International Journal of Database Theory and Application
Vol.9, No.5 (2016), pp.197-202
http://dx.doi.org/10.14257/ijdta.2016.9.5.20

Research of Dynamic Access to Cloud Database Based on Improved Pheromone Algorithm

Yongqiang Li[1] and Jin Pan[2]

[1](*Software Technology Vocational College, North China University of Water Resources and Electric Power, Zhengzhou 450045, China)*
1696450309@qq.com
[2] *(Henan Procuratorial Vocational College Zhengzhou 451191, China)*
1632921040@qq.com

Abstract

In cloud computing environments, the database is dynamic and real-time features, and in data access it has been a Gordian knot of cloud computing research; because in the ant colony algorithm that ants finding food shares similar correlation with cloud computing node finding the access to the database. Therefore, the introduction of ant colony algorithm into cloud computing database, the introduction of the chaos function in the pheromone update, both will make the improved pheromone avoid the possibility of getting into local convergence, and hence improve the efficiency of database access in the cloud computing, reduce the load in the cloud computing. The simulation results show that the algorithm in this paper has been significantly improved in terms of cloud computing network queries and database consumption, and greatly improved the efficiency of cloud computing.

Keywords*: cloud computing data; chaos; pheromone; ant colony algorithm*

1. Introduction

As the concept of cloud computing emerges, the network model, combining distributed concept with grid technology, has been widely applied, and the emergence of cloud computing can help cloud users use various resources under the cloud environment according to the demands [1]. How to best make users have access to data in the cloud database has always been a focus of the study. Owing to the dynamic and distributive natures of cloud computing, the database under cloud computing environment differs significantly from the traditional database. Besides, the emergence of cloud computing database has broken the traditional centralized, static and non-extendable storage method, and has won more and more attention.

Literature [2] proposed a CCRP algorithm. Based on the characteristics of data storage in the cloud computing database, this algorithm first applied continuous reading feature to date in query scheduling, solved the idle system resources in the cloud computing compared to other algorithms, and the simulation experiments proved that CCRP algorithm was superior to other algorithms in system utilization rate and system performance. Literature [3] proposed to combine ant colony algorithm with the cloud database, rapidly and reasonably find the database required to access from the cloud database through pheromone evaporation, and improved the efficiency of cloud computing. Literature [4] proposed a new cloud relational database platform, described the vertical mechanism and horizontal mechanism of data, and the experimental results showed that the cloud relational database platform had good reliability, extendibility and usability. Literature [4] proposed a tree-based cloud database. The database organized and retrieved data in the form of tree, reorganized the existing data storage method in the form of tree, and satisfied the requirements over data storage and retrieval. However, it had the weaknesses of higher data retrieval costs, so that it was impossible to implement cloud computing database retrieval under big data. Literature [6] proposed the best practice and the actual operating technical indicators for database cloud platform in telecommunications companies. In foreign counties, Google uses the computer cluster composed of cheap PCs, and achieved the GFS distributed, BigTable distributed and MapReduce [7-9] environments, in order to handle big data in the cloud database as many as possible. Literature [10] proposed a performance function based on Session and content ratings, took full account of the customer priority, and improved the database access efficiency.

The traditional ant colony algorithm is a discrete optimization to solve NP problems, which is a good solution to the traditional static combination optimization problems. Based on the traditional ant colony algorithm, this paper improves the ant colony algorithm, and the improved ant colony algorithm has good advantages in solving the access of cloud computing database. The simulation results show that the presented algorithm can effectively reduce the access time of cloud computing database and enhance the access efficiency of cloud computing network.

2. Relevant Knowledge

2.1 Ant Colony Algorithm

Ant colony algorithm is a swarm intelligence and bionic optimization algorithm, which is mainly used to solve complex combination optimization problems. It is characterized by intelligent search, global optimization and distributed computing. Ant colony algorithm means that ants leave a secretion called pheromone on the path of seeking food, and other ants will choose a path according to the pheromone secretion. As a result, many ants travel down a certain path within a short time, forming a positive feedback mechanism and thus producing the shortest path from the colony to a food source.

2.2 Description of Cloud Database Problems

Cloud computing database consists of mass databases distributed in various parts. However, the actual cloud database has the following problems. On the one hand, a part of databases are stored in the fixed cloud server, whereas the other parts are stored in the mobile intermediate service providers. These mobile service providers continuously interact with cloud computing with the joined or quit cloud nodes. On the other hand, could computing has cloud nodes or link failures at any time and new cloud nodes and links join in, which bring big difficulties to the access to resources in the cloud database. Therefore, how to achieve data access in this dynamic environment has become a key to solving problems.

The user query operation process of cloud database is as follows:

(1) Cloud users send the contents to be queried to the database server through the network;

(2) Cloud server processes SQL commands sent by the users and distribute data sub-commands to each storage cloud nodes through pre-processing.

(3) Each storage cloud node interacts between the different nodes after receiving SQL commands.

(4) Upload the query results of each cloud node to the cloud server for summary.

(5) Return to users and complete the query after server making summary.

In the process the user query, there are many factors involving the query efficiency, such as time for users transmitting to server, server resolved time, time for server sending to nodes, query time between nodes and time for nodes feedback to server. In order to best guarantee the efficiency of the user access to the database, the key question is how do nodes find the required resources from the database. The process of resources seeking is to quickly and safely seek an optimal path from the cloud starting point to the end point, which is very similar to the food seeking process of ant colony algorithm in reality. In ant colony algorithm, after ants seeking foods, the process from source node to the original node is equivalent to cloud users sending requests to the database, and the process of recording returning from the database is obtained. In this process, different nodes obtained from different positions is equivalent to the database nodes in cloud computing.

Since the cloud computing adopts the column storage to manage data, it is very important to make query scheduling for the database in cloud computing. The data query process in cloud computing is combined with the flow distribution, congestion control and route selection processes in cloud computing. Compared with traditional static databases, the cloud computing database gives more consideration to dynamic and real-time storage. The routing table records the information required for the routing algorithm, which can ensure that the scale of the cloud computing database can seek the optimal solution under certain conditions. However, if the cloud computing database has a relatively large scale, it may be impossible to seek the optimal solution. Obviously, it is an inevitable trend for the increased scale of cloud computing database, which cannot reach the accuracy of routing algorithm to some extent. Therefore, the access to cloud computing database is the combination optimization problem in planning.

3 Application of the Algorithm in Cloud Computing Database

3.1 State Transition

Combined with the features of ant colony algorithm, assuming that m represents the database number in cloud computing, d_{ij} represents the distance between the cloud database i and the cloud database j, $\eta_{ij} = 1/d_{ij}$ stands for the heuristic information, τ_{ij} is the amount of pheromone between ij, and $p_{ij}^{k}(t)$ is the possibility of k th request at t in the cloud database i to query the cloud database j, then

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum\limits_{s \in allowedk} [\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, j \in allowed_{kj} \\ 0, otherwise \end{cases} \tag{1}$$

In the equation, the log sheet $tabu_{kj}(k = 1, 2, ...m, j = 1, 2.......n)$ is used to represent the record j having access to the k th database, being used for the current traveling database set of database record j, and $allowed = \{C - tabu_{kj}\}$ represents the record j can have access to the next database k. α and β respectively represent the weights of τ and η in computing. If α =0, the database nearest to the access request may be first accessed. If β =0, when the pheromone in ant colony algorithm works, it becomes a heuristic search algorithm. When all access records have searched all databases, the pheromone similar to the ant colony algorithm will be updated.

3.2 Updating Rules for Traditional Pheromone

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{2}$$

In Equation (2), ρ represents the factor of pheromone evaporation, and $\Delta\tau_{ij}^{k}(t)$

represents the trail (pheromone) that the j th record has access to the database k and traveling through the path i, j at t.

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L_{kj}} \\ 0 \end{cases} \tag{3}$$

Where, $\dfrac{Q}{L_{kj}}$ represents the ratio of length between node access database j and the database k, where, L_{kj} represents the number of database accessed by the j th record. In the previous ant colony algorithm, an optimal solution can be obtained through the positive feedback mechanism. However, as the path (nodes in cloud computing) increases, it is easy to cause relatively long searching time. If the recorded node access trails (pheromones) are centralized to the optimal path, the stagnation problem will be emerged easily and the local optimal solution is obtained. The optimal query to the database access records in the cloud database may not be necessarily achieved, and thus lose the query to the optimal solution.

3.3 Improved Pheromone Rules

Since the ant colony algorithm convergence is slow, the ants cannot be guided to the optimal solution to a certain extent, which is largely influenced by pheromones. The pheromone updating cannot guide the generation of optimal solution from a global perspective, cannot guide the cloud nodes correctly access the cloud database from the perspective of cloud computing database, or it has the problems of low access inefficiency. The pheromone updating in the ant colony algorithm cannot accurately reflect the path, it cannot reflect the recorded accurate database access in the cloud computing database, and thus the optimal solution is affected. On this basis, this paper changes the pheromone equation, namely, introduces chaos algorithm to the pheromone updating. The idea of chaos algorithm is to compare in the traversal process, save the relatively good solution in the comparison process and seek the optimal solution in the iteration search process. Ideas are as follows:

3.4 Algorithm Description

In the ant colony algorithm, the process of ants seeking food is the process of cloud nodes access to the data in server. This part combines the database query with the ant colony algorithm, and the algorithm steps are as follows:

Step 1: Initialization: Place the ants of simulation node access on a cloud node x, list the end point server node information to the log sheet $tabu_{kj}$, set the maximum frequency of ant food seeking nodes as Max; meanwhile, find whether the node x belongs to the feasible node region $Feasible_i$. If feasible, go to Step 2

Step 2: Take the current node x as the center, select the adjacent selection strategy and go to the next node y. The node y belongs to $Feasible_i$, then the node y constitutes a combination $\{y_i\}$, the nodes select according to the possibility value of adjacent nodes, and select the node with relatively larger possibility value as the cloud node x and as the next node.

Step 3: According to the pheromone and other information in the ant colony algorithm left by ant in the traveling process, as time goes by, pheromone updating will follow the formula (2) for local updating, and leave pheromone for subsequent ants, in order to provide reference to the database query of other nodes in the cloud computing and reduce the access time.

Step 4: After the request sent by node x is replied, the global pheromone updating rule, as mentioned in Section 4.3, is used for the query pheromone in the shortest path, then update and save.

Step 5: According to the definition of ant colony algorithm, check whether the node access requirement in cloud computing is replied by server. If yes, go to Step 6. If not, go to Step 2.

Step 6: Find the path, the server returned node information returns to the starting position of node x along the original path.

4. Experiment and Analysis

In order to prove superiority of the algorithm, this paper conducts the simulation experiment in the VC ++ 6.0 design environment, and uses six sets of computers configured as CPU Core i32.2G, DDR4G memory and 500G hard disk. One set of PC is selected as a server to control the database system, whereas the rest five sets are installed with SQL Server2008 as the data storage nodes. This paper mainly applies system analog data for experimental data collection, sets the capacity of access data between [50M, 100M], makes comparison between two indicators of the data access network consumption time and data access time, which compares the Literature [6] algorithm with Literature [10] algorithm.

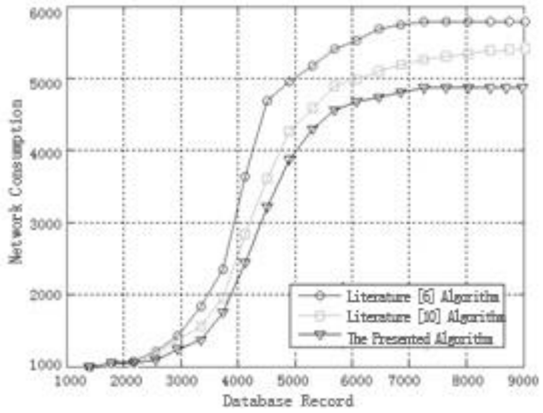

Figure 1. Comparison of Network Consumption Time

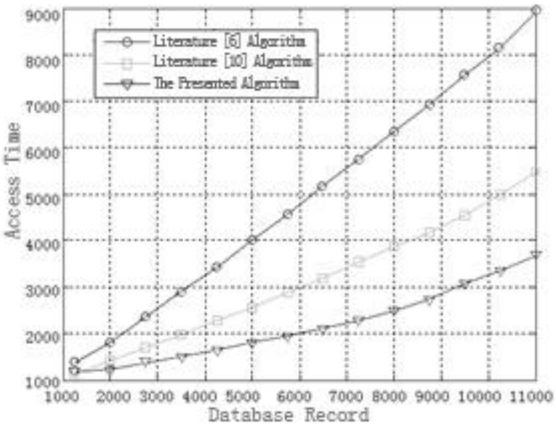

Figure 2. Data Access Time

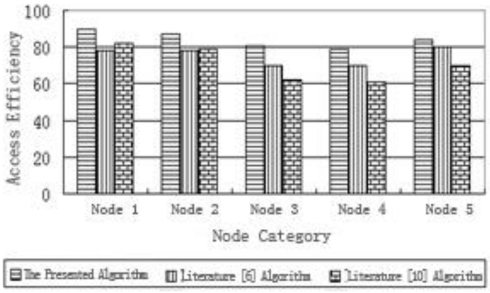

Figure 3. Node Access Efficiency

According to Figure 1-3, compared with the Literature [6] algorithm with Literature [10] algorithm, the presented algorithm is significantly improved in terms of network consumption time, database access time and node access efficiency, which is mainly caused by the pheromone updating improvement in ant colony algorithm so as to avoid the possibility of local optimal solution, make the time for targeting database of node access reduce and effectively decrease the network consumption.

5. Conclusion

This paper combines the ant colony algorithm with cloud computing database, and simulates the record query process of node access database in cloud computing as the process of food seeking. First of all, this paper improves the pheromone based on the ant colony algorithm, enables the improved pheromone to effectively prevent the possibility of local convergence, conducts the simulation experiment, and further proves that the presented algorithm is significant improved in terms of query and network consumption in the cloud computing database as compared to the literature algorithms, which provides a reference for the database query in the cloud computing environment.

References

[1] L. Z. Yun, “Research on cloud Database”, Journal of software, vol. 23, no. 5, **(2012)**, pp. 1148-1166.

[2] Z. L. Yun, “Database Query Scheduling Algorithm Based on the C loud Computing”, Journal of Zhengzhou University (Engineering Science), vol. 31, no. 7, **(2010)**, pp. 65-68.

[3] S. H. Liang, “Cloud database dynamic route query based on self-adaptive ant colony optimization”, Computer Engineering and Applications, vol. 46, no. 9, **(2010)**, pp. 10-12.

[4] C. P. Ping, “Research on scalable cloud relational database”, Computer Engineering and Design, vol. 33, no. 7, **(2012)**, pp. 2690-2695.

[5] K. Y. Mei, “Adaptive Resource Provisioning for Cloud Computing”, Telecommunications Science, vol. 28, no. 1, **(2012)**, pp. 37-41.

[6] L. Yong, “Construction and Operation Program of Enterprise-Class Database Cloud Platform”, Telecommunications Science, vol. 28, no. 2, **(2012)**, pp. 146-155.

[7] S. Ghemaw, H. Gobioff and L. S. Tak, “The Google Files System”, SIGOPS Operation Systems Review, vol. 37, no. 5, **(2003)**.

[8] F. Chang, J. Dean and S. Ghemaw, “Bigtables: A Distributed Storage System for Structured Data”, 7th Symposium on Operating Systems Design and Implementation. Seattle, WA, USA, **(2006)**, pp. 205-218.

[9] J. Dean and S. Ghemaw, “MapReduces: Simplified data processing on large clusters”, Communications of the ACM, vol. 51, no. 1, **(2005)**, pp. 107-113.

[10] Z. J. Juan, L. Tao and Q. Yan, “Classification Algorithm of Database Request Based on Session and Content Level in Cloud Computing”, Computer Science, vol. 40, no. 2, **(2013)**, pp. 177-179.

Authors

Yongqiang Li, (1974.1-) Lecturer, Master, Research Orientation: Computing Network.

Jin Pan, (1974.11-) Associate Professor, Master, Research Orientation: Computer Network.