International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Efficient Task Scheduling for Quality of Service in Cloud Computing Network

T. Manoranjitham, Dupati Srikar

Abstract: ***Task scheduling in cloud is the process of allocating a resource to a task at specific time. The allocation of limited cloud resources to large number of tasks to satisfy the required quality of service is the key challenge in cloud. Allocation of a resource with less capability to a task increases the response time, makespan of the task and waiting time of the entire tasks in the waiting queue. This problem will result to an unsatisfied Quality of Service. In this paper we proposed an efficient task scheduling that uses three threshold values to specify the resource to be allocated to a task at a given time. This method ensures that a capable resource is allocated to task such that the response time and makespan of the all task are minimized. The proposed method was simulated using CloudSim and the result shows a better response time and makespan than the well known Min-Min and Max-Min Method.***

Keywords: Task Scheduling, waiting time, response time, VM utilization, QoS.

I. INTRODUCTION

Cloud computing as a model provides computing resources that allow users to store data and run their applications on a remote resource. The emergence of cloud computing has significantly reduce the cost of infrastructures by providing a pool of shared resources capable of executing users applications concurrently. Resources in cloud are provided in form of a service to clients on demand and the clients pay for the services they consumed. Due to the limited number of shared resources provisioned, clients compete on the resources for data storage, execution of applications and other tasks [1]. Cloud computing uses the concept of virtualization to provide services to multiple users concurrently [2]. An efficient management of cloud resource is required to ensure that the required Quality of Service is satisfied. Resource management in cloud includes resource provisioning and resource scheduling. Resource provisioning identify the adequacy of resources for a particular workload based on the QoS requirements defined by the cloud service consumers and resource scheduling involve mapping and execution of the client task based on the selected resources [3].

Manuscript published on January 30, 2020.
* Correspondence Author
Dupati Srikar*, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India, dupatisrikar12@gmail.com
T. Manoranjitham, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India, manorant@srmist.edu.in

Request for task execution from the users are submitted to the data center broker in the form of workload detail. A resource provisioning agent finds a suitable resource for the given workload and determines the feasibility of provisioning the demanded resource based on QoS requirements [3]. Finally the broker sends a request to the resource scheduler for scheduling of the workload after the provisioning task is completed. The key issue in cloud is how to efficiently allocate a resource to user task to maximize the benefit of the cloud provider while guaranteeing the required Quality of Service [4]. The complexity in scheduling of cloud resources is proportional to the increase in number of request. Inappropriate matching of resource to a task degrades the overall QoS required by the user task [4]. In this paper we focus on appropriate matching of a resource to a task so that the required QoS is provided.

II. LITERATURE REVIEW

J.Praveenchandar et al [5], proposed an algorithm that helps to improve the existing job scheduling algorithm to increase the performance of resource allocation process. Initially all the tasks are sorted according to the size of them. If the size of tasks is same then more priority is given on basis of arrival time. Then mapping of the tasks to the available agent is done so that agent can allocate resources to the task. With this algorithm, the ratio of successfully executing jobs is also increased and also overall response time was improved.

Ajay Thomas et al[6], proposed Delay time algorithm (DTA) is used to maximize the utilization of resources with efficient scheduling strategy. Delay time algorithm is used to analyze, prioritize and schedule the requests based on deadline constraint. This would help in reducing the delay in processing by allocating resources efficiently. Delay time algorithm assumes that all the resources is of same cost but practically it varies. So, giving priority to the cost helps in maintaining efficiency.

Jayachander Subiryala et al [7], proposed a method that deals with security concerns of the user's data. Some major concerns of customers are whether the cloud service provider can reconstruct the deleted data and once customer stops using the services whether all his previous data is deleted or not. In order to ensure the security, a separate module USM (User Shredder Module) is added to the architecture of cloud. USM can interact with data organizer and service provider Mashayekhy et al [8], proposed an online mechanism for auction for virtual machine provisioning but didn't make any assumptions on upcoming demand of VMs i.e. not considering real world scenario.

The proposed auction based online mechanism invokes that whenever new request for resources come or resources released by previously allocated task, each task will be given certain slot and users will continue to use that resources for the entire period.

Himani et al[12], used several parameters to schedule the tasks like profit for provider, task penalty, user loss and task profit. Cost-deadline based task scheduling algorithm used to schedule the task which improves the performance of system and task to be completed within deadline and constraints are to be taken care. This Cost-deadline based scheduling algorithm helps to reduce the overall turnaround time.

Aazam et al[9], proposed a Holistic brokerage model in which brokers manage the service resource utilization and users request simultaneously. This paper concludes that, it would provide an efficient on-demand and future service reservation, pricing for the cloud users. This paper deals with interoperability of multiple clouds, which can also call as inter-cloud computing or cloud federation.

Thiago et al [10], evaluates the impact of imprecise knowledge about the available bandwidth and its impacts on cost estimates and the makespan. it proposes a mechanism in which a deflating factor on the bandwidth value given as input to the scheduler. it deals with impact of uncertainty in the relation to the available bandwidth on the schedules produced by a hybrid cloud scheduler. It also deals with the mechanism to minimize the impact of uncertainties on the schedule generated by the scheduling algorithms. The proposed mechanism showed that it is able to increase the number of solutions with makespans that are shorter than the predefined deadline and minimize the underestimates of the makespan and cost provided by the schedulers.

Jing chen et al[11], proposed a cloud resource allocation method that helps in case of sudden and urgent demands, which can allocate requires resources in optimized and timely manner in case of urgent resource demands. This method helps to rebuild the new priorities for resource allocation in order to allocate resources earlier in case of sudden and urgent demands for virtual machines. A multi-objective optimization model is established, which helps to achieve minimum performance match distance between physical machines and VMs.

III. PROPOSED METHOD

In this chapter we describe the detail of the proposed method as follows: First we create a List of Virtual Machines with different capabilities and sort the list in descending order of capabilities. Second, we partitioned the list into three groups and used the least capability in each list as a threshold to search for a resource within the partition. Three threshold T_1, T_2, T_3 were created form the three partitions. Third, for each user task submitted to the broker we calculate the capacity of virtual machine required by the task as

$$Task_{Capacity} = \frac{Task\_Length}{Task\_Deadline}$$

Finally, we compare the required VM capacity of each task with the threshold values to determine the partition that contains a capable VM to execute the task within its allocated deadline. If there is no capable resource to provide the required QoS, the task fails.

A. VM Partition Algorithm

1. Sort VM's in Order of Capacity
2. Partition VM List into Three
3. Set T_1 = Least VM Capacity in First Partition
4. Set T_2 = Least VM Capacity in Second Partition
5. Set T_3 = Least VM Capacity in Third Partition

B. VM Selection Algorithm

1. For I=1 to Task.Size()
    a. Capacity = capacity of Task[i]
        i. If Capacity ≥ T_1
            1. Search for capable resource in first partition
            2. Assign found resource to Task[i]
        ii. else
        iii. If Capacity ≥ T_2
            1. Search for capable resource in second partition
            2. Assign found resource to Task[i]
        iv. Else
        v. If Capacity ≥ T_3
            1. Search for capable resource in third partition
            2. Assign found resource to Task[i]
        vi. Else
        vii. Insert capacity to List and sort list.
        viii. Return failure

    End for.

IV. EXPERIMENTAL SETUP

To implement the proposed method, CloudSim simulator was used to create cloudlets of different length; deadline unique ID and their respective VM requirements are calculated by dividing the length with the deadline. Series of physical machines with specific configurations were defined in the data center. The virtual machines were created on the physical machines and each VM is identified by a unique ID. The created VM parameters are sorted in descending order of capacity and stored in a list. The VM List was partitioned into three and the threshold variables were initialized to the least capacity of each partition. The cloudlets were bind to the VM's and the bindings were submitted to the data center broker for scheduling. The simulation result shows improvement in performance when compared to Min-Min and Max-Min.

V. RESULT ANALYSIS

Table 1: Comparison of Makespan

No. of Tasks	Max-Min	Proposed	Min-Min
100	380	170.5	205
200	595.5	201	550.1
400	900.1	400.1	795.2
600	1392	600.1	980
800	1620	800	1165
1000	1935.5	1000.5	1343.5

Table 1 represents the comparison of makespan for the proposed method, Min-Min and Max-Min

Table 2: comparison of VM utilization

No. of Tasks	Max-Min	Proposed	Min-Min
100	43	72	55
200	45	79	57
400	51	83	61
600	53	86	65
800	58	91	69
1000	64	94	74

Table 2 represents the comparison of VM utilization for the proposed method, Min-Min and Max-Min.

Table 3: comparison of Response time

No. of Tasks	Max-Min	Proposed	Min-Min
100	15	6	10
200	68.5	25	60.5
400	175	48	151
600	207	58.5	195
800	225.5	79.5	201.5
1000	242.5	95.7	220.7

Table 3 represents the comparison of response time for the proposed method, Min-Min and Max-Min.

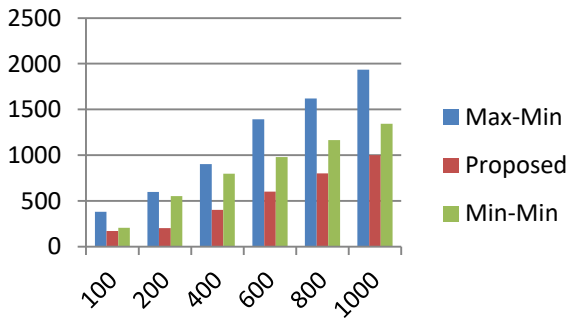

Figure 1: Comparison of Makespan Time

The chart in figure 1 represents the Makespan for the proposed method, Min-Min and Max-Min algorithm

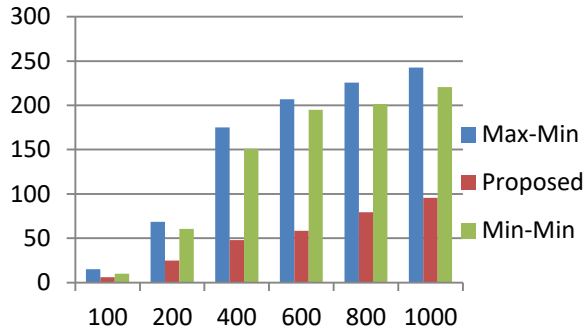

Figure 2: Comparison of Response Time

The chart in figure 2 represents the average Response time for the proposed method, Min-Min and Max-Min algorithm

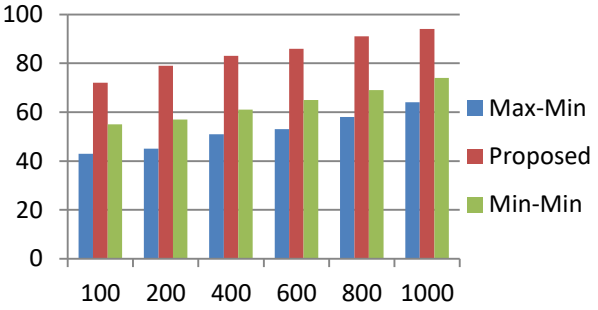

Figure 3: Average VM Utilization.

The chart in figure 3 represents the average VM utilization for the proposed method, Min-Min and Max-Min algorithm

VI. CONCLUSION AND FUTURE WORK

Recently, a number of scheduling techniques have been proposed to minimize waiting time, makespan, execution time and maximize resource utilization. Also, the techniques tend to guarantee QoS through task priority but fail to ensure the assignment of appropriate resource to ensure QoS delivery. In this paper we proposed an efficient scheduling technique that assigns a capable resource to users tasks. The result of the simulation records less response time and makespan when compared with the well known Min-Min and Max-Min method. Our future work will focus on priority assignment to user task and deployment of the method on real cloud model.

REFERENCES

1. A. Y. Gital. Ismail Z. Y., Ilya M. A, and S. Boukari, "An Efficient Grouped task scheduling and resource allocation in cloud computing environments," International Journal of Recent Technology and Engineering, p. 12203-12206, 2019.
2. S. Garg, D. D. Gupta, and R. K. Dwivedi, "Enhanced Active Monitoring Load Balancing algorithm for Virtual Machines in cloud computing," 2016 International Conference System Modeling & Advancement in Research Trends (SMART), 2016.
3. S. Singh and I. Chana, "A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges," Journal of Grid Computing, vol. 14, no. 2, pp. 217–264, Jun. 2016.
4. P. Zhang and M. Zhou, "Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy," IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp. 772–783, 2018.

5. Praveenchandar, J., & Tamilarasi, A. (2018). The Feasible Job Scheduling Algorithm for Efficient Resource allocation Process in Cloud Environment. 2018 International Conference on Recent Trends in Advance Computing (ICRTAC). doi: 10.1109/icrtac.2018.8679241
6. S, A. T., & C, S. (2017). Dynamic resource scheduling using Delay time algorithm in Cloud environment. 2017 2nd International Conference on Computing and Communications Technologies (ICCCT). doi: 10.1109/iccct2.2017.7972238
7. Surbiryala, J., Agrawal, B., & Rong, C. (2018). Improve Security Over Multiple Cloud Service Providers for Resource Allocation. 2018 1st International Conference on Data Intelligence and Security (ICDIS). doi: 10.1109/icdis.2018.00031
8. Mashayekhy, an Online Mechanism for Resource Allocation and Pricing in Clouds, IEEE Transactions on Computers, 12 June 2015.
9. Aazam, Cloud Customer's Historical Record Based Resource Pricing, IEEE Transactions on Parallel and Distributed Systems, 27 August 2015.
10. Genez, T. A. L., Bittencourt, L. F., Fonseca, N. L. S. D., & Madeira, E. R. M. (2014). Refining the estimation of the available bandwidth in inter-cloud links for task scheduling. 2014 IEEE Global Communications Conference. doi: 10.1109/glocom.2014.7036960
11. Chen, J. (2018). A Cloud Resource Allocation Method Supporting Sudden and Urgent Demands. 2018 Sixth International Conference on Advanced Cloud and Big Data (CBD). doi: 10.1109/cbd.2018.00021
12. Himani, Cost-Deadline BasedTask Scheduling in Cloud Computing,Advances in Computing and Communication Engineering (ICACCE),1-2 May 2015.

AUTHORS PROFILE

Dr. T. Manoranjitham is an Assistant Professor (S.G) of Computer Science and Engineering at SRM Institute of Science and Technology. She obtained her PhD in Mobile Ad hoc Networks at SRM Institute of Science and Technology. She has twenty years experience of teaching with over twenty publications. Her area of interest includes Wireless Sensor Networks, Software Defined Networking, Cloud Computing, Internet of Things, Mobile ad hoc Networks

Dupati Srikar is a PG Student of Computer Science and Engineering at SRM Institute of Science and Technology, Chennai. He Completed his B.Tech in Computer Science and Engineering at Sreenidhi Engineering College, Gatkesar, Hyderabad. His area of interest include Cloud Computing, Data Communication and Networks, Machine Learning, Internet of Things and Software Engineering.