International Journal of Grid and Distributed Computing
Vol. 9, No. 6 (2016), pp.269-278
http://dx.doi.org/10.14257/ijgdc.2016.9.6.25

Pheromone-Based Genetic Algorithm Adaptive Selection Algorithm in Cloud Storage

TIAN Junfeng[1], LI Weiping[2]

*[1]School of Computer Science and Technology,
HeBei University, Baoding071002, China
[2]School of Management, Hebei University,
Baoding071002, China*

Abstract

Aiming at the problem of replica selection optimization in cloud storage load balancing technology, a new dynamic selection algorithm based on genetic algorithm(GASA in short) is proposed. According to the principle of genetic algorithm, the model of dynamic selection strategy based on genetic algorithm is constructed, and then the key steps of the replica selection criteria and genetic algorithm are mapped, and then the optimal solution is obtained by using the probability equation. Lastly. simulation results from cloud test bed. which is based on Optorsim. show that GASA can reduce data access latency and bandwidth consumption. and effectively achieve cloud load balancing between storage nodes and improve the speed of data access.

***Keywords**: cloud storage; replica selection; genetic algorithm; load balancing Optorsim*

1. Introduction

With the continuously rapid development of internet technology, explosively growing information follows up with it. In addition, order of magnitudes for amount of information acquired by end users is gradually tending to quantization. How to acquire required information through a simple path or method from massive amounts of information for end users has become a realistic question that has troubled end users for a long time. The occurrence and constant popularity of cloud computing make cloud storage become a resource information sharing technology with seamless integration[1]. In the process of conducting cloud storage, cloud computing center needs to copy massive data resource information into multiple physical devices of cloud storage and provides related services in line with virtual hosts. Because massive data resources are very enormous, the position of virtual hosts is ever-changing. Furthermore, network loading is equipped with uncertainty and difference of physical server processing capacity, so that virtual resource information loading is unbalanced [2, 3, 4]. Therefore, in the process of cloud storage resource allocation, how to select reasonable replica selective and scheduling algorithm is the key and core of the entire cloud network resource allocation[5].

Replica selective and scheduling algorithm of cloud storage load balance is a very important algorithm. At present, many scholars and experts at home and abroad have studied it, which becomes a research hotspot and is a process that mainly selects a reasonable and available replica in the distributed cloud computing environment [6]. In the process of selecting replica, it depends on a great number of factors, primarily including the position of placing replica, data size of replica, network bandwidth and delay, network state between virtual hosts and servers, node load condition of placing replica, and I/O reading rate of disk itself, *etc*[7]. Sari *et al*[8] designed a set of game strategies aiming at transcript action to obtain the optimized combination including participators of replicaand client. This is called as the Nash Equilibrium. Being specific to

ISSN: 2005-4262 IJGDC

network load change, Govinda *et al* [9] designed a set of data transmission regression forecast model based on GridFTP, but this model needs to collect a mass of historical data. Ye *et al*[10] analyzed relevant factors of impacting replica selection, combined with grey system correlation theory to predict corresponding time of replica and established a set of GM(1, 1) gray level dynamic fitting model. Mostafa *et al* [11] utilized modern integrated intelligence algorithm to select major factors of impacting replica selection and designed a replica selective strategy based on ant colony algorithm. Genetic algorithm[12,13] has positive feedback and cooperativity, so that it can be better applied and distributed in storage system. Moreover, expandability equipped makes it better adapt to network structure and replica dynamics, so as to change cloud storage environment.

Aiming at problems faced by the replica selection strategy, this paper proposes a Pheromone-based genetic algorithm adaptive selection algorithm in cloud storage (for short GASA). The algorithm makes use of replica genetic algorithm to carry out dynamic interaction between pheromone and cloud storage environment, adjusts replica selection strategy by virtue of feedback information, conducts overlapping mutation constantly, and ultimately obtains optimal solution of replica selection. This algorithm selects the most suitable replica to place in accordance with replica node load data, so as to realize replica distribution and load balance processing of virtual machine cluster.

2. Genetic Algorithm

Genetic Algorithm simulates biotic population evolutionary mechanism in natural environment, including multiply, overlapping and mutation, *etc*. Looking for optimal solution by using this mechanism is a self-adapting optimal probabilistic algorithm[14][15]. The specific flow chart of optimization problem algorithm based on genetic algorithm is shown in Figure 1:

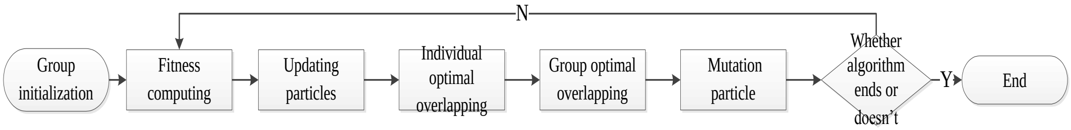
N
Group initialization
Fitness computing
Updating particles
Individual optimal overlapping
Group optimal overlapping
Mutation particle
Whether algorithm ends or doesn't
Y
End

Figure 1. Flow Chart of Genetic Algorithm

Here group initialization initializes particle swarm population for every particle in the entire group. Fitness computing module calculates individual fitness value through preset fitness computational formula. Updating module updates individual optimal particles and group optimal particles in line with fitness value. Individual optimal overlapping intersects individuals and individual optimal particles to obtain a new particle; Group optimal overlapping intersects individuals and group optimal particles to obtain a new particle; Particle mutation refers to the new particle through particle mutation.

Specific steps of algorithm[16], first of all, initialize particles, set up the corresponding fitness function, calculate particles through the fitness function, update particles constantly; conduct individual optimal overlapping and whole optimal overlapping on particles, obtain mutation particles, and judge whether the algorithm ends or doesn’t. If it doesn’t end, it will continue to repeat the above-mentioned process until the algorithm finds out optimal solution or the algorithm reaches the end of iterations.

The GASA algorithm proposed in this paper obtains replica probabilistic information in cloud storage environment by virtue of group constant overlapping, realization of mutation mechanism and environmental dynamic interaction and ultimately gains optimal replica. Applying genetic algorithm to replica dynamic selection process mainly includes advantages in two aspects:(1) Genetic algorithm ultimately can achieve the optimal solution by using this mechanism constantly in the process of overlapping and mutation. (2) It is a distributed optimization algorithm that can adapt to distributed environment and adopts an overlapping mutation strategy to enlighten optimization for the optimal solution.

3. Dynamic Prediction Selection Algorithm of Cloud Storage

Here GASA algorithm will be introduced in details from chromosome coding, pheromone updating and algorithm flow, *etc*., aspects.

3.1. Chromosome Coding

Aiming at the transcript selection optimization problem in cloud storage, the algorithm selects binary system to code[17][18]˙ The specific rules of coding mean that once users select to use data resource information of this replica, the corresponding value will be set up as 1. Otherwise, it will be set up as 0. Binary coding rule can greatly reduce and shrink searching space and is beneficial to speed up entire optimal rate of the algorithm. In the process of chromosome coding, nodes of storage position for every replica in cloud storage environment will request to send and predict replica price in line with different users and compare with reservation price of corresponding users. Once predicted replica price is lower than user reservation price, this predicted replica price should be retained.

3.2. Calculation of Fitness Function

m replicas of a data resource in cloud storage environment adopt set *Replica* to express. The specific formula is as follows:

$$\mathrm{Re}\,plica = \{r_1, r_2, r_3, ..., r_m\} \tag{1}$$

In the storage environment, *n* users select costs required by *m* resources to adopt a set X_{ij}, n*m matrix to express:

$$X_{ij} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{1m} \\ x_{21} & x_{22} & x_{23} & x_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & x_{n3} & x_{n4} \end{bmatrix} \tag{2}$$

In the Formula (2), x_{n1}, x_{n2}, ..., x_{nm} refers to resource cost required by replica, which is selected by nth user.

It is necessary to adopt set Bandwidth to refer to corresponding network bandwidth situation of *m* nodes in the data transmission path. The specific situation is shown as follows:

$$Bandwidth = \{b_1, b_2, ..., b_m\} \tag{3}$$

The vector V is applied to express whether jth node to acquire replica resource information situation. The specific situation is shown as follows:

$$V = \{v_1, v_2, ..., v_m\} \tag{4}$$

In the Formula (4), *j*=1, 2, 3,…, m. As v_j=1, it refers to acquire replica resource information from the jth node. Otherwise, it means that it doesn’t acquire resource information from the jth node.

In the initialization phase of the algorithm, assuming that predicted price of all replicas is lower than reservation price of users. Initialized chromosome coding is a random 0-1 coding sequence. Any replica on the node should satisfy two constraint conditions: cost required by acquiring replica resources is the minimum. Bandwidth allocated by replica resources is the maximum. The specific expression is as follows:

$$\min f_1(v) = \sum_{j=1}^{m} v_j X_{ij} \qquad (5)$$

$$\max f_2(v) = \sum_{j=1}^{m} v_j b_j \qquad (6)$$

In the Formula (6), in order to better analyze influences of related parameters on target value, the formula is translated into:

$$\min f_2'(v) = P - \sum_{j=1}^{m} v_j b_j \qquad (7)$$

In the Formula (7), P is a very large constant, P>0. Therefore, strategy selection fitness function of replica can be expressed as:

$$\min F = \alpha f_1(v) + (1-\alpha) f_2{}'(v) \qquad (8)$$

Aiming at the Formula(8), α is weight factor. If α is larger, it indicates that this user requires smaller cost for acquiring replica resource expenses. Otherwise, if 1-α is larger, it means that the user requires for larger network bandwidth in cloud storage environment. There are two different emphases.

3.3. Pheromone Generation

Replica selection strategy fitness function designed is regard as initial pheromone distribution correlation situation. When t=0, generation and computational formula of pheromone are as follows:

$$\tau_j(0) = \alpha f_1(v) + (1-\alpha) f_2{}'(v) \qquad (9)$$

In the following time, pheromone will update constantly. The specific form of updating is as follows:

$$\tau_j(t+1) = \rho \tau_j(t) + \Delta\tau_j \qquad (10)$$

In the Formula (10), variable P in the formula refers to residual coefficient of pheromone, which is set up as 0.8 in line with empirical value in the experimental process. $\Delta\tau_j$ means that with the continuous renewal of selection replica resource information and actual variation of pheromone, computational formula can be divided into three situations in line with actual situation:

(1) When user adopt the way of remote access to access replica information resource, it is necessary to reduce pheromone concentration of this replica. The main reason is because if pheromone concentration of the replica is recued, it can avoid other users from continuing to access this replica and causing network congestion situation, so as to reduce network bandwidth loading. The specific computational formula of pheromone is as follows:

$$\Delta\tau_j = -\frac{Filesize}{Brandwidth} \qquad (11)$$

(2) Once users select a replica information resource and access successfully, pheromone concentration of the entire replica information resource should be adjusted immediately. The specific computational formula is as follows:

$$\Delta\tau_j = -C_e \frac{Filesize}{Brandwidth} \tag{12}$$

In the Formula (12), C_e refers to incentive factor after accessing a certain replica resource successfully and is set up as 0.8 in line with actual empirical value.

(3) Once users select a certain replica information resource under the situation of accessing unsuccessfully, pheromone concentration of replica resource information should be adjusted immediately. The specific computational formula is as follows:

$$\Delta\tau_j = -C_p \frac{Filesize}{Brandwidth} \tag{13}$$

In the Formula (13), C_p refers to the penalty factor after accessing a certain replica resource successfully and is set up as 1.1 in line with actual empirical value.

3.4. Setting and Selection Strategy

As pheromone concentration of replica resources change, selective probability for users to select this replica also should be changed correspondingly. Probabilistic selective formula of the replica has the following specific calculation:

$$P_j^k = \begin{cases} \dfrac{\left[\tau_j(t)\right]^\alpha \left[\eta_j\right]^\beta}{\sum_{u=1}^{n}\left[\tau_u(t)\right]^\alpha \left[\eta_u\right]\beta} \\ 0, other \end{cases} \tag{14}$$

In the Formula (14), j and u are attained data replicas. η_j refers to pheromone concentration of replica j at the initialization time. $\tau_j(t)$ refers to pheromone concentration of replica j at t. α and β are denoted as weight factors and are set up as 0.5 in actual experimental process. Two weight values mean that pheromone concentration in replica selective process is determined by initial pheromone and current pheromone together.

3.5. Algorithm Flow

Based on the above-mentioned formula and basic idea of the algorithm, the specific process of replica selection algorithm based on genetic algorithm is shown as follows:

Step 1: Initialize the group: adopt binary coding rules to code chromosome at random and form an initialized group;

Step 2: Calculate in line with fitness function: According to network condition in cloud storage environment, replica service condition and transmission network bandwidth, *etc*., consumption situation, calculate fitness in accordance with fitness function;

Step 3: Initialize pheromone: According to the fitness formula in Step (2), calculate initial pheromone concentration of all replicas;

Step 4: Calculate selective probability of replica resources: By setting up selective probability formula of replica resources, calculate selective probability of every replica in cloud storage environment;

Step 5: Update pheromone concentration: By setting up pheromone updating formula, conduct update processing on pheromone concentration of every replica in replica resources;

Step 6: Judge ultimate terminal condition: According to setting threshold value, judge whether the algorithm satisfies ultimate iteration conditions. Once it satisfies, the algorithm will be calculated. Otherwise, it shifts to Step 4.

Table 1. The Configuration Situation of Sites

Nodes	CE	SE	Connection Status									
1	5	1	0	0	1000	0	0	0	0	0	0	0
2	1	1	0	0	1000	0	0	0	0	0	0	0
3	3	1	1000	1000	0	1000	1000	0	0	0	0	0
4	5	1	0	0	1000	0	0	0	1000		0	0
5	0	1	0	0	0	0	0	0	0	1000	0	0
6	2	1	0	0	0	1000	0	0	0	1000	0	0
7	0	1	0	0	0	0	0	1000	1000	0	1000	1000
8	0	1	0	0	0	0	0	0	0	1000	0	0
9	1	1	0	0	0	0	0	0	0	1000	0	0

4. Algorithm Simulation Experiment and Result Analysis

4.1. Experimental Environment

This paper mainly adopts Optorsim simulator to conduct simulation experiment on replica dynamic selective mechanism of cloud storage based on genetic algorithm and verifies experimental results by comparing and analyzing experiment. In the experiment, network topological graph adopted is shown in Figure 2. Grid in the figure refers to gridding sites. There are a total of 10 sites. The configuration situation of sites is shown in Table 1.

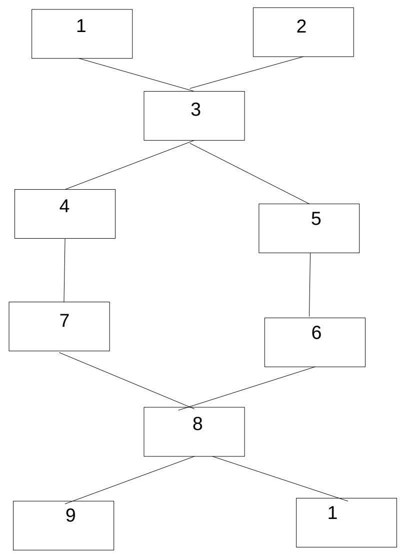

Figure 2. Network Topological Graph

In the experiment simulation process, request replica is primarily distributed in cloud storage environment at random and comes from different clients to execute in the parallel manner. In the accessing process, the same replica at a moment only can accept on client request. If the situation that multiple clients request simultaneously appears, some clients have to wait until this replica can access.

4.2. Analysis of Experimental Results

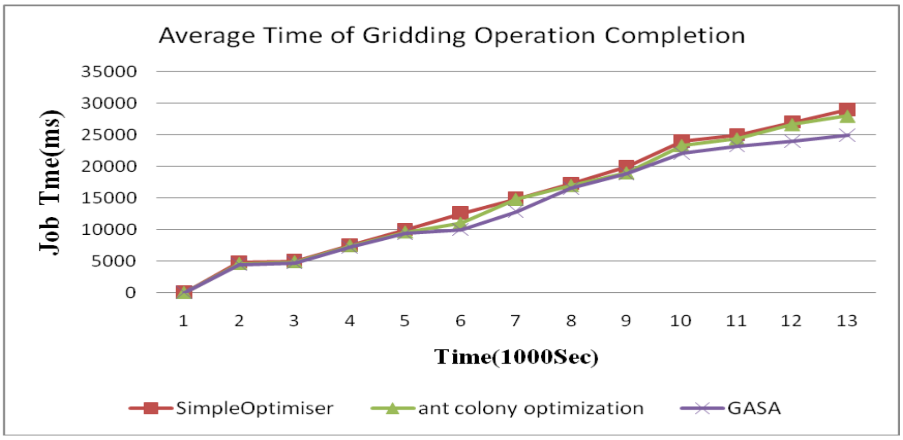

Figure 3. Average Time of Gridding Operation Completion

In the Table 1, CE represents the number ofcomputing elements in every site. SE refers to the number of storage elements in every node. The remaining one presents linkage information and bandwidth information between mutual nodes. 0 means there is no connection. In the experimental process, first of all, the number of operation is set up as 10. Average time of operation through the simulation experiment is shown in Figure 3. The average time of operation at Site 7 is shown in Figure 4.

It can be observed from Figure 3 and Figure 4 that in the situation of setting up operation number as 10, compared with two other strategies, the replica selection strategy based on genetic algorithm has reduced operation time in the entire cloud storage environment and single gridding site environment and can improve access speed of users to a certain extent. However, the improved effect is not obvious.

Based on such a situation, the operation number is adjusted 50 and 100 from 10 to carry out simulation experiment. The experimental results are shown in Figure 5.

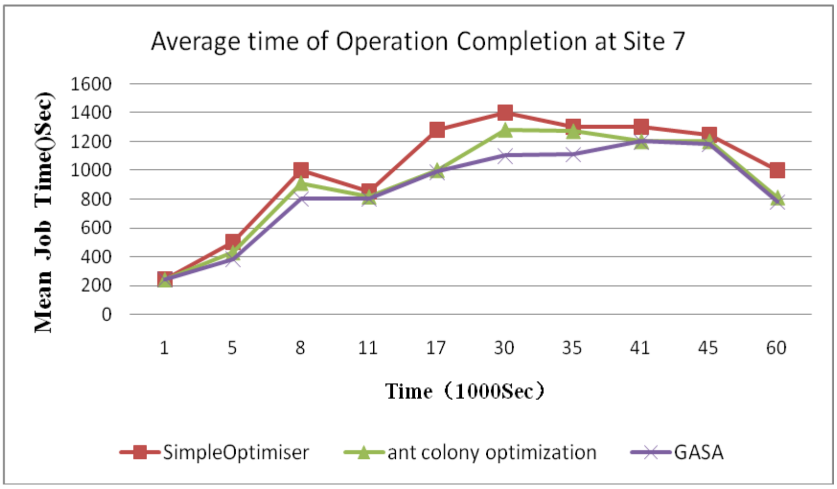

Figure 4. Average Time of Operation Completion at Site 7

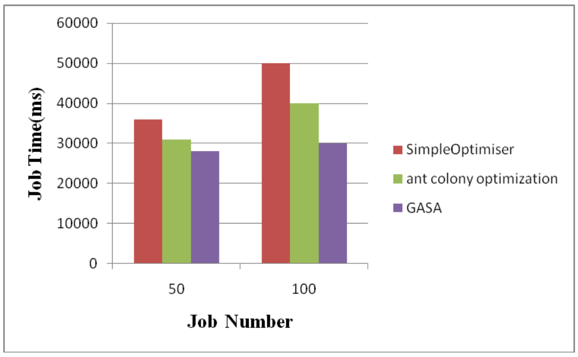

Figure 5. Average Time of Operation Completion

It is obviously noted from Figure 5 that with the constant increase of operation number, the replica selection strategy of cloud storage environment proposed in this paper and based on genetic algorithm can improve average time of operation completion in the entire data environment significantly and enhance entire gridding performance.

5. Conclusions

Aiming the replica optimization problem in cloud storage environment of massive data and on the basis of analyzing replica selection problems, this paper come up with a replica dynamic update selection algorithm based on genetic principle. Experimental simulation indicates that this algorithm has better performance for cloud storage of massive data, reduce operation time of replica, reduce average delay of network and realize the load balance between nodes effectively.

Acknowledgements

This work is supported by the Natural Science Foundation of Hebei province (F2016201244) , the Natural Science Foundation of Hebei institution (ZD2015088)and the National Natural Science Foundation of China (Grant No. 61379116).

References

[1] D. Catrein, and C. QSC AG,J. Maintaining User Control While Storing and Processing Sensor Data in the Cloud[J]. International Journal of Grid & High Performance Computing. 5,4 **(2013)**

[2] D. Q. FENG, J. FENG, Z. X. TANG, and C. WANG, J. Hydraulic Hybrid Cloud Storage Platform Oriented to Share and Exchange. Computer and Modernization.12 **(2014)**

[3] W. J.MA, K. RUAN, J. FENG, and Y. SHEN, J. Optimizing Algorithm for I/O Performance of Cloud Storage System Based on Dynamic Hybrid Range Mechanism. Journal of Beijing University of Technology, 5 **(2013)**

[4] G. S. Park and H. Song, J. A novel hybrid P2P and cloud storage system for retrievability and privacy enhancement. Peer-to-Peer Networking and Applications,1**(2015)**

[5] G. J. V' Noordende, S. D. Olabarriaga, M. R. Koot, and C. T. A de Laat. J. A Trusted Data Storage Infrastructure for Grid-Based Medical Applications. International Journal of Grid & High Performance Computing, 1,2 **(2009)**

[6] Y. S. Dong, G. C. Xu, and X. D. Fu, J. A distributed parallel genetic algorithm of placement strategy for virtual machines deployment on cloud platform.. Scientific World Journal.2014 **(2014)**

[7] E. Yuan, L. I. Fei, J. Tang, and B. Zhao, J. Research on Task Schedule Algorithm of Cloud Storage Based on Improved Ant Colony Algorithm. Journal of Sichuan University of Science & Engineering,1 **(2014)**

[8] A. Sari, J. A Review of Anomaly Detection Systems in Cloud Networks and Survey of Cloud Security Measures in Cloud Storage Applications. Journal of Information Security.6,012 **(2015)**

[9] K. Govinda and E. Sathiyamoorthy, J. Privacy Preservation of a Group and Secure Data Storage in Cloud Environment[J]. Cybernetics & Information Technologies.15 **(2015)**

[10] Z. Ye, S. Li, and J.Zhou , J. A Two-layer Geo-cloud based Dynamic Replica Creation Strategy[J]. Applied Mathematics & Informationences.8,1**(2014)**

[11] N. Mostafa, I. Al Ridhawi,and A. Hamza, Editors. An intelligent dynamic replica selection model within grid systems// GCC Conference and Exhibition (GCCCE), 2015 IEEE 8thIEEE **(2015)**

[12] J. J. Shen, H. Jiang, and S. Wang ,J. Extraction of Cloud Storage Classification Rule Based on Genetic Algorithm. Computer Engineering, .39,7 **(2013)**

[13] H. Tang and B. Q. Zeng ,J. Research on Method of Text Classification Rule Extraction Based on Genetic Algorithm and Entropy. Acta Scientiarum Naturalium Universitatis Sunyatseni.46,5**(2014)**

[14] A. Sano and M. M. Vilela , J. Optimization of Test Cases Using Genetic Algorithm. Japanese Journal of Medical Mycology.41,3 **(2015)**

[15] V. A. Kostenko, and A. V. Frolov ,J. Self-learning genetic algorithm[J]. Journal of Computer & Systems Sciences International. 54 **(2015)**

[16] M. Pelikan, M. W. Hauschild, and D. Thierens, Edit. Pairwise and problem-specific distance metrics in the linkage tree genetic algorithm.// Conference on Genetic & Evolutionary Computation.**(2011)**

[17] H. Li, J. Design of evolutionary algorithm for the optimization of cloud storage deployment[J]. Journal of Southeast University.**(2013)**

[18] Z. X. Qin, X. Chen, X. P. Tang, and Z. H. YU, J. Optimization on fixed-charged transportation problem based on immune clonal selection algorithm. Application Research of Computers. 26,7**(2009)**

Authors

Tian Junfeng（1965-）,Male, hebei province, Hebei University professor. network security

Li Weiping（1988-）**,** Male, hebei province, Doctor degree, network security