Ant Colony Optimization Through Cloud Computing Techniques

P.Veeralakshmi*, Dr S.Anbu **,
*Associate Professor,*Department of Information Technology,
*Prince Shri Venkateshwara Padmavathy Engineering College

Abstract

How to distribute and coordinate tasks in cloud computing is a challenging issue, in order to get optimal resource utilization and avoid overload. In this paper, we present a novel approach on load balancing via ant colony optimization (ACO), for balancing the workload in a cloud computing platform dynamically. Two strategies, forward-backward ant mechanism and max-min rules, are introduced to quickly find out the candidate nodes for load balancing. We formulate pheromone initialization and pheromone update according to physical resources under the cloud computing environment, including pheromone evaporation, incentive, and punishment rules, *etc.* Combined with task execution prediction, we define the moving probability of ants in two ways, that is, whether the forward ant meets the backward ant, or not, in the neighbor node, with the aim of accelerating searching processes. Simulations illustrate that the proposed strategy can not only provide dynamic load balancing for cloud computing with less searching time, but can also get high network performance under medium and heavily loaded contexts.

Keywords:
Load balancing; cloud computing; ant colony optimization; swarm intelligence

I Introduction

Cloud computing is increasingly being adopted by large businesses, as well as small and medium sized businesses, for “on-demand” and “utility computing”, which holds huge promise for the future of service computing [1]. Virtualization is a key enabling technology for cloud computing environments, which makes it possible to run multiple operating systems and multiple applications on the same
hardware at the same time, so as to provide services by a virtual unit [2]. Through virtualization technology, not only can overall hardware utilization improve and lower costs for disaster recovery, but it can also achieve automatic monitoring for all hosts. However, it is very difficult to assign a large number of tasks to dynamic resources for distributed computing. There are a variety of factors that may lead to some nodes in the overload state while others remain in the underload state, such as uneven allocation of resources, user needs changing over time, newly joining nodes, and a high likelihood of failure in the overload nodes, *etc.* [3–5]. Load balancing is the most effective way to solve the above problem in a cloud computing infrastructure, which ensures that services are delivered transparently regardless of the physical implementation and location within the “cloud”. In recent decades, great progress has been achieved for load balancing, and one of the most promising branches is swarm intelligence algorithms, such as ant colony optimization [6–8], artificial bee colony [9,10], particle swarm optimization [11,12], *etc.* Ant colony optimization, proposed by Marco Dorigo in 1992 [3], is a class of stochastic optimization algorithms based on the actions of an ant colony. By analyzing the previous work of ACO, we found that the ant colony optimization is suitable for load balancing applications in cloud computing because [10–11]:the ant colony is able to crawl among different nodes to search for the optimal solution in cloud computing infrastructure; the ACO is a kind of parallel mechanism that can be applied in distributed computing with high performance; and it is a self-organizing algorithm based on the local information to make judgments and actions, that is, the system does not require a global control center. The motivation of this paper is to establish a load balancing mechanism which utilizes ACO to balance the tasks among nodes in cloud computing. The

targets, including overload or underload nodes, will be quickly identified to operate load balancing by two types of ants with their communications by pheromones. The rest of the paper is structured as follows.

II Related Work

Load balancing plays an essential role in providing quality of service (QoS) guarantees in cloud computing, and it has been generating substantial interest in the research community. There are a great deal of approaches that have coped with the load balancing problem in cloud computing. We discuss the previous related work of load balancing by dividing them into two classes according to the underlying algorithm. The first class consists of diverse conventional approaches without utilizing any kind of swarm intelligence algorithms. Many load balancing approaches were proposed in recent years and each focused on different aspects of algorithms and policies, e.g., leveraging a central load balancing policy for virtual machines [7], the scheduling strategy on load balancing of virtual machine (VM) resources based on genetical algorithms [8], a mapping policy based on multi-resource load balancing for virtual machines [9],adaptive distributed algorithm for virtual machines [2], weighted least-connection strategy [1], and two-phase scheduling algorithms [6]. Additionally, several methods of load balancing were presented for different cloud applications, for example, a service-based model for large scale storage [3], datacenter management architecture [4], and a heterogeneous cloud [5]. Although these contributions have made great progress in load balancing under cloud computing, it has a high degree of centralization and is not easy to extend. Furthermore, these presented approaches did not fully reflect the characteristics of resource nodes and are more suitable to the static situation of cloud computing.

III Dynamic Load Balancing Technique

Master-slave architecture is a mature architecture with a single master server or job tracker and several slave servers, which has been widely used in cloud computing like Google's MapReduce and Hadoop. Figure 1 shows the typical scenario of network topology of virtual resources in cloud computing, which is based on the master-slave architecture and the cloud platform discussed in this paper.

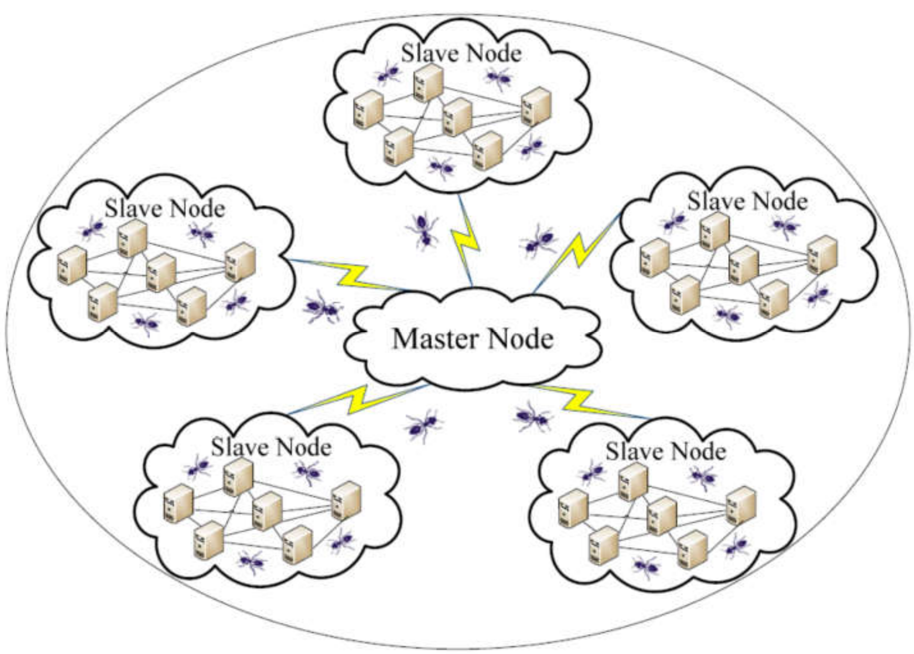

Fig.1 Distributed Nodes in Cloud Computing

In master-slave architecture, a job is firstly submitted to a master node by the user. Then the job is divided into several executable tasks in the master node and the generated tasks are distributed to different slave nodes. After that, the tasks are executed in the slave nodes separately with the guidance of the master node, and the results are returned to the master node. Finally, the distributed results are combined together in the master node and sent to the requesting user. Furthermore, the master node is responsible for monitoring the whole steps and re-executing the failed tasks. During this process, the uneven distribution of tasks may cause that some slave nodes are in light load conditions while others are in heavy load conditions. In this case, load balancing operation ought to be carried out dynamically for the cloud platform in order to keep the platform stable and operating efficiently. By analyzing the cloud computing platform, there are several characteristics in common with ACO, such as cloud nodes being analogous to food locations, data repositories to nests, and load allocation to foraging activity.

IV Forward-Backward ant Mechanism

The ants are divided into two categories: forward ant and backward ant, which is the same mechanism described in [3] but with distinct definition. The forward ant is responsible to find the candidate nodes for load balancing in cloud computing platform and it starts the searching activity from its generated node. The candidate nodes include overload nodes and underload nodes. The backward ant is in charge of updating information pheromones for the path as that of its corresponding forward ant, but in the opposite direction. The backward ant is generated at each time when the forward ant identifies a candidate node. As described in [4], the forward ant calculates the moving probability for each neighbor before it moves and then chooses the largest one as its next destination. To accelerate convergence, we add a special strategy to our model when the forward ant meets the backward ant in the same node, that is, the moving probability is computed by considering both the information pheromone of the node itself and the information pheromone from all backward ants in this node. To simulate the meeting process, we make use of a timer to record the life cycle of a backward ant after it is produced. Some storage units are set for each node and they are used to save the information pheromone carried by backward ants, with one unit for one backward ant. Two types of ants are regarded as meeting each other only when the forward ant arriving at one node before the timer of one backward ant running out in this node. The information pheromone carried by a backward ant will be cleared when the timer reaching zero. When there is more than one backward ant in one node, all the influences by these backward ants should be considered when computing the moving probability of forward ant in this node.

V Results and Discussions

Table 1 gives the experimental results for the number of iterations (Num) and the convergence time(CT) with diverse parameters when reaching load balancing state. For each group of parameters, we carryout the simulation 20 times and the results of Num and CT are the average values. During the process, 50 slave nodes that the task number is bigger than nine are selected as overload nodes and 50 slave nodes that the task number is smaller than two are selected as under load nodes at random. If there are not enough slave nodes that can satisfy such conditions, the actual number of overload nodes and under load nodes are chosen. The forward ants are generated from both the overload nodes and under load nodes in order to accelerate searching process. The load balancing is performed when reaching the maximum searching step.

Table 1. Number of iterations and convergence time with diverse parameters.

NO	μ	ν	θ	ω	σ	α	β	γ	κ	Num	CT
1	0.1	0.1	±0.1	0.3	0.1	1	1	1	1	334	15.432
2	0.1	0.1	±0.1	0.3	0.1	1	1	2	2	319	16.212
3	0.1	0.1	±0.1	0.3	0.1	2	2	1	1	394	20.134
4	0.1	0.1	±0.1	0.3	0.1	2	2	2	2	405	19.293
5	0.2	0.2	±0.2	0.4	0.2	1	1	1	1	315	14.289
6	0.2	0.2	±0.2	0.4	0.2	1	1	2	2	287	12.785
7	0.2	0.2	±0.2	0.4	0.2	2	2	1	1	376	16.112
8	0.2	0.2	±0.2	0.4	0.2	2	2	2	2	389	17.834
9	0.3	0.3	±0.3	0.2	0.3	1	1	1	1	329	14.115
10	0.3	0.3	±0.3	0.2	0.3	1	1	2	2	301	13.454
11	0.3	0.3	±0.3	0.2	0.3	2	2	1	1	392	16.298
12	0.3	0.3	±0.3	0.2	0.3	2	2	2	2	421	21.103

VI Conclusions and Future Work

Cloud computing is a rapidly evolving field and changing the way a business or activity can operate, which provides more flexibility and less time to deploy a project for us. Load balancing is one of the key challenges in cloud computing. In order to optimize resource allocation and ensure quality of service, this paper proposed a novel approach for dynamic load balancing based on the improved ant colony optimization. Two dynamic load balancing strategies were applied with the forward-backward ant mechanism and max-min rules. According to the characteristics of cloud computing, we redefined and improved the ant colony optimization by pheromone initialization and pheromone update. By means of such improvements, the speed for searching candidate nodes in load balancing operations can be greatly accelerated. Two kinds of moving rules for the forward ant were given to update the pheromone so as to speed up the convergence and the detailed dynamic load balancing algorithm was also described. Several simulations were illustrated by the improved approach in a cloud computing platform. The results showed that the proposed approach is feasible and effective on load balancing in cloud computing and also has better performance than a random algorithm and LBVS algorithm.

VII References

1. Wang, L.; von Laszewski, G.; Younge, A.; He, X.; Kunze, M.; Tao, J.; Fu, C. Cloud computing:A perspective study. *New Gener. Comput.* **2010**, *28*, 137–146.
2. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.;Rabkin, A.; Stoica, I.; *et al*. A view of cloud computing. *Commun. ACM* **2010**, *53*, 50–58.
3. Zhang, Q.; Cheng, L.; Boutaba, R. Cloud computing: State-of-the-art and research challenges.*J. Internet Serv. Appl.* **2010**, *1*, 7–18.
4. Lee, Y.C.; Zomaya, A.Y. Energy efficient utilization of resources in cloud computing systems.*J. Supercomput.* **2010**, *60*, 268–280.
5. Rimal, B.P.; Choi, E.; Lumb, I. A taxonomy and survey of cloud computing systems. In Proceedingsof the Fifth International Joint Conference on INC, IMS and IDC, 2009. NCM'09, Seoul, Korea,25–27 August 2009; pp. 44–51.
6. Nishant, K.; Sharma, P.; Krishna, V.; Gupta, C.; Singh, K.P.; Nitin, N.; Rastogi, R. Load Balancingof Nodes in Cloud Using Ant Colony Optimization. In Proceedings of the 2012 UK Sim 14thInternational Conference on Computer Modelling and Simulation (UKSim), Cambridge, UK,28–30 March 2012;

7. Zhang, Z.; Zhang, X. A load balancing mechanism based on ant colony and complex network theoryin open cloud computing federation. In Proceedings of the 2010 2nd International Conference onIndustrial Mechatronics and Automation (ICIMA), Wuhan, China, 30–31 May 2010; Volume 2, pp.240–243.
8. Mishra, R.; Jaiswal, A. Ant colony optimization: A solution of load balancing in cloud. *Int. J. WebSemant. Technol.* 2012, *3*, 33–50.
9. Sesum-Cavic, V.; Kuhn, E. Comparing configurable parameters of swarm intelligence algorithmsfor dynamic load balancing. In Proceedings of the 2010 Fourth IEEE International Conferenceon Self-Adaptive and Self-Organizing Systems Workshop (SASOW), Budapest, Hungary, 27–28September 2010; pp. 42–49.
10. Sesum-Cavic, V.; Kuhn, E. Applying swarm intelligence algorithms for dynamic load balancing toa cloud based call center. In Proceedings of the 2010 4th IEEE International Conference onSelf-Adaptive and Self-Organizing Systems (SASO), Budapest, Hungary, 27 September 2010; pp.255–256.
11. Liu, Z.; Wang, X. A PSO-Based Algorithm for Load Balancing in Virtual Machines of CloudComputing Environment. In *Advances in Swarm Intelligence*; Lecture Notes in Computer Science;Tan, Y., Shi, Y., Ji, Z., Eds.; Springer: Berlin, Germany; Heidelberg, Germany, 2012; Volume 7331,pp. 142–147.
12. Feng, X.; Pan, Y. DPSO Resource Load Balancing in Cloud Computing. *Comput. Eng. Appl.* 2011,*11*, 1–8.