International Journal of Database Theory and Application
Vol.9, No.4 (2016), pp.71-78
http://dx.doi.org/10.14257/ijdta.2016.9.4.06

Research in the Cloud Calculation Database Based on Improved Ant Colony Algorithm

Xuan Chen and Jun Chen

(Zhejiang Industry Polytechnic College Shaoxing Zhejiang 312000 China)
734140999@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

Along with its emergence, the concept of cloud computing which is a kind of network model that combing distributing concept and grid technology is being used widely, and it can help the cloud users to take use of all kinds of resources under cloud environment according to their needs [1]. How to maximally satisfy users with data access in database is always the research focus, because the database under cloud computing environment is very different from traditional one for its dynamic and distributed characters. The appearance of cloud computing database breaks the traditional kind of centralized, static and non-extensible storage mode and is getting more and more attention from people.

In reference [2], CCRP algorithm is proposed; the algorithm solves the idle problem of partial system resources for other algorithms in the cloud based on data storage mode in cloud computing database and by using continuous reading characteristics applied to the data as the prior when querying scheduler, and the simulation shows that the CCRP is better than others both in respects of system utilization and system performance. The reference [3] presents that by combing ant colony algorithm with the cloud database and volatilization of the pheromone, the required database can be fast and reasonably found in the cloud database and the efficiency if cloud computing will be enhanced. Reference [4] presents a new cloud relational database platform which describes the vertical and the level mechanism of data, and the experiment shows its reliability, extensibility and ease of use. Reference [5] proposes a kind of tree cloud database which organize and retrieve data in the form of a tree, to reorganize existing data storage mode in form of tree meets the requirements of data storage and retrieval but the drawback is high cost of data retrieval so that it is unable to cope with the cloud computing database retrieval under large amount of data. In reference [6], it proposes the technical indicators of best practices and actual operations for the telecommunications company's database cloud platform. Google in abroad uses computer clusters composed of inexpensive PC to achieve the GFS Distributed, BigTable Distributed and MapReduce [7-9] cloud database environment to process more data in the large as possible.

The traditional ant colony algorithm is a kind of discrete optimization problem that solves NP problem and it has good solution to solve the traditional static combinatorial optimization

ISSN: 2005-4270 IJDTA

problem. In the basis of the traditional ant colony algorithm, this paper will make improvement on it and the improved one shall have very good advantage in addressing access of cloud databases. According to the simulation results, algorithm in this paper effectively reduces the access time to cloud databases and enhances access efficiency of cloud computing networks.

2. Relevant Knowledge

2.1 Introduction for Cloud Database

Cloud database system is composed with several cloud nodes, each of which has its own database system and hardware processing mechanism. Therefore, in a sense, it is a collection of centralized database which logically belong to the same system but locate in different region in respect of physical structure. It is a data resource management platform facing cloud computing, mainly to provide cloud computing accessing environment for resource sharing of database that largely exists in network backstage and to provide infrastructure's database resource access and generic integrated solution for the front-end application. Its advantages are as follows:

(1) It at the most avoids the possibility of overall paralysis caused by centralized database failure. When the traditional centralized database has a data failure may cause a visitor inaccessible to the data. As loud computing database is distributed, fault only confined to the local cloud database, and will not affect the normal access to other databases.

(2) It effectively solves the problem that cloud node database need to establish an internal links because of its independent structure. Database under cloud computing environment have different physical locations and various database structures, however, under the cloud computing condition, internal access between databases shall be carried out according to a uniform format.

(3) Cloud database effectively solves the bottleneck problem of resource load of centralized database. Each database under cloud computing environment all process data so as to achieve a local optimum and they can deal with resource in a more reasonable way with each other, which effectively avoid resource load bottlenecks. Cloud database is composed by the massive database distributed around the world, however it actually has problems as below: on one hand, some of the database is stored in a fixed cloud server while the other is stored in the mobile service providers who make interactional communication with cloud node along with their join or quit; on the other hand, there may exist possibility of cloud node and link failure at any time in cloud computing and at the same time there are new cloud nodes and links added, which brings great difficulty for resource access to cloud database, so how to access data under this kind of dynamic environment consequently becomes the key to solve the problem.

Processes of query operation for cloud database users [10] are as follows:

(1) Cloud user sends the query contents to the database server through the network;
(2) The cloud server processes the SQL commands sent by user and then distribute data query subcommands into each of the stored nodes after a pretreatment
(3) Each of the nodes makes interaction within different ones when they receive the SQL command.
(4) Upload the query results from each node to cloud server for collecting and summary.
(5) After the collecting and summary, the server sends them back to user to complete the query.

In the user query process, many factors will make effect on query efficiency, mainly includes: the time users used to transfer data to the server, the server resolution time, the time for the server to send a query to node, interact checking time between each of the nodes and the feedback time for the node to the server and so on. The key to maximum ensure the efficiency of the user's access to the database is that how the node can find the resources for its own from the database. The process of finding resources is actually the process to find an

optimal path that the query can quickly arrive to the end of the cloud from the starting point. This is very similar to the form of food finding process of the ant colony algorithm in reality. In ant colony algorithm, the process of ants to return from the source node to the first node after finding the food is equivalent to the process of users to send requests to database and receive turn-back records from the database, moreover, the different nodes encountered in this process is equivalent to the cloud database node.

As cloud computing uses a column storage method to manage data, the way to make inquire and dispatch for the cloud computing database becomes consequently very important that the data lookup process shall be combined with the process of flow distribution, congestion control and routing choosing in cloud computing. Different from the traditional static type of databases, cloud database take more consideration on dynamic and real-time storage. The routing table in it records the information required for routing algorithm, while the routing algorithm is able to ensure to find the best resolution if the size of the cloud database is under certain conditions but becomes unable if the size of the database is larger which is obvious the inevitable trend and that it to some extent cannot meet the need of the routing algorithm to achieve accuracy, so the accessing problem of cloud computing database is combinatorial optimization problems in programming.

2.2 Ant Colony Algorithm

Ant colony algorithm is a kind of Bionic optimization algorithm of swarm intelligence that mainly to solve complex combinatorial optimization problems. It has characters as intelligent search, global optimization, and distributed computing. The ants leave a kind of secreted substance named pheromones in the path of finding food and the follow-up ants can choose the path in accordance with the level of the secrete pheromones, this is called ant colony algorithm . And this results that there are many of ants getting through a certain path and a positive feedback mechanism is formed, thus the shortest path from the nest to a food source is formed.

3. Applications of Algorithm of this Paper in the Cloud Computing Database

3.1 State Transition

Combining ant colony algorithm features, the author assumes: m represents the number of databases in the cloud, d_{ij} represents the distance between the cloud databases i and j, $\eta_{ij} = 1/d_{ij}$ equals to heuristic information, τ_{ij} represents pheromone between i and j, $p_{ij}^{k}(t)$ means the probability of the k request in i database to query j database at time of t, then:

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

In formula, recording table $tabu_{kj}(k = 1,2,....m,\ j = 1,2.....n)$ is used present j record of accessed k database and database collections the j record has ever gone through, $allowed = \{C - tabu_{kj}\}$ means that j record is allowed to access the next database k. α and β separately represent the importance of τ and η in computing, as α =0, the database nearest to access request may be firstly accessed; as β =0, when the ant colony algorithm pheromone work, it becomes a heuristic search algorithm. When all access records complete the searching for the entire database, pheromone similar to the ant colony algorithm

will be updated.

3.2 Updating Rule of Traditional Pheromone

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

In formula (2), ρ represents pheromone evaporation factor, $\Delta\tau_{ij}^{k}(t)$ means the traces (pheromone) left by the record j in the path when accessing database k at the moment t.

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

$\dfrac{Q}{L_{kj}}$ represents the length ratio between the database j and the database k, while L_{kj} means the number of accessed database k for record j. In the ant colony algorithm described foregoing, an optimal solution can be obtained by using positive feedback mechanism, but along with the path (cloud computing nodes) increase, it is prone to appear a situation of relatively long time for searching. However, if the record access node traces (pheromones) concentrate on the optimal path, it is easy to have a stagnation situation, which would produce a local optimal solution that not necessarily to achieve cloud database access to database records with the best finding, thus losing the query for the optimal solution.

3.3 Improved Pheromones Rule

It is unable to guide the ants to the optimal solution to a certain extent for the convergence of the ant colony algorithm is slow and mainly because of the influence from pheromones. The update of pheromones cannot guide to produce the optimal solution from a global point of view, and from the view of cloud computing database, it cannot guide cloud nodes to rightly access the cloud databases or it can be said to a low efficiency of accessing. The update of pheromones is not able to reflect precisely the path pros and cons and so is unable to reflect the phenomenon of record of accurate access to database in cloud computing database and that affects the optimal solution. On this basis, the paper makes change on pheromone formula which means to introduce chaos algorithm into pheromone update. Chaos algorithm idea is to compare in the course of traversing, to store relatively good solution in the process of comparison, and to make iterative query process until finding the optimal solution to meet the conditions. The idea is as follows:

3.4 Algorithm Description

The process for ants to search food in the ant colony algorithm is the process for cloud nodes to send access data to the server; this section will make combinational description on database query and ant colony algorithm, and the algorithm steps are as follow:

Step 1: Initialization: place the ants accessing the simulation node onto one of the cloud nodes and at the same time list the node information of end server that need to access into record table, then set the maximum number of nodes of ants foraging as Max; meanwhile, check whether the nodes belong to feasible node area, and if it is yes, then proceed to step 2.

Step 2: With the current node x as the center, go to the next node y that belong to $Feasible_i$ by selecting approaching selection policy, so that node y form into a combination $\{y_i\}$, according to the probability values of the adjacent thereto nodes, the node select a node with relatively large probability value as a cloud node for the next node.

Step 3: According to the left pheromone and other information in ants' forwarding process

in the ant colony algorithm and along with the passage of time, the pheromone update will make partial update according to equation (2) to leave message for subsequent ants. The role of it is to provide a reference for other nodes in process of the cloud database query, which reduces the access time.

Step 4: After getting response to the requests sent by the performing node x, it will make update and storage for the pheromone on the shortest path in this query by using global pheromone update principle mentioned in Section 4.3.

Step 5: Define whether the cloud computing node access get response from the server according to the ant colony algorithm, if yes, go to step6, if not, go back to step 2.

Step 6: Find the path, node information replied from the server will return back to the starting position of node x through the original path.

4. Algorithm Implement

It can be found in foregoing described cloud computing database access that time is the main factor that determines the query efficiency in which node's accessing time in the link plays the crucial role. Therefore, the algorithm of this essay needs to effectively reduce the link time. According to actual conditions of the cloud computing database, the related records may distribute on different cloud nodes, so it will not influent largely if they are on the same node database but needs to make data transmission if on the contrary. Below the essay takes query algorithm as an example, wherein, $sub\_sql \to x$ means to present access request as x, $L = count(\{x_1, x_2 .....x_n\})$ presents the number of database that x has accessed, $\tau_i(t+1) \to length_i$ expresses the path for node i corresponding with improved pheromone to access database.

Client Port:

$input : user\_query$

$begin$

$send\ \ user\_query\ \ to\ \ Server$

end

$output : user\_result$

Server Port:

$input : sql$

$begin$

$sql1 = gets(sql\ \ )$

$sql2 = process(sql\ \ 1)$

$snode(sql\ \ 2, nodes\ \ )$

$noticefy(user\ \ \_\ \ inof$

$receive\_reply(snode$

end

$output : sql\_result$

Node Port:

$input: sql2$

$begin$

$sub\_sql = get(sql2)$

$sub\_sql \rightarrow x$

$$(p_{ij}^{k})_x = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{s \in allowedk} [\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}$$

$L = count(\{x_1, x_2 ..... x_n\})$

$L = Sin(\pi L)$

$$\tau(t+1) = (1-\rho)\tau(t) + \sum_{k}^{m} \frac{Q}{L}$$

$length_x = \min\{\tau_i(t+1) \rightarrow length_i,$

$\tau_2(t+1) \rightarrow length_2, ... \tau_n(t+1) \rightarrow length_n\}$

$length_x \rightarrow x_i$

$reply(x_i)$

end

5. Experiment and Analysis

In order to verify our algorithm has advantages, simulation experiment is carried out in VC ++ 6.0 designed environment by using six computers which have configuration of CPU core i32.2G, DDR4G memory and 500G hard drive. One of the six PC is used as the server who will take control on database system and the other five ones are installed with SQL Server2008 as the data storage node, the experimental data collection mainly uses system simulated data, the data capacity size is set among [50M, 100M], and the comparison is taken between two index of time-consuming for data to access internet and time of data accessing. The method for comparison is adopted from reference [6] and [10].

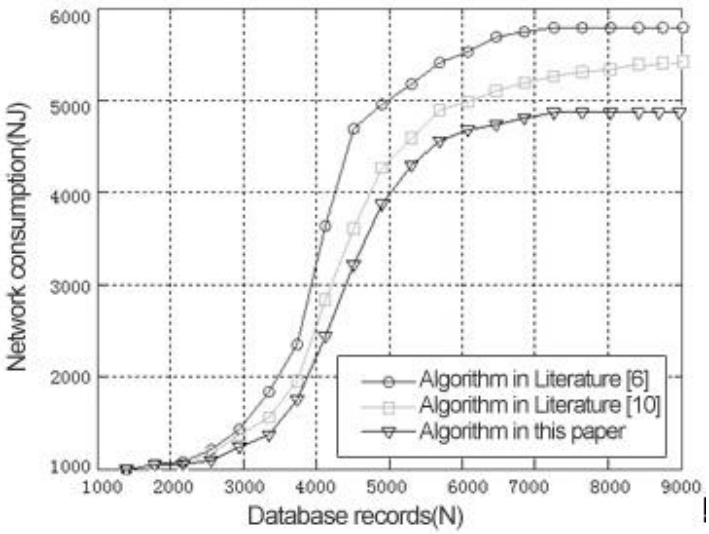

Figure 2. Comparison of Internet Time-Consuming

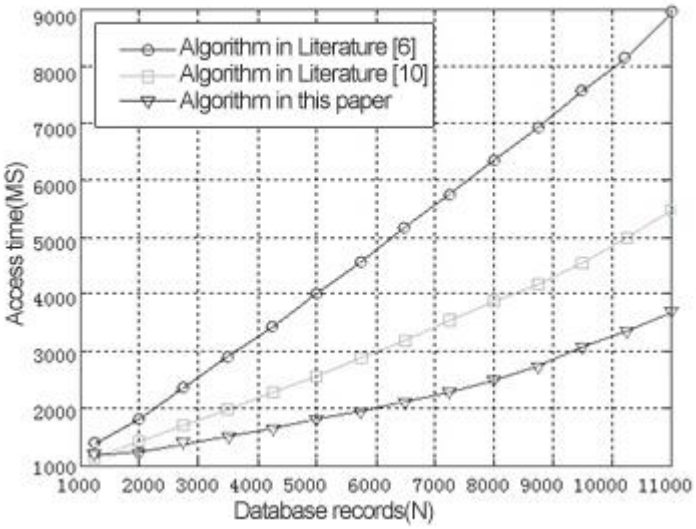

Figure 3. Time of Data Accessing

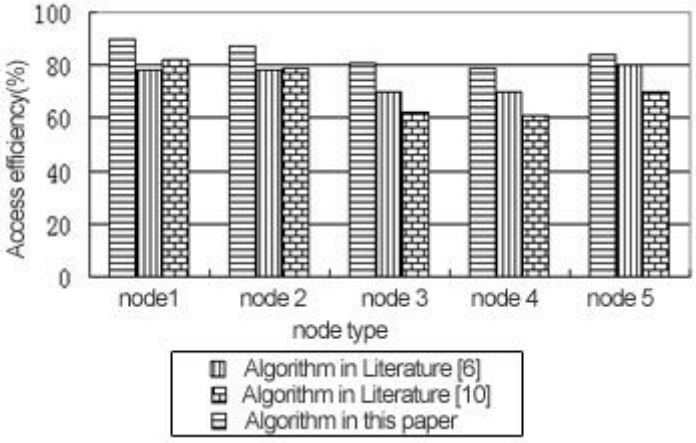

Figure 4. Nodes' Accessing Efficiency

It can be found in Figure 2 to 4 that our algorithm has significant improvement either in respect of time-consuming in the network, database access times and the node access efficiency, which mainly because of the ant colony algorithm pheromone update has been improved that avoiding the possibility of falling into local optimal solution and reducing the time for nodes to access database and the network consumption effectively.

6. Conclusion

How can access the data in the cloud computing in a more accurate way is always a hot for the cloud computing research. In this paper, the ant colony algorithm and cloud computing database is combined, the record query process for nodes to access to database in the cloud computing is simulated as the process of finding food in ant colony algorithm. On the basis of ant colony algorithm the pheromone is firstly improved so that the improved information effectively avoids the possibility of local convergence for the algorithm. It further proof through simulation experiment that the algorithm of this essay has an obvious improvement comparing with the one mentioned in other references in respects of query and network consumption in the cloud computing database which provides a kind of reference for database query under the cloud computing environment.

References

[1] L. Z. Yu, "Research on Cloud Databases", Journal of Software, vol. 23, no. 5, **(2012)**, pp. 1148-1166.
[2] Z. L. Yun, "Database query scheduling algorithm based on Cloud 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 antcolony optimization. Computer Engineering and Applications", Computer Engineering and Applications, vol. 46, no. 9, **(2010)**, pp. 10-12.
[4] C. P. Ping, "Reaearch on scalable cloud relational database", Computer Engineering and Design, vol. 33, no. 7, **(2012)**, pp. 2690-2695.
[5] K. Y. Mei, "Tree based cloud database for SAAS mode cloud services", Telecommunications Science, vol. 28, no. 1, **(2012)**, pp. 37-41.
[6] L. Yong, "The construction and operation of enterprise 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, November, **(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] W. M. Bo and D. Z. Ming, "Selection Oriented Database Data Distribution Strategy for Cloud Computing", Computer Science, vol. 37, no. 9, **(2010)**, pp. 168-172.

Authors

Xuan Chen, (1979.03-), male, master, lecturer, research direction: information security, wireless sensor and cloud computing.

Jun Chen, (1980.08-), Female, master, deputy professor, research direction: cloud computing, information security.

