarding access request is relatively large, and it needs to select a data center node between light and heavy nodes as a new replication node. The new replica nodes can provide services to users and share the pressure of other data centers at the same time.While describing the whole process of selecting a new replica node and guaranteeing the data center load balancing we define some concepts as follows.

Definition 1: Load Balance Degree: The degree of load balance among the various data centers in the system. Expressed by variance of load values, the smaller v means the higher load balance degree.

Definition 2: light node and heavy node. A data center in which the number of requests for replica n is less than $1\text{-}\Delta_L(0\le\Delta_L<1)$. We define the node as light node. Similarly, if the number of requests for replica n is more than $1+\Delta_U(0\le\Delta_U<1)$, we defined the node as heavy node. Δ_L and Δ_U are the parameters for the system settings. E can be calculated as:

The load balance degree is calculated as :

$$E=\frac{\sum_c r_{cn}}{D} \quad (2)$$

$$\min \quad v=\sqrt{\frac{\sum_{i=1}^{D}(load_i-E)^2}{D}} \quad (3)$$

$$s.t.: \quad load_i\ge 0$$
$$D>0$$

Definition 3: Data center capital value P. Each data center competition for the new replica node, and P is directly comparison with other nodes. Capital value is calculated as:

$$P=\beta_1\cdot bw_i+\partial_1\cdot f_i+\partial_2\cdot load_i+\partial_3\cdot\tau_i \quad (4)$$

β_1, ∂_1, ∂_2 and ∂_3 are parameters for the system settings. Replica node require high network bandwidth, low failure rate, low load and low service response time. Hence we limit the range of parameters of $\beta_1>0$, $\partial_1<0$, $\partial_2<0$, $\partial_3<0$.

In the process of balancing the load of the data center, comparing the distance between light node and heavy node with the standard distance. If $RD_{lh}>RD_{standard}$ we need to select a new replica node. Alternative nodes send its own parameters information to the center nodes. Center node will choose the most appropriate node with the biggest capital value. Since then, the new replica node has been determined. It will serve as the new light node to receive forwarded access requests.

III. REPLICA PLACEMENT ALGORITHM

A. *Load balance Algorithm*

The datacenter manager (DM) records the load of each data center from the number of requests received from users. The ideal load value E can be obtained by calculation. Light and heavy nodes are selected from all the data centers. If $RD_{lh}<RD_{standard}$, heavy node will transmit one request to light node. If $RD_{lh}>RD_{standard}$, a new replica placement node needs to be selected between light and heavy node. The new data center will be identified as $light'$ node to receive request information from the heavy node, and we will not go to select a new replica placement node between light and heavy node in theory. So until the load balance of the system reaches the expected value. The core code of the algorithm is as follows:

Algorithm 1. Load Balance Algorithm

1. **while**(v>0.5) **do**
2. light node= minmum load[]
3. heavy node =maxmum load[]
4. **if**($RD_{lh} <= RD_{standard}$)
5.   heavy node transfer one request to light node
6.   update load[]
7.   light node= minmum load[]
8.   heavy node =maxmum load[]
9.   calculate v using formual (3)
10. **else if** ($RD_{lh} > RD_{standard}$)
11.   **break**;
12. **endif**
13.**endwhile**

B. Replica Placement Algorithm

Replica Placement Based on Load Balance (RPBLB), and the algorithm is based on the balance of the load of the whole system, and comparing the distance between light and heavy node to the standard distance to make decision that if it is necessary to select a new replica placement node. Then evaluate the value of alternative data center according to the parameters of data center.

In a large scale cloud storage system, data center nodes are heterogeneous. We use the collection $D=\{bw, f, load, \tau\}$ to represent the data center. When the system needs to select a new replica node, all the alternative nodes will send their parameters set D to the center node. The center node calculates the capital value of each data center according to the information received by formula (5):

$$P=\beta_1 \cdot bw_i + \partial_1 \cdot f_i + \partial_2 \cdot load_i + \partial_3 \cdot \tau_i \qquad (5)$$

The new replication node is used as $light'$ node to balance load of the heavy node. The core code of the algorithm is as follows:

Algorithm 2. RPBLB Algorithm

1.Collect system information $\{D, load, dist, bw, f\}$
2. Initialize load[] with the collected load
3. Calculate current v using formula (3)
4. Calculate $RD_{stantard}$ using formula(1)
5. **while**(v>0.5) **do**
6. load balance algorithm
7. **for each** alternative node
8. send $D_i=\{bw_i, f_i, load_i, \tau_i\}$ to center node
9. calculate P using formula (4)
10. **endfor**
11. best datacenter= maxmun P node
12. light' node = new node
13. **endwhile**

IV. PERFORMANCE EVALUATION

The performance of the proposed approach RPBLB will be evaluated in this section. Our simulation platform consists of a master data center node and multiple slave data center nodes. In the master data center, a data center manager is set up to collect some information from other data center nodes, including routing information, access times, and so on.

We use the Cloudsim[9] simulation platform to verify the RPBLB strategy, in order to reflect the superiority of our approach. It will be compared with the other two strategies. One strategy is implementing the Hadoop Distribute Replica Placement (HDRP) method based on the default method of the Hadoop distributed file system. Another method is Cost-effective Dynamic Replica Placement (CDRP)[10]. In this paper, the results of these two methods are compared with our method.

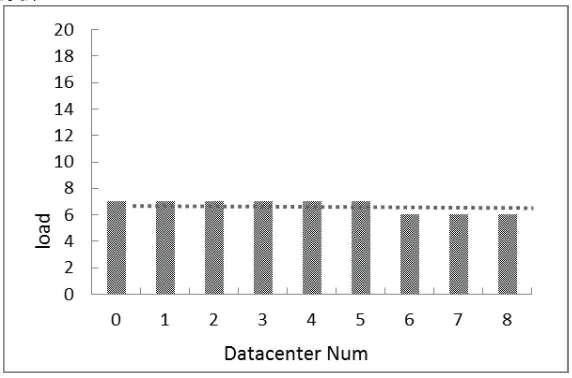

Fig. 1. Result of Load Balance

After several rounds of light and heavy nodes, the load balance degree of system has reached the most optimal state at current time, and the variance of V is the best. At this point, the system judges the process of load balance is finished. As shown in Figure 1, three new replica placement node have been created. The load value between each datacenters is almost balance. In this case, system can respond to users' requests better and quickly.

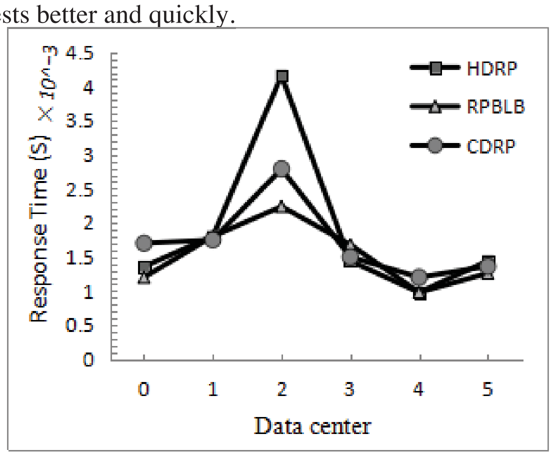

Fig. 2. Response Time Comparison

Figure 2 presents a comparison of the response time of each datacenter between three methods. As shown in the figure, Datacenter_2's service response time is the longest because of the largest load value. In HDRP algorithm, the response time of Datacenter_2 is the highest, because the number of tasks allocated to it is the largest, and other datacenters receive fewer service requests, so bring out maximum service response time. The service response time between data centers in the HDRP is relatively asymmetrical. CDRP's performance is better than HDRP's. The response time of each data center is relatively uniform. Compared with RPBLB, CDRP is relatively not good. Figure 8 proves that RPBLB is much better performance in response time at the same condition.

Figure 3 shows the comparison of the overall service response time of three methods in the case of different load levels. As shown in the graph, the service response time of HDRP is always higher than the other two algorithms due to the uneven distribution of service request information in the data center. Before the system load level reached 40%, the CDRP algorithm has a better performance than RPBLB. When the system load level is more than 50%, the overall service response time of RPBLB has a significantly smaller trend. Because the RPBLB algorithm will select new replica placement nodes in the implementation process, so as to deal with the task more quickly. Hence the response time of the task will be getting less. With the task quantity increasing, CDRP's overall response time is getting more excellent at this stage, but with the workload continues to increase, the overall response time showed a linear increasing. RPBLB and CDRP are almost equally in response time when load is fully loaded.

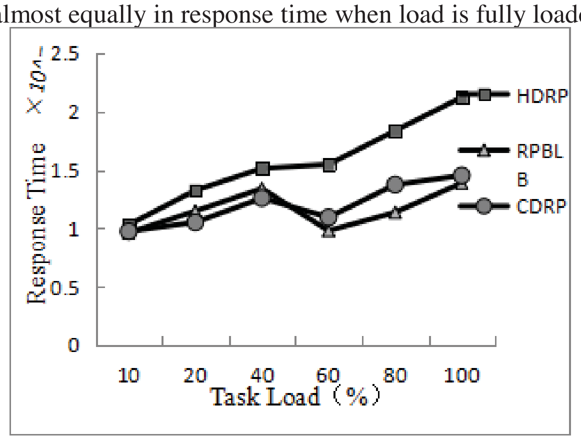

Fig. 3. Response time comparison under different load levels

In this paper, RPBLB is proposed to determine whether the actual distance between the two nodes for forwarding requests is required to select an appropriate data centers to be a new replica placement node. Figure 4 presents the number of created replicas nodes is different for different scales of system. From the graph we can see that with the increasing of data center scales, the number of newly selected replica nodes is linear growth.

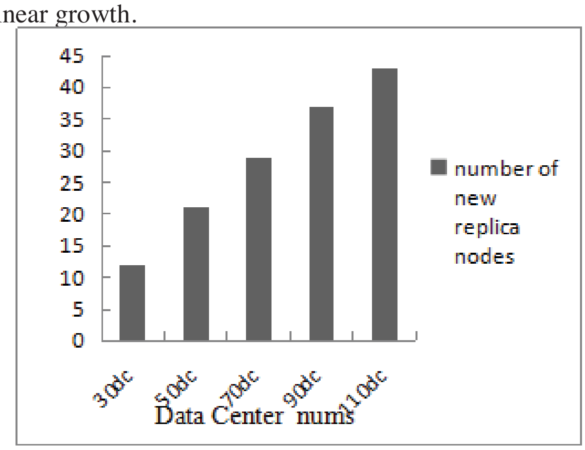

Fig. 4. Number of created replica nodes under different scales

Simulation experiment results show that RPBLB can achieve higher level in load balance degree, and service response time in various conditions is also performance good. RPBLB can satisfy user's requests and achieve load balance between datacenters at the same time. Selecting appropriate replica placement nodes to improve the service performance level of the system is also able to realize

V. CONCLUSION

The RPBLB algorithm proposed in this paper is based on the load of all data centers in the whole system, and according to the distance between two nodes for forwarding service request information, the new data center nodes are determined. Then the parameters of the alternative data centers will be sent to the center for data center data. The center node will calculate capital value of each datacenter, so as to select data center with the highest capital value as a new replica placement node. However, there are some shortcomings in RPBLB. In theory, RPBLB will create many new replica nodes. The cost of management will increase rapidly. In response to a variety of service request pattern also need to verify. In the future, we will study the problem of replica management in data center from the perspective of energy consumption to improve RPBLB.

ACKNOWLEDGMENT

This work is sponsored by Primary Research & Development Plan (Social Development) of Jiangsu Province (BE2017743).

REFERENCES

[1] Ju J, Wu J, Fu J, et al. A survey on cloud storage[J]. Journal of Computers, 2011, 6(8): 1764-1771.

[2] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50-58.

[3] Long S Q, Zhao Y L, Chen W. MORM: A Multi-objective Optimized Replication Management strategy for cloud storage cluster[J]. Journal of Systems Architecture, 2014, 60(2): 234-244.

[4] Shang P, Wang J, Zhu H, et al. A new placement-ideal layout for multiway replication storage system[J]. IEEE Transactions on Computers, 2011, 60(8): 1142-1156.

[5] Rajalakshmi A, Vijayakumar D, Srinivasagan K G. An improved dynamic data replica selection and placement in cloud[C]//proceedings of International Conference on Recent Trends in Information Technology (ICRTIT), 2014: 1-6.

[6] Al-Arnaout Z, Fu Q, Frean M. A content replication scheme for wireless mesh networks[C]//Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video. 2012: 39-44.

[7] Al-Arnaout Z, Hart J, Fu Q, et al. MP-DNA: A novel distributed replica placement heuristic for WMNs[C]//Proceedings of IEEE 37th Conference on Local Computer Networks (LCN), 2012: 593-600.

[8] Waqas A, Majeed N, et al. Dynamic object replica placement using underlying routing protocols: Ensuring the reliability[C]//Proceedings of the International Conference on Information and Communication Technology for the Muslim World 2013: 1-4.

[9] Calheiros R N, Ranjan R, Beloglazov A, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software: Practice and Experience, 2011, 41(1): 23-50.

[10] Wei Q, Veeravalli B, Gong B, et al. CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster[C]//Pproceedings of IEEE International Conference on Cluster Computing (CLUSTER). 2010: 188-196.