Library Electronic Resource Scheduling Research based on Ant Colony Algorithm

JunWei Ge [a], Xinxiu Cheng [b], Yiqiu Fang [c]

College of Computer Science and Technology Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[a]gejw@cqupt.edu.cn, [b]783696590@qq.com, [c]fangyq@cqupt.edu.cn

Keywords: library electronic resources, cloud storage, ant colony algorithm.

Abstract. By studying ant colony algorithm, aiming at the library electronic resources storage sharing system, proposed an improved ant colony algorithm based on cloud storage, so that the system allocate tasks can be performed more efficiently, and raise the scheduling performance of the overall architecture. Proved by experiments, this ant colony algorithm can rapidly find the resources that customer needed.

Introduction

Cloud storage combine with virtualization technology can make a number of different types of storage devices in network grouped into a unified system, providing data storage function, and achieve the purpose of library electronic resources share to customers. For the safety and efficiency, the same data file can have multiple copies, when the user need data, he will submit request services to the cloud storage system, according to the algorithm the cloud system will select the optimal data copy and the path to the applicants, this is the cloud storage resource scheduling. Ant colony algorithm is very suitable for solving the problem of resource scheduling in cloud environment, but there are also have slow convergence problem in the early period.

Library Electronic Resource

Facilities of Chongqing University of Posts and Telecommunications digital library status: gigabit backbone network switches, 40 sets of PC server, 1 sets of SAN architecture storage, 8GB optical switches, physical capacity 200TB. This project is based on the VMware virtualization platform, through virtualization technology, creating multiple virtual hosts in a single-homed host, and appointing the operating system and application environment for each virtual host. User don't need to consider the environment deployment, maintenance, purchase and other issues, it can quick to get the desired application environment from the cloud.

VMware sphere virtualization platform provides infrastructure with a server hardware independent, provides users reading, writing service including the local mirror electronic resources of library (e-books, electronic journals, dissertations etc.) and multimedia resources (audio, video, CD etc.) .

In a cloud environment, let n independent sub tasks assigned to the m virtual resource node executing task (m<n). Among them, T= {t1, t2,... tn} represent the task set, VM= {vm1, vm2,... vmm} represent the virtual resource node set, each sub task can only be executed in a virtual resource node [1].In this paper, applied ant colony algorithm to the resource scheduling in cloud environment, to find the optimal mapping for the task, and to make the task be completed in the shortest time.

Ant Colony Algorithm

Ant colony algorithm used to solve the well-known traveling salesman problem. Reference [2] shows that when the number of iterations is large enough, the ant colony algorithm tends to converge in probability to 1 in the global optimal solution.

The total number of ants set for m, d_{ij} means city distance between city i and city j, η_{ij} is the expectations that ants go through from city i to city j; The probability of ant k at time t from node i to node j, calculate formula as：

$$P_{ij}^{k}(t)=\frac{\left[\tau_{ij}(t)\right]^{\alpha}\cdot\left[\eta_{ij}(t)\right]^{\beta}}{\sum\limits_{j\in allowed}\left[\tau_{ij}(t)\right]^{\alpha}\cdot\left[\eta_{ij}(t)\right]^{\beta}} \qquad （1）$$

α determine the relative influence of the pheromone trail. β determine the relative influence of the heuristic information[3], which heuristic function $\eta_{ij}=1/d_{ij}$; $tabu_k$ record cities that ant k has gone through. The total pheromone is determined by the pheromone evaporation rate and the release of pheromone, which satisfied the following formula:

$$\tau_{ij}(t+1)=(1-\rho)\cdot\tau_{ij}(t)+\rho\cdot\Delta\tau_{ij}(t) \qquad （2）$$

Where $\Delta\tau_{ij}(t)=\sum\limits_{k=1}^{m}\Delta\tau_{ij}^{k}(t)$ .The pheromone updating strategy satisfies:

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}Q/L_k & Antstraveledpathin\,the\,k-th\,cycle\\ 0 & else\end{cases} \qquad （3）$$

ρ means pheromone evaporation rate; $\Delta\tau_{ij}(t)$ means pheromone increment this cycle on paths (i, j) ;; $\Delta\tau_{ij}^{k}(t)$ represents ant k in this cycle remain pheromone on the path (i, j) .

Improved ant colony algorithm

According to the principle of ant colony algorithm, we can find that: ant colony algorithm search speed and search quality depend on the pheromone. This paper setting an initial value to pheromone and calculate the distance between each city, it can accelerate the speed of finding the shortest path, and don't need to calculate the probability between each cities with every step. In order to make the algorithm has a better efficiency, among all ants selecting L ants that have better performance and updating pheromone.

1) Set pheromone an initial value and limit the city scope when selecting city. At first, According to the distance each city generate a sort table. In order to find the shortest path, we can give the shortest path more favorable condition, based on sort table, calculate the initial value of pheromone $\tau_0=1/m*\eta_{ij}$, m means the amount of ants. The shorter the distance, the bigger the initial pheromone value, so the city been accessed is more probable.

Reference[4] put forward a conclusion: a map with a complex circumstance and many nodes, during the process of finding the shortest path, ant k in city i choose the next city j, it can't be those cities that far away from city i, the range of city been searched should be in the neighborhood. So use the opinion, we don't need to calculate every two cities, make the current city as the center, according to the city's sort table, selecting the first w nodes as the feasible neighborhood, calculating the probability between two cities, based on the results choose the next city. The value of w depends on the size of city:

$$w=\begin{cases}8 & n<=20\\ 9 & n<=30\\ 13 & n<=50\\ 19 & \mathrm{n}<=100\end{cases} \qquad （4）$$

2) Update pheromone. There are a variety of ways to update pheromone, if for all paths that ants have went through, updating all path's pheromone, it is so easy to cause the results of shock, and the result is not easy to get; if only update the best edge pheromone, that will further strengthen pheromone positive feedback effect of ant colony algorithm, lead to the search process quickly fall into the local optimal solution. In order to solve the above problem, this paper selects the L ants that have better performance, updating pheromone, and add an effecting factor λ_k. The calculate formula as following:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\Delta\tau_{ij}(t) \\ \Delta\tau_{ij}^{k} = \dfrac{Q}{\lambda_K \cdot L_k} \\ \lambda_k = \dfrac{L_k}{L_{k-1}} \end{cases} \tag{5}$$

3) Ant colony algorithm optimization steps. Ant colony optimization algorithm flow chart shown in Figure 1:

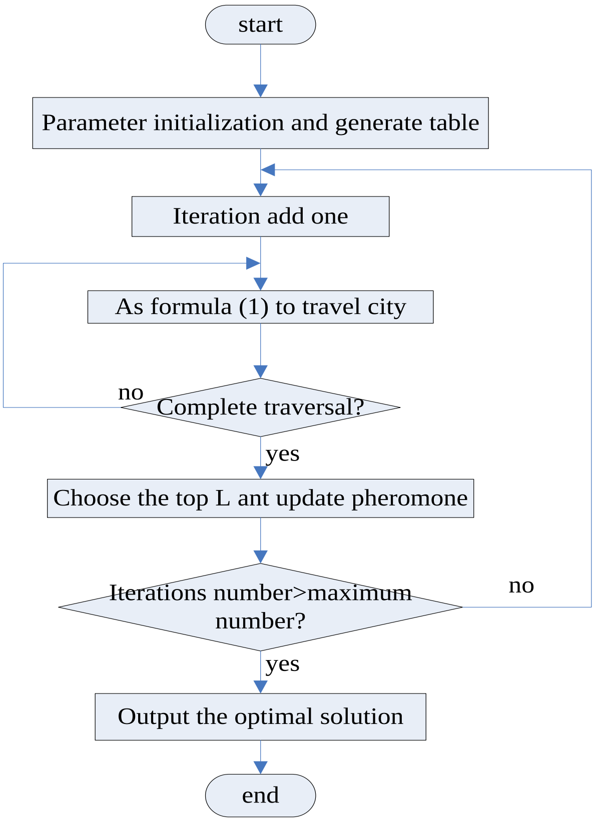
start
Parameter initialization and generate table
Iteration add one
As formula (1) to travel city
no
Complete traversal?
yes
Choose the top L ant update pheromone
Iterations number>maximum number?
no
yes
Output the optimal solution
end

Fig.1 improved ant colony algorithm flow chart

Experiment

In order to test the ant colony optimization algorithm proposed in this paper about feasibility and effectiveness of resource scheduling in cloud storage, we choose cloud simulation platform CloudSim3.0 on the TSPLIB instances Oliver30 simulation test. Sets the maximum number of iterations is 50, the number of ants m is 100, the parameters of the ant colony algorithm is set to: α = 1, β = 2, ρ = 0.5, Q = 5. And compare with the basic algorithm, the improved algorithm of reference[5], analyze the performance of the two algorithms, in order to avoid interference caused errors, run each experiment 5 times, The simulation results shown in Table 1 and Figure.2:

Table 1 and Figure 2 shows the improved algorithm is easier to get the optimal solution compared with traditional ant colony algorithm. We can see the optimal tour value of 424 in the 50th iteration, it decreases rapidly. Meanwhile, the improved algorithm's makespan is less than the other two algorithms with an increase in the task. This is because the new strategy made the pheromone accumulated rapidly in a short time, so it can find the shortest path from ant cave to food source quickly.

The experiment results verify that the pheromone initialization and update strategy proposed in this paper can improve the quality of solution and speed of convergence.

Table 1 Comparing different algorithms in Oliver30 problem

No.	Basic		Reference[5]		Improved	
	Iterations	Length	Iterations	Length	Iterations	Length
1	38	437	45	428	42	426
2	47	439	29	427	30	424
3	49	429	26	433	36	420
4	37	430	30	424	43	421
5	45	438	28	430	26	430
Average		435		428		424

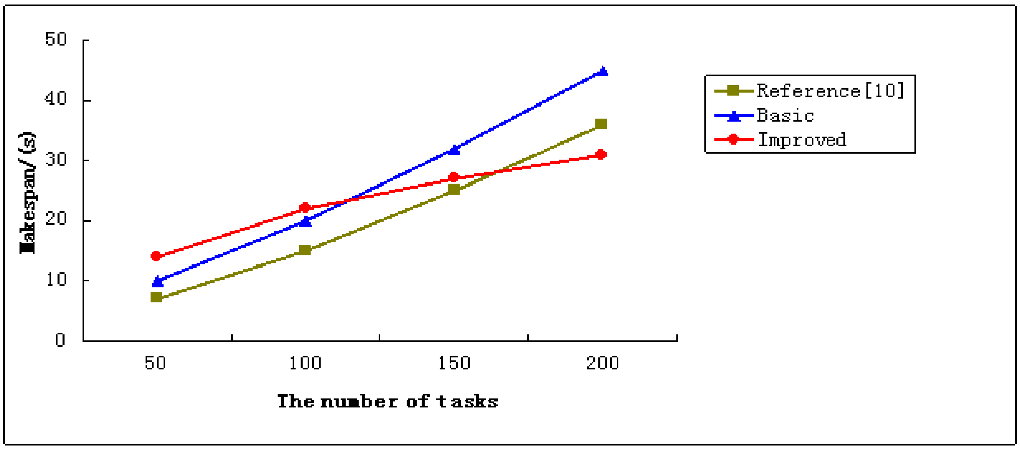

Fig.2 comparing different algorithms about makespan

Conclusion

Based on recent research on ant colony algorithm, and consult the characteristics of the storage of library electronic resources, this paper proposed an improved ant colony algorithm in cloud storage, improved completion time, speed and the optimal solution, Use the distance as the value of initial pheromone, made the shorter path being visited more probable; Narrow the scope of choosing city, reduce the complexity of the algorithm, and reduce the calculation time; At the same time, by improving the strategy of updating pheromone, make the algorithm can get the optimal solution as quickly as possible. Finally, experiments on Cloudsim prove the algorithm is effective.

Acknowledgment

This work is supported by Scientific and Technological Research Program of Chongqing Municipal Education Commission (No.KJ130533).

References

[1] L.D. Liu:Journal of Guangdong University of Education[J],2012,32(5):69-73.

[2] Stutzle.T and Dorigo.M: IEEE Transactions on Evolutionary Computation [J], 2002, 6(4): 358-365.

[3] D.W.Zeng, Q . He and B.Leng: IEEE International Conference on Robotics and Biomimetics, 2012, p2039-2043

[4] Q. B. Zhu and Z.J.Yang: Journal of Software, 2004,15(2):185- 192.

[5] L. Y. Zuo and L.F.Zuo: Journal of Computer Application[J].2012,32(7):1919-1919.