Turkish Online Journal of Qualitative Inquiry (TOJQI)

Volume 12, Issue 6, July, 2021: 5201 - 5209

Implementation of Machine Learning Algorithm for Task Scheduling In Cloud Computing Environment

Amit Kumar Dhingra

Ph.D Scholar

Sushant University Gurugra, Haryana, India

Dr. Dinesh Rai

Assistant Professor,

Department of Engineering & Computer Application

Sushant University Gurugram,(NCR, Haryana,India

Abstract

For each application, efficient scheduling of heterogeneous workloads to heterogeneous processors is critical to achieving high performance. Cloud computing gives a diverse environment in which to conduct numerous tasks. In a cloud context, scheduling user requests (tasks) is an NP-hard optimization problem. To propose a sub-optimal solution to the problem, researchers present a variety of heuristic and metaheuristic strategies. In this research, we present an Ant Colony Optimization (ACO) based task scheduling (ACOTS) algorithm for lowering the average waiting time and optimising the system's makespan. In the CloudSim simulator, the designed method is implemented and simulated. Simulation results are compared to Round Robin and Random algorithms, both of which produce good results.

Keywords: Cloud Computing , Ant Colony Optimization , task scheduling

INTRODUCTION

Cloud computing is a rapidly evolving technology that provides users with online computing resources, storage, and the ability to structure programmes with increased scalability, availability, and fault tolerance. Cloud computing entails storing data on remote servers rather than on personal PCs or other devices. This data can be accessed via the internet[1] from any device, anywhere in the world, as long as the device is capable of supporting cloud computing technologies. A front-end, which is the client side, and a back-end, which is a collection of servers and computers owned by a third party that keeps the data, make up the cloud computing system. To interact amongst networked computers, a central server, which is a part of the back-end, follows protocols and employs middleware. Cloud computing[2] collects and manages all computing resources automatically. 1st It has the following qualities that define a cloud computing system: on-demand self-service, resource pooling, internet connectivity, elasticity in service availability, and metering of services used by individual[3] users. Google Drives has replaced Microsoft Office, Amazon Web Services has replaced traditional enterprise data storage, banking websites have replaced branch offices, and Dropbox now stores all of our data and files. Even multiple deployment and service methods are available in the cloud.

Load Balancing using machine learning Load balancing is a serious concern in cloud computing[2]. With the increase in attractiveness of cloud computing among users, the load on the servers and the quantity of processing done is surging drastically[4]. There are multiple nodes in the cloud, and due to the random allocation of a request made by the client to any node, the nodes become unevenly loaded[5]. So to avoid the condition where some nodes are either severely loaded or under loaded, the load balancer will evenly divide the workload among all the nodes

Related work

Load management is required in distributed computing to spread dynamic native work evenly among all machines. It aids in achieving a high customer satisfaction and resource consumption ratio by ensuring an efficient and equitable deployment of all computing resources. Accurate load balancing reduces resource use, minimises failure, allows for scalability, and avoids bottlenecks, among other things. A comprehensive analysis of existing load balancing methods is provided in this section. This analysis shows that all current approaches are effective, with a focus on lowering related overhead, service response time, and refining performance, among other things. A number of parameters are also identified, and they are used to evaluate the current procedures.

E. Gupta etal[1] The proposed technique of load balancing is based on Ant Colony Optimization which detects overloaded and under loaded servers and thereby performs load balancing operations between identified servers of Data Center. The proposed technique ensures availability, achieves efficient resource utilization, maximizes number of requests handled by cloud and minimizes time required to serve multiple requests. The complexity of proposed algorithm depends on datacenter network architecture.

W. Wen, et al[2] At monitoring stage, it takes both the previous and current system condition into account to avoid unnecessary migrations. Besides, it adopts two different traversing strategies for ants in order to find the near-optimal mapping relationship between virtual machines (VMs) and physical machines (PMs). Experimental results show that ACO-VMM outperforms the existing migration strategies by achieving load balance of whole system, as well as reducing the number of migrations and maintaining the required performance levels.

X. Luo and B. Cheng[3] Task scheduling plays an important role in cloud computing. This paper develops an efficient task scheduling method using biogeography-based optimization (ETS-BBO). Experimental results show that ETS-BBO outperforms ant colony optimization (ACO), Genetic algorithm (GA) and round-robin (RR) algorithms and can solve the joint optimization problem effectively.

Y. Fang et al[4] In order to solve the problem of task scheduling under cloud conditions, a virtual machine real-time state improvement ant colony algorithm (VM-ACO) is proposed. The algorithm takes time into account to complete the load balance of the task. Experiments on the cloudSim platform show that the VM-ACO algorithm performs better in task latency, time to fulfill the task and load balance than the ant colony algorithm based on resource state and polling, and it can realize the excellent scheduling under cloud conditions

M. Kaur et al[5] Cloud computing is really a new computing mode. Load balancing of resources across virtual machines is the fundamental problem of Cloud Computing. Effective job scheduling device must meet people 'requirements and increase the source usage, to be able to increase the entire efficiency of the cloud processing environment. In optimization issue. Genetic Algorithm and Ant Colony Optimization Algorithm have already been referred to as excellent option method. GA is created by adopting the organic progress process, while ACO is encouraged by the foraging behavior of ant species.

M. Mouawad, et al[6] In this paper, we propose a Markov decision process (MDP) based model for RRH selection to serve a connection demand in C-RAN that leads to the user blocking probability minimization as well as the operators' revenue maximization. Moreover, the Ant Colony optimization (ACO) method is used to obtain the best BBU-RRH mapping that leads to further reduction of the user blocking probability and the quality of service (QoS) improvements. The presented simulation results show that the proposed techniques reduce the blocking probability when compared to Received signal strength (RSS) approach.

M. Mouawad, et al.[7] The traffic load is predicted by employing Markov Model based on current users location for each cell. Ant Colony Optimization (ACO) technique is employed to search optimized mapping between BBU and RRH. The balanced traffic load leads to the minimization of blockage connections in the IoT network and hence improves the QoS. The obtained results indicate that the proposed technique ensures the maximization of QoS by minimizing the number of blockage connections as well as handovers which leads to more balanced IoT network traffic.

W. Chen et al[8] Task scheduling is a difficult non-deterministic polynomial problem. Optimization of the scheduling algorithm is the key to improve the efficiency of cloud computing. The traditional meta-heuristic algorithm has slow convergence rate and is easy to fall into local optimal value. This paper proposes a new scheduling method based on a coral reefs algorithm. Firstly, the task scheduling model is formally described. The objective function is proposed to calculate load balancing rate, resource utilization and load balancing stability. Then the representation method of coral reef and the coding scheme of polyps are designed. Matrix random mapping method is applied to improve the variation effect of polyps

Task Scheduling In Cloud Computing Environment using machine learning

The ACO Algorithm in Action[9] The main technique of load balancing with ACO in our approach consists of two steps before load balancing execution, as shown below.

(1) Ant reproduction: Only after scanning the cloud platform on a regular basis to see if there are any overloaded or underloaded[10] nodes are ants formed.

(2) To locate the target node, follow these steps: According to its search rules, the ant is hunting for target nodes in its immediate vicinity that match the load balancing conditions. The target node for load balancing is also known as a candidate node[10]. Maximum and Minimum Limits To trigger the forward ant generation, we construct two unique criteria, known as max-min rules, with the goal of decreasing the time spent searching for candidate nodes, as shown below.

The first rule is the maximum value trigger rule. When the load on a slave node exceeds a given threshold, the node generates a forward ant[12]. It means the node is nearing or above its maximum load, and it needs to distribute work to idle nodes in order to maximise resource utilisation.

The second rule is the minimum value[8] trigger rule. When the load on a slave node is lower than a specific threshold, a forward ant is generated[13]. It indicates that the node is in a light load state, allowing it to accept a variety of new jobs while sharing its resources with the overload nodes. Load Balancing is a procedure for balancing loads. The early steps of load balancing are explained as follows, based on the above strategies[14]. (1) Calculate moving probabilities for all of its neighbours and choose the largest one as its next destination; (2) After moving to a new node, determine if it is a candidate node or not. If this is the case, create a backward ant and initialise it. (3) The backward ant returns to the starting place of its forward ant, as well as the path taken by its front ant in the opposite direction. (4) Calculate the sum resources of the candidate nodes and halt the process if they can meet the load balancing demand; and (5) Perform the load balancing operation. Except for the approach to calculate the moving probability, these procedures are the same for max-min rules.

Proposed Algorithm

Proposed hybrid algorithm for load balancing

Step1. Initialize the pheromone .Set the maximum number of iterations, the pheromone energetic factor , the expected heuristic factor ù, the volatile factors U and U1 , the number of ants m, and p0 .

Step 2. Place all ants at the starting VMs randomly.

Step 3 . Each ant calculates the probability of the current task selected on each virtual machine in the set of optional virtual machines based on the formula. And then the ant chooses the matching VM for the current task according to the roulette method. And then add the selected VM to the taboo table.

Step 4. When an ant completes a solution, update the pheromone on the matching scheme path found by the antaccording to formula. Compare with the previous optimal solution and update the optimal solution.

Step5. If all the ants end their tour, Nc = Nc + 1(Nc is the number of iterations), calculate the global optimal solution and update the pheromone on on the optimal solution path according to formula otherwise, repeat Step3.

Step6. If the current number of iterations is less than the maximum number of iterations, Clear taboo table and return to Step2. Otherwise, end the iteration and output the best solution.

Cloud computing task scheduling[15] can be described as the allocation of n independent tasks assigned to m virtual machine implementation, which according to the optimization objectives to achieve, build a match between tasks and virtual machine to achieve optimal scheduling. In order to simplify the complexity of the scheduling process, make the following assumptions: 1) The tasks are independent of each other and there is no dependency before or after 2) The unit cost of running the task on each resource is known 3) The task is not interrupted during execution Figure 1 shows the process of task scheduling.

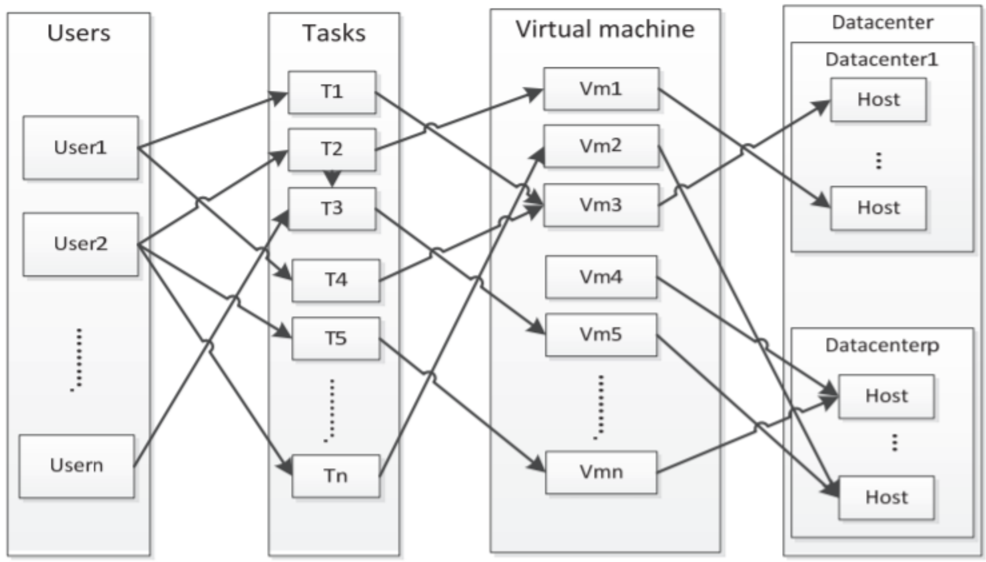

FIGURE 1. Task scheduling process

ACO based solution[16]. Global optimization requirements are key challenges for the scheduler. To deal with these issues, this paper proposed an ACO-based method[17]. On the condition[18] of many SMDs requesting for services at the same time or nearly the same time, the method treats each SMD as a colony. To optimize the overall performance, it introduces the following rules: Rule Ants in one colony only search for one special kind of food. Rule Ants prefer[19]to choose the path with more pheromone of other ants coming from the same colony[16]. Rule Ants prefer to choose the path with less pheromone of other ants coming from different colonies. Rule Ants prefer to choose the path with wider bandwidth, which normally means they can creep faster. Rule The pheromone will decrease by the time but as least last for T s . Rule The pheromone will be nearly zero at the time of Ts . With these rules, original ACO parameters[17] will be set as follows:

The CloudSim toolkit[18] package has been used to perform the simulation of the proposed algorithm. The simulation has been performed for the comparison of following existing scheduling algorithms such as FCFS[19], primary ACO, and with proposed algorithm. A. System Model The following assumptions have been used for simulation. ¬ All tasks are a periodic and homogeneous. ¬ There are no dependencies and precedence constraints between tasks. ¬ All tasks are computationally exhaustive. ¬ All tasks are non-primitive, and they are not interrupted during execution. The goal of this scheduling algorithm is to optimize the total execution time of tasks, and another most important goal is to maintain the balance of load among all VMs. In this work,[20] we consider two factors: first, optimization of tasks execution time and second evenly distribution of workload among all VMs. B. Simulation Environment The following assumption has been made to simulate the proposed algorithm. We used CloudSim toolkit package for experiment and simulation. The experiment has been conducted on Intel® Core™ i3-4005U CPU @ 1.70GHz × 4 machine with 3 GB RAM on Ubuntu 14.04LTS.Table 1 shows the parameters taken to evaluate and simulate the overall performance

Table 1 is represented the data values from various algorithms and its average number of failed cloudlets, average total cost, and average total execution time. Figure 1 shows the relationship between average number of failed cloudlets and algorithm. It clearly determines that which algorithm has less average number of failed cloudlets. As shown, it can be easily seen that Fig. 2 determines that less average which algorithm occupies total cost.

Table 1 Comparison ACO with other existing fault tolerant algorithms

Sr. no.	Algorithm name	Average no. of failed cloudlets	Average total cost	Average total execution time (s)
1	Round Robin	14.33	2843.30	4439.820
2	MCT	14.32	2873.40	4152.128
3	Min–min	19.39	3225.14	5135.2345
4	Proposed ACO	6.52	3345.32	3567.22

Results and Discussion

Figure 2: Graphic user interface

We did an experiment in facility broker policy of cloud analyst, the experiment comprise sorting and subsequent to sorting map function will run to map the user bases through the data center. Service broker policy is the policy by which an algorithm choose to distribute load among the data center. We use optimize response time facility broker policy in which data center prefer according to their response time. We be relevant a sorting in the optimize response time service broker policy and then discover out the results and evaluate with the result which is devoid of sorting .we are using our proposed algorithm for distribution of load

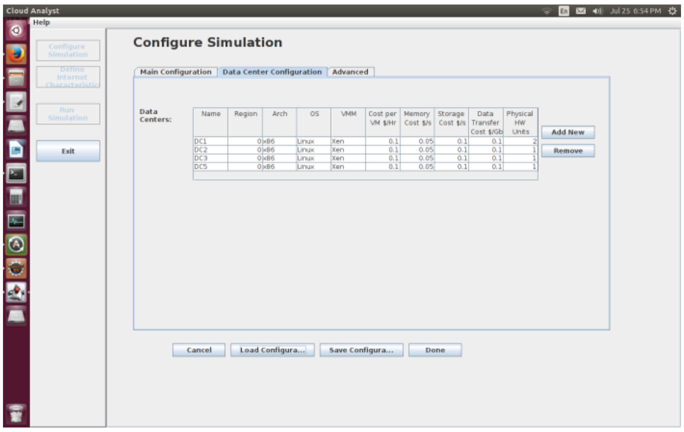

Fig 3 VM allocation using existing and proposed System

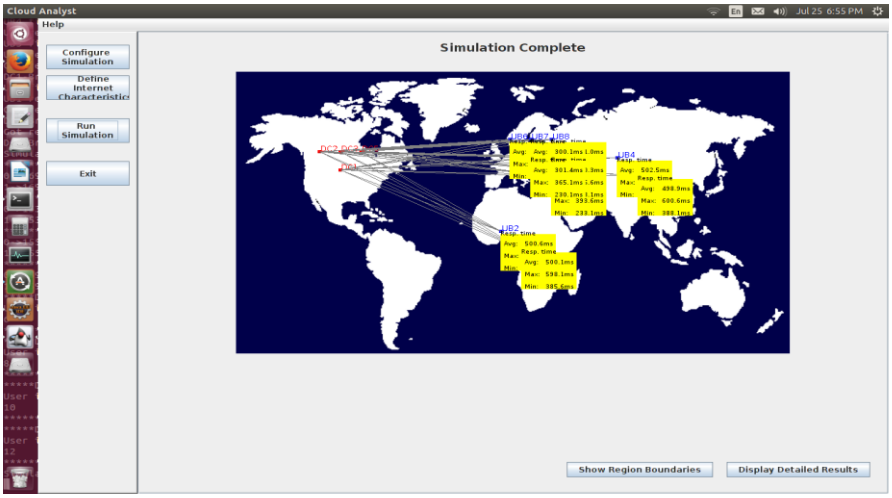

Fig 4 completed the simulation display the results on GUI

Conclusion

We covered cloud computing and load balancing in this paper. Apart from that, we talked about numerous load balancing goals, challenges, components, classification, methodologies, and measurements. Load balancing is a fundamental topic in cloud computing, and its main goal is to meet the needs of users by distributing the load evenly among all cloud servers in order to maximise resource utilisation, enhance throughput, provide fast response times, and reduce energy usage. This research provided a unique approach for dynamic load balancing based on improved ant colony optimization to optimise resource allocation and ensure service quality.

Future work

As far as future scope is concerned, we would like to introduce to minimize the complexity of the ACO-based cloud computing which is not calculated in any paper properly.

Reference

[1]. E. Gupta and V. Deshpande, "A Technique Based on Ant Colony Optimization for Load Balancing in Cloud Data Center," 2014 International Conference on Information Technology, Bhubaneswar, India, 2014, pp. 12-17, doi: 10.1109/ICIT.2014.54.

[2]. W. Wen, C. Wang, D. Wu and Y. Xie, "An ACO-based Scheduling Strategy on Load Balancing in Cloud Computing Environment," 2015 Ninth International Conference on Frontier of Computer Science and Technology, Dalian, China, 2015, pp. 364-369, doi: 10.1109/FCST.2015.41.

[3]. X. Luo and B. Cheng, "Joint Makespan-aware and Load Balance-aware optimization of Task Scheduling in Cloud," 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2019, pp. 547-550, doi: 10.1109/ICSESS47205.2019.9040708.

[4]. Y. Fang and X. Li, "Task Scheduling Strategy for Cloud Computing Based on the Improvement of Ant Colony Algorithm," 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, 2017, pp. 571-574, doi: 10.1109/ICCTEC.2017.00129.

[5]. M. Kaur and M. Agnihotri, "Performance evaluation of hybrid GAACO for task scheduling in cloud computing," 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Noida, 2016, pp. 168-172, doi: 10.1109/IC3I.2016.7917953.

[6]. M. Mouawad, Z. Dziong and K. Addali, "RRH selection and load balancing through Dynamic BBU-RRH Mapping in C-RAN," 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 2019, pp. 1-5, doi: 10.1109/CCECE.2019.8861518.

[7]. M. Mouawad, Z. Dziong and A. El-Ashmawy, "Load Balancing in 5G C-RAN Based on Dynamic BBU-RRH Mapping Supporting IoT Communications," 2018 IEEE Global Conference on Internet of Things (GCIoT), Alexandria, Egypt, 2018, pp. 1-6, doi: 10.1109/GCIoT.2018.8620146.

[8]. W. Chen and h. xu, "Cloud Computing Task Scheduling Method Based on a Coral Reefs Optimization Algorithm," 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 2019, pp. 27-34, doi: 10.1109/ICPADS47876.2019.00013.

[9]. Q. Zheng, J. Li, B. Dong, R. Li, N. Shah and F. Tian, "Multi-objective Optimization Algorithm Based on BBO for Virtual Machine Consolidation Problem," 2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS), Melbourne, VIC, Australia, 2015, pp. 414-421, doi: 10.1109/ICPADS.2015.59.

[10]. K. Sood, A. Jain and A. Verma, "A hybrid task scheduling approach using firefly algorithm and gravitational search algorithm," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 2997-3002, doi: 10.1109/ICECDS.2017.8390005.

[11]. Durgesh, P.&Anand, S.R. (2015). Efficient Throttled Load Balancing Algorithm in Cloud Environment. International Journal of Modern Trends in Engineering and Research,2(3), 463-480.

[12]. A. Michalas and M. Louta, "Adaptive Task Scheduling in Grid Computing Environments," 2009 Fourth International Workshop on Semantic Media Adaptation and Personalization, San Sebastian, Spain, 2009, pp. 115-120, doi: 10.1109/SMAP.2009.25.

[13]. Y. Zhang, J. Wang, L. Gong, J. Yuan and K. Lei, "A Quantified Forwarding Strategy in NDN by Integrating Ant Colony Optimization into MADM," 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou, 2017, pp. 214-220, doi: 10.1109/ISPA/IUCC.2017.00039.

[14]. A. Singh Rajawat and S. Jain, "Fusion Deep Learning Based on Back Propagation Neural Network for Personalization," 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 2020, pp. 1-7, doi: 10.1109/IDEA49133.2020.9170693.

[15]. Kalyampudi, P.S.L., Krishna, P.V., Kuppani, S. et al. A work load prediction strategy for power optimization on cloud based data centre using deep machine learning. Evol. Intel. (2019). https://doi.org/10.1007/s12065-019-00289-4

[16]. Singhal S., Sharma A. (2020) Load Balancing Algorithm in Cloud Computing Using Mutation Based PSO Algorithm. In: Singh M., Gupta P., Tyagi V., Flusser J., Ören T., Valentino G. (eds) Advances in Computing and Data Sciences. ICACDS 2020. Communications in Computer and Information Science, vol 1244. Springer, Singapore. https://doi.org/10.1007/978-981-15-6634-9_21.

[17]. Mousavi S., Mosavi A., Varkonyi-Koczy A.R. (2018) A Load Balancing Algorithm for Resource Allocation in Cloud Computing. In: Luca D., Sirghi L., Costin C. (eds) Recent Advances in Technology Research and Education. INTER-ACADEMIA 2017. Advances in Intelligent Systems and Computing, vol 660. Springer, Cham. https://doi.org/10.1007/978-3-319-67459-9_36

[18]. A. S. Rajawat and A. R. Upadhyay, "Web Personalization Model Using Modified S3VM Algorithm For developing Recommendation Process," 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 2020, pp. 1-6, doi: 10.1109/IDEA49133.2020.9170701.

[19]. Ansar K., Javaid N., Zahid M., Tehreem K., Bano H., Waheed M. (2019) A Hybrid HS-Mean Technique for Efficient Load Balancing in Cloud Computing. In: Barolli L., Leu FY., Enokido T., Chen HC. (eds) Advances on Broadband and Wireless Computing, Communication and Applications. BWCCA 2018. Lecture Notes on Data Engineering and Communications Technologies, vol 25. Springer, Cham. https://doi.org/10.1007/978-3-030-02613-4_4

[20]. Menouer T., Manad O., Cérin C., Darmon P. (2019) Power Efficiency Containers Scheduling Approach Based on Machine Learning Technique for Cloud Computing Environment. In: Esposito C., Hong J., Choo KK. (eds) Pervasive Systems, Algorithms and Networks. I-SPAN 2019. Communications in Computer and Information Science, vol 1080. Springer, Cham. https://doi.org/10.1007/978-3-030-30143-9_16