International Journal of Database Theory and Application
Vol.9, No.4 (2016), pp.55-60
http://dx.doi.org/10.14257/ijdta.2016.9.4.04

Research of Access to Cloud Database Based on SVM-ACO Algorithm

Bin Ning, Qiong Gu and Yi Wang

(School of Mathematics and Computer Science, Hubei University of Arts and Science, Xiangyang, Hubei, 441053, China)
hubeiningbin@163.com

Abstract

How to improve the accessing efficiency of database based on cloud computing has always been a research direction of cloud computing and this paper combines genetic algorithm, ant colony optimization as well as support vector machine to firstly initialize the ant colony optimization and improve the pheromone so as to make the advanced information effectively avoid algorithm local convergence. What's more, the collecting result can be processed via support vector machine and meanwhile the simulation experiment proves that the algorithm adopted in this paper is beneficial to improve the query efficiency of cloud database and provides valuable references for the cloud data research.

Keywords*: cloud computing data; pheromone; ant colony algorithm; penalty factor*

1. Introduction

How to meet users' requirements for having access to data in cloud database at maximum has been the focus of research, and cloud computing has such features as dynamic nature, distributivity and others, resulting in the great difference between the database under cloud computing environment and traditional database. The emergence of cloud computing database has broken the traditional, centralized, static and non-expandable storage mode, which has arose more and more attention [1].

The literature [2] adopts cloud platform virtual machine as the partial storage node to realize massive user data storage based on sharding and read-slaves. What's more, it adopts flexible key value to customize data and puts forward customized data visit controlling method based on metadata and at the same time designs a visit controlling algorithm. Literature [3] adopts ICFV based on proper vector to optimize the data load assess performance. ICFV algorithm extracts the proper vector of database loading and then carries out division which is different from the traditional methods because it adopts incremental clustering algorithm and it is not necessary to calculate the center distance to optimize the eigenvector dimension. The experiment elaborates that adopting ICFV algorithm is able to realize the optimization of database loading self-adaptation and improve loading division efficiency. Literature [4] adopts parallel schedule for a complicated database with chaos to integrate the chaotic priority factor and parallel database schedule technology and introduce it to optimum calculation to obtain the chaotic characteristics and improve the overall searching ability. Literature [5] comes up with a dynamic path optimization for adaptive immune polymorphism ant colony optimization. By supervising the scouts ant and search ant, introducing self-adaptive multi-ant colony competition strategy, improving the demerits of local optimal solution we can further integrate the artificial immune algorithm with overall searching ability to improve the search ant path optimization which can improve the search speed and accuracy. In addition, the simulation experiment proves that this algorithm can better deal with the problem of convergence speed and global optimum so as to quickly and reasonably find out the needed database. Literature [6] comes up with a load assessment algorithm based on great entropy difference, adopts evaluation function to change the target

ISSN: 2005-4270 IJDTA

optimization to non-differentiable single-objective optimization. Besides, it takes advantage of the greatest entropy function to change non-differentiable optimization to unconstrained optimization problem with parameter and finally adopts differential evolution to find the solution and find out optimal resources so as to provide theoretical basis for overloading nodes and further realize cloud database design. The experiment also proves that this algorithm can effectively improve the overall performance of the whole system and avoid the bottle neck of single node. Literature [7] suggests to adopting ontological theory to summarize the common characteristics of various management information and establish ontology representation system which is large and can reflect with human with compact treatment. At the same time, based on the principle of application, convenience and low-cost, we can carry out database layer architecture design to construct a static metadata index term which can make logic relationship with management information ontology to finally create a cloud service system based on the ontology requirement. Literature [8] comes up with a data query method based on cloud computing, which firstly adopts cloud computing technique to find out targeted nodes and then introduces single gene theory to traditional cloud computing genetic algorithm to define similar nodes and equal nodes comparing to service node to carry out a second searching to combine excellent individuals as a totally new. Besides, it takes advantage of any excellent genes in the course of inheritance so as to guarantee the accuracy of optimal query result. The experiment result shows that this algorithm is superior comparing to traditional methods in terms of data quality, service node load capacity and query efficiency.

Based on the above researches, this paper introduces ant colony alorithm into cloud database, and initializes it through genetic algorithm. Then, it improves the pheromone so that the improved information effectively avoids the possibility that the algorithm falls into local convergence. Aiming at the collection results of the algorithm, it processes results through the supporting vector machine. Simulation experiment shows that algorithm in this paper has some certain effects.

2. Problem Description of Cloud Database

Deploy database to the cloud database can simplify information available through the Web network connection of business process, support and ensure that the cloud business application as part of the software as a service (SaaS) deployment. In addition, the enterprise database deployment to the cloud can also realize storage consolidation. Stored in the cloud computing vast amounts of data information, on the one hand, part of the data is stored in the cloud server, and there is another aspect of data providers in the middle, with the cloud of cloud computing nodes to join or quit dynamically, this leads to a cloud computing the data in the database when it has certain uncertainty, therefore access to cloud computing database will have certain difficulty.

3. Applications of SVM-ACO Algorithm in Cloud Computing Database

3.1 Population Initialization

The genetic algorithm is adopted to initialize the solution of initial ants group so as to obviously improve the efficiency. Suppose the population size of ant colony optimization is M and all ant groups are divided into two groups, which are parent and subpopulation, among which the parent population is $Z = \{\alpha_1, \alpha_2, \ldots a_M\}$ and suppose the fitness function of every individual is $f(\alpha_i)$, subpopulation is $X = \{\delta_1, \delta_2, \ldots \delta_M\}$. The individuals of parent population are sorted from big to small, and then the group is $Z^{'} = \{\alpha_1^{'}, \alpha_2^{'}, \ldots a_M^{'}\}$ and the fitness value from big to small is $f(\alpha_1^{'}) > f(\alpha_2^{'}) > .... f(a_M^{'})$. Then, we will sum the fitness value, which is $\sum_{i=1}^{M} f(\alpha_i^{'})$ to calculate the selection rate $P(i)=\frac{f(\alpha_i^{'})}{\sum_{i=1}^{M} f(\alpha_i^{'})}$ of being selected and every individual cumulative probability is $Q_i=\sum_{i=1}^{M} P(i)$. The strategy for wheel disk is adopted to generate M random number and if $R \le Q_i$, b_1 chromosome is chosen or we will choose i chromosome b_i. Therefore, M_i is $b_{\min(i1,i2,\ldots in)}$, and we can combine M_i and subpopulation X to make $X(t)=X(t-1)\bigcup M_i$. Based on this, these individuals are crossed and mutate to have the final solution which can be deemed as the initial solution of the ant colony optimization.

3.2 Improvement of Pheromone Rules

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

In equation (1), ρ represents factor of pheromone volatilization, $\Delta\tau_{ij}^{k}(t)$ represents trace left by the j th record in the process of getting access to the database k at t point through (i, j) path (pheromone).

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\frac{Q}{L_{kj}} \\ 0\end{cases} \quad (2)$$

$\frac{Q}{L_{kj}}$ Represents the ratio between the length of the node to access the database j and that of the node access to the database k, of which L_{kj} represents the number of database k accessed by the j th record. In the foregoing description of Ant Colony Optimization, an optimal solution can be obtained by a positive feedback mechanism, but with the path (cloud computing nodes) increases, the search time is relatively long. But if the node access traces in the records(pheromones) are focused on the optimal path, stagnation is prone to happen, which would produce a local optimal solution and may not necessarily achieve the optimal searching for cloud database access records, thus failing to query for the optimal solution.

From the perspective of cloud computing database, it is impossible to guide the cloud nodes to get access to the cloud databases or the access efficiency is not high, and since pheromone update in Ant Colony Optimization can't reflect the pros and cons of paths accurately, the accurate access of cloud computing database is impossible, affecting the optimal solution. On this basis, the article has altered the pheromone formula, which means that chaos algorithm is introduced in pheromone update. Chaos Algorithm idea is that comparison is carried out in the ergodic course and relatively good solution is saved in the process of comparison, and iteration will be carried out in the searching process until you find the optimal solution to meet the conditions. Idea is as follows:

$for\ i=1\ to\ N\ do$

$\rho\in(0,1)$

$for\ \ j=1\ to\ \ k\ do$

$L_{i,j}=Sin(\pi L_{i,j})$

$endfor$

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\sum_{k=1}^{m}\frac{Q}{L_{kj}}$$

$endfor$

$endfor$

4. Algorithm Description

As for Ant Colony Optimization, the process of looking for food for ants is equivalent to the process of getting access to data for cloud node through server, and this section will combine database queries with Ant Colony Optimization, the algorithm is as follows:

Step 1: Initialization, classify the ants into main group and subgroup and then initialize the genetic algorithm and put in into the cloud. What's more, the assess server should be recorded and the greatest iteration number should also be set.

Step 2: Take the current node x as the center and choose close selection strategy to next node y, which belongs to $Feasible_i$. Then node y will construct a group $\{y_i\}$, and the close probable value is adopted to choose the next cloud node x.

Step 3: according to such information as pheromone left by ants in the advancing process in Ant Colony Optimization, along with the passage of time, the pheromone update will be competed partially according to equation (2) to leave a message for subsequent ants. Its function is to provide a reference for other nodes in the cloud database query, reducing access time.

Step 4: Once the request sent by the node x has been replied, pheromone on the shortest path for this query should be updated and saved.

Step 5: according to the definition of Ant Colony Optimization, we need to know whether the access request of cloud computing node can be replied through server, if it is replied, it can proceed into Step 6; if not, it can proceed into Step 2.

Step 6: find path, server can return node information to the original location of node x along with the original path, and then record the preliminary result.

Step 7: preliminary classification should be carried out for the above preliminary result, it is assumed that the result is the document set $D_n=\{txt_1,doc_2,excel_3\cdots\}$ under the cloud computing environment

Step 8: according to the content of the document set, apply Map function to convert to $f(D_i)\to<c_i,D_{ni}>$, it is indicated as $<c_i,f(D_{ni})>$ through Reduce function

Step 9: Introduce formula (4) into the output result $<c_i,f(D_{ni})>$, calculate the target formula of each document D_{ni} and introduce formula (7) to calculate the optimal solution set of the document $D_{xn}=\{txt_{x1},doc_{x2},excel_{x3}\cdots\cdots\}$.

5. Experiment and Analysis

This paper chooses the school network center as the cloud server and the surrounding schools are taken as other clouds. The training sample of N is $f_i=(x_i,y_i)_i^m$ and support vector machine adopts $k(x,y)=\exp(-\|x-y\|^2)/\beta$. Besides, the network centers of surrounding schools are extracted based on the book information result to test the effect of detection algorithms.

In order to verify the advantages of the algorithm in the article, simulation experiments is conducted in the designed VC ++ 6.0 environment, and a total of 6 computers have been applied, and the computer configuration is CPU core i32.2G, internal storage is DDR4G, hard disk capacity is 500G, a PC is regarded as server to control database system, and SQL Server2008 will be installed in the rest of the five as the data storage node, system analog data is mainly used in experimental data collection, book data capacity is set to be among [1M, 10M], and comparison should be conducted under such two indicators as access network consumption time and data access time. Carry out comparison between classical and intelligent algorithm and the comparison result is shown in Figure 1 and Figure 2.

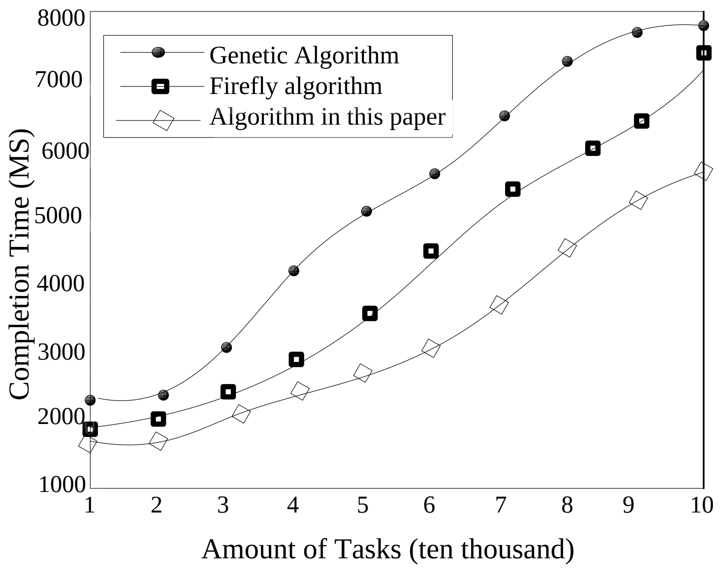

Figure 1. Inquire the Performance Comparison of Algorithm with 1 Thousand Tasks

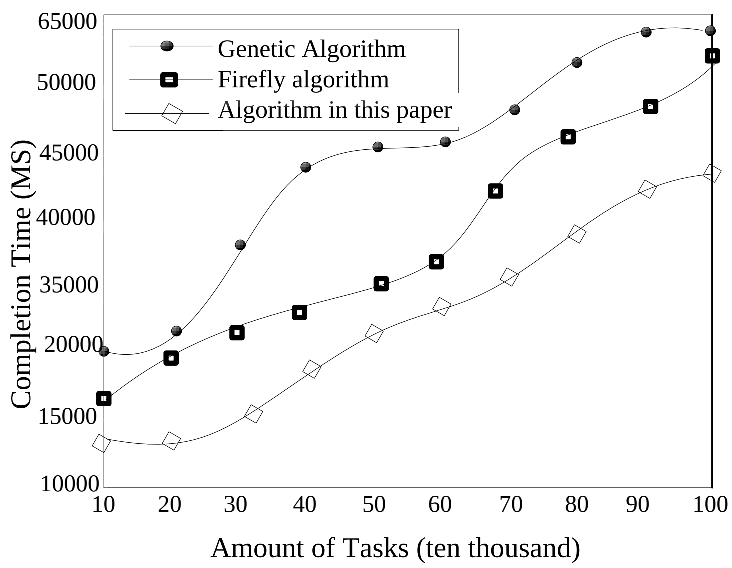

Figure 2. Inquire the Performance Comparison of Algorithm with 1 Million Tasks

From the Figure 1-2 we can find out comparing to traditional intelligent algorithm the algorithm adopted in this paper is superior when there are fewer or more tasks, which elaborates that the initialization in ant colony optimization as well as introduction of pheromone improvement has created good effect and can further decrease the network consumption.

6. Conclusion

How to improve the database accessing efficiency based on cloud computing has always been a research direction of cloud computing and this paper firstly initializes the ant colony optimization and improve the pheromone so as to make the advanced information effectively avoid algorithm local convergence. What's more, the collecting result can be processed via support vector machine and meanwhile the simulation experiment proves that the algorithm adopted in this paper is beneficial to improve the query efficiency of cloud database and provides valuable references for the cloud data research.

References

[1] L. Z. Yu, "Research on Cloud Databases", Journal of Software, vol. 23, no. 5, **(2012)**, pp. 1148-1166.
[2] Z. L. Yun and W. L.- Hai, "Database Query Scheduling Algorithm Based on the 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 ant colony optimization", Computer Engineering and Applications, vol. 46, no. 9, **(2010)**, pp. 10-12.
[4] C. P. Ping, T. D. Ying and L. X. Feng, "Research on scalable cloud relational database", Computer Engineering and Design, vol. 33, no. 7, **(2012)**, pp. 2690-2695.
[5] K. Y. Mei, H. Jiang and W. Guan, "A Tree Cloud Database for SAAS Cloud Service", Telecommunications Science, vol. 28, no. 1, **(2012)**, pp. 37-41.
[6] Q. Hong and M. Qing, "A Load Access Optimization method of Cloud Computer Database Based on ICFV", Wrong Computer Applications and Software, vol. 32, no. 2, **(2015)**, pp. 41-44.
[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.

Fund Project:

This work was supported by the project of the science and technology research of Educational Department of the Hubei provice(D20152601)

Authors

Bin Ning, (1977.07-), Male, Master, Associate Professor, Research Direction: Software engineering, Algorithm design.

Qiong Gu, (1973.03-), Female, Doctor, Associate Professor, Research Direction: Software engineering, Algorithm design.

Yi Wang, (1980.03-), Male, Doctor, Lecturer, Research Direction: Software engineering.

