

Research of Replica Replacement Strategy Oriented Cloud Storage

Genqing Bian, Rong Li, PeiJiao Gong

College of information and control engineering

Xi'an University of Architecture and Technology

Xi'an China

bgq_00@163.com, huanglongzhijiao@126.com, gudengxuran@126.com

Abstract—The long latency on reading the files from remote node of cloud storage leads to degrade the whole system service performance and waste the network resource. In order to improve the speed of data access, replica management ability and optimize the whole system performance, the dissertation proposes a replica replacement strategy based on gravitational search model, which can achieve the goal by removing unnecessary replica, inactive replica and updating replica. The simulation results show that the presented method is effective and reasonable.

Keywords—Cloud Storage; Universal Gravitation; Replica Replacement; Replica Management

I .INTRODUCTION

Low data access ^[1] latency becomes the primary measure of system performance when the cloud storage is widely used in all kinds of data intensive industries ^[2], the user and data are growing rapidly. Replica replacement technology can not only effectively reduce the delay time of accessing the file, but also make the local client to access the data file quickly.

Replica management mainly involves the creation, choice, replication, consistency and so on. A reasonable replica management can improve the ability of system processing data ^[3] greatly. Increasing the number of replica can quicken the response that the client access the data file, but the growing ^[4] replica will occupy too much storage space. Therefore, replica replacement technology ^{[5] [6]} can be used to delete some unnecessary and inactive replica.

At present, most of the replica replacement algorithms ^[7] are from traditional page replacement

algorithm based on operating system, which achieved certain effect. But these algorithms have some limitations due to the different application platform between operating system and cloud storage. In order to balance system load effectively, reduce the consumption of network bandwidth and the latency of data access, and realize the service performance of cloud storage, this paper explores a replica deletion strategy based on gravity model.

II. THE DESCRIPTION OF UNIVERSAL GRAVITATION

Gravity is one of the four basic forces in nature, particles close to each other constantly under the action of gravity. In nature, every particle attracts other particles constantly. For the separate particles, gravity among particles has no interval and delay.

The mathematic expression of the universal gravitation is described as follows:

$$F = G \frac{M_1 M_2}{R^2} \quad (1)$$

Among them, the F said the size of two particles' gravity; G is the universal gravitational constant; M_1 , M_2 stand for inertial mass of two particles, R represents the Euclidean distance between two particles. The formula (1) is suitable for calculating the gravitation between two particles; the universal gravitation calculates each particle's gravity by dividing non-particle into thousands of particle.

Esmat Rashedi and H.N etc. put forward gravitational search algorithm in 2009, which is similar to particle

swarm optimization (PSO), all of them are heuristic algorithm. In order to save network resources, the algorithm explore optimization search by the swarm coming from gravity interaction.

III. RPLICA REPLACEMENT STRATEGY BASED ON GRAVITY

A. Model Architecturing

The strategy establishes a gravity model between files and saved-nodes according to the related parameters in the evaluation of replica value. It is easy to replace the replica when the attraction is smaller. Finally the system's overall efficiency will be improved until the strategy replaces all the lowest value of replica completely.

The replica replacement strategy based on gravity is described as follows:

$$F_i(f) = G_0 \frac{N_i(t, T) \times S_i}{B_i} \quad (2)$$

By (2), F expresses the value of replica equaled to the size of gravity, $F_i(f)$ presents the value of the i th replica in file f , $N_i(t, T)$ stands for the t times that the i th replica was visited in the time T , B_i presents the bandwidth between file f and the i th replica node, S_i said the size of the replica i , $G_0 = 0.02$ is the gravitational constant.

B. Algorithm analysis and design

Formula (2) shows that, if the frequency of the replica being accessed is high, and its file capacity is larger, but the bandwidth between file and the replica node is narrow, then the replica's attraction is larger and the replica's probability of being replaced is lower; reversely, the probability is higher.

The cost of the ideal replica contains access cost and transport cost. If the selected replica is not in local, user need to visit other nodes, which will increase the visiting time. In addition, if the capacity of replica is bigger, the transmission time is correspondingly longer. The operation time of the whole system will also increase when the file access time increases. Operation time can be expressed as follows:

$$T_{ft} = T_{vt} + T_{pt} + T_{wt} \quad (3)$$

Formula (3), T_{ft} represents operation time, T_{vt} said the file access time, T_{pt} stands for documents processing time, T_{wt} said the queue time. The paper builds the model by considering the size of the replica, access frequency, and bandwidth etc.

The dissertation designs the replacement algorithm steps:

- 1) Obtaining the record of the visited replica in the time t , storing the visiting records in a queue;
- 2) Finding all nodes that contain the replica, then getting the available max bandwidth between request node and replica node;
- 3) Calculating the size and the times of the replica;
- 4) Ordering these replicas by the value;
- 5) Replacing the replica according to the remaining storage space, which includes replica's copying and update.

Updating replica refers to compare the capacity of new replica and the sum of remaining space and old replica; Updating the replica when the latter is greater than the former; If the value of the replica is the smallest, the algorithm will delete it directly when the later is less than the former; If the value is not the smallest, the algorithm deletes the replica that is older and lowest access frequency until the node has a space to place the new replica. Copying replica refers to compare the remaining space on the storage nodes and the capacity of the replica, the algorithm will copy the replica when the former is greater than the latter; conversely, the algorithm will delete the replica, which is lowest access frequency until the node has space to copy the replica.

Replacement algorithm's flow chart is shown in figure 1.

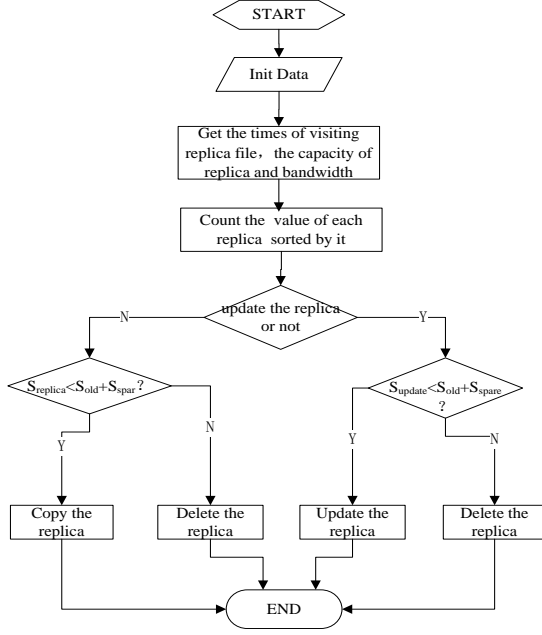


Figure 1 flow chart of the replacement algorithm

Algorithm process:

GetLogFile (); // getting access to records of the copies in a time t;

GetAvailableBandwidth (); // getting the maximum available bandwidth between the replica and other replicas' nodes;

GetSize (); // getting capacity of the replica;

CountValue (); // calculating the value of the replica;

Sort (); // ordering the replica by value;

if (need to update replica) {

if($S_{\text{replica}} < S_{\text{old}} + S_{\text{spare}}$)

Replace (); //the new replica replace the old replica;

else {

if(the value of replica is smaller)

FileDelete (); // deleting replica;

else

FileDelete (); // deleting the replica that is lowest value until there is space to copy new replica;

Replace (); // updating the replica;

}

}

else (copy replica) {

if($S_{\text{replica}} < S_{\text{old}} + S_{\text{spare}}$)

AddFile (); // copying the replica;

else {

FileDelete (); // deleting the replica its value is lowest

until there is space to copy new replica;

AddFile (); // copying the replica;

}

}

IV. SIMULATION AND ANALYSIS

Simulation experiment is OptorSim simulator developed by nuclear physics research center. The simulator studies performance about the optimization algorithm in the specific environment. The simulator simulates the replacement algorithm based on gravity model (GMRA) need to extend OptorSim. OptorSim replaces the replica by using interface optimisable in the optor package. The interface encapsulates two classes: `getAccessCost()` and `getBestFile()`; The method of `getAccessCost()` is used to calculate the cost of unit visit the file, then the simulator selects the minimum cost, namely the minimum access time. `getbestfile()` is the method being used to create a file copy in the right time, then choose the most suitable replica.

This experiment compared GMRA algorithm strategy with several simulator's classic algorithms. Operation numbers of the experiment are 200, 500, 1000 and 2000 respectively. The simulation experiment has evaluated the performance of the algorithm by the average operation time and network utilization ratio^[8]. This study set the parameters is shown in table 1:

Table1 parameter settings about OptorSim' s practical platform

Parameter Names	GMRA	LRU	MRU	LFU	MFU
Grid ConfigurationFile	examples/cms_tested_grid.conf				
Job ConfigurationFile	examples/cms_tested_jobs.conf				
Number of Jobs	200、500、1000、2000				
Optimiser	2	8	7	3	6
Access Pattern	Sequential				
InitialFile DistriBution	14				
Job Delay	5000(ms)				
Max Queue Size	200				
User	simple				

The dissertation compared the average time and network utilization on several kinds of replacement algorithm. The average operation time is shown in figure 2, the network utilization is shown in figure 3.

1) The average operation time

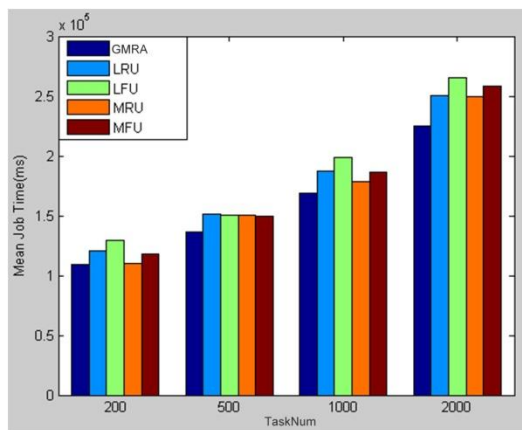


Figure 2. the comparison of average operation time

2) Network Utilization

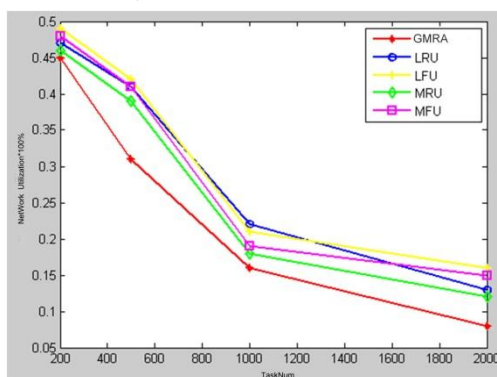


Figure 3. the comparison of network utilization

The replica replacement strategy based on gravity (GMRA) proposed in the paper is better than other several classical algorithms in different conditions. The GMRA spends less time than classical algorithm when there is less operation number, which has certain advantages. The GMRA algorithm is obviously super when there is more operation number. The GMRA proposed in this dissertation is superior to several classical algorithms in network utilization.

V. CONCLUSION

This strategy improves the speed of data access and management ability by replacing replica. The simulation results show that the strategy (GMRA) is better than other several classical algorithms in average operation time and network resource utilization.

Acknowledgment

This research is supported by National Natural

REFERENCES

- [1] Liu Yang, Zhuo Tang, Renfa Li, Zongli Zhang. The lookup algorithm based on user access requirements in cloud computing environment[J]. Journal of communication Journal, 2011, 32 (7) : 169-175.
- [2] Malewicz, et al. Pregel: A system for Large-Scale Graph Processing[C].//PODC, 2009.
- [3] Linlin Ding, Junchang Xin, Guoren Wang, etc. The Huge Amounts of Data Skyline query processing based on the Map - Reduce [J]. Journal of computers, 2010, (10) : 1785-1796
- [4] Lin Cai. the large-scale distributed storage system research's implementation and management [D]. Chengdu: university of electronic science and technology, 2008.
- [5] Dongsheng Li, Chunjiang Li, Nong Xiao. Adynamic adaptive locating method [J]. Journal of computer research and development, 2003 (12) 1775-1780:
- [6] Qingsong Wei, Xianliang Lu, Mengshu Hou, AdpReplica: adaptive replica management mechanism [J]. Journal of computer science, 2004, 31 (12) :34-51.
- [7] Xiaodong wang. A computer algorithm design and analysis. China: electronic industry press. 2007:34-89
- [8] David G. Cameron, Ruben Carvajal-Schiaffino, Jamie Ferguson, et al. OptorSim v2.1 Installation and User Guide, 2006