Maximizing Resource Efficiency using Task Scheduling in Cloud

Harshitha N*[*1], Devi T[2], Harshitha S[1], Vaishnavi N[1], Kambalapalle Sravani[1]
[1]UG Students, [2]Assistant Professor
[1,2]Department of Computer Science and Engineering, Rajarajeswari College of Engineering,
Bengaluru, Karnataka, India
***Email:** harshitha8298@gmail.com*

***DOI:** http://doi.org/10.5281/zenodo.3270575*

Abstract

Considering the growing use of cloud computing and the want for best use of resources within the cloud, and attention to users that get services they use supported their pay-as-you-go basis, there ought to be a faster approach for users to decrease the user's waiting time and task's waiting time. The main purpose of this paper is to provide comparison between various task scheduling algorithms hence gives better understanding as to which algorithms aids better in maximizing resource efficiency and also enhance security by giving various layers of security. There are many scheduling algorithms within the world. In this paper, we have selected a few better working algorithms along the traditional max-min and min-min algorithms and also the security is enhanced using ECC encryption.

***Keywords:** Cloud computing, ECC, encryption*

INTRODUCTION

Due to the increasing development of storage and processing technology, moreover because the success of the web, the cost-effectiveness of resource process and convenience, this technology has junction rectifier to the emergence of a brand new computing model known as cloud computing. During which resources and applications are free once the user's use and is shared with other users. In fact, cloud computing creates a versatile environment for providing resources [1] and supported massive computer networks. Cloud computing has lots of resources available for users in numerous ways that, like virtualization, Internet protocols, etc. Cloud computing hides its technical details from the attitude of users, and putting layers between these details and users. [2]

If multiple users will do a range of tasks within the cloud, cloud computing uses planning algorithms to grasp that user to figure on, or that user is most imperative. Scheduling algorithms create it doable to take advantage of resources. By mistreatment cloud computing, a user will use the services and pool of resources through the net. Cloud computing has revolutionized the IT business by mistreatment virtualization, service-oriented design (SOA), and computing tools. [3]

Task scheduling may be a method during which a specific resource is appointed to a task at such time. The most purpose of the planning is to maximize resource potency and its main purpose is to scale back waiting times. The scheduling of tasks in cloud computing is one in every of the foremost difficult tasks. "A good scheduling will increase the system's potency." [4]

In second part of this paper, we study the connected work. In third part, we study proposed work and in the fourth part, we study analysis and result.

RELATED WORK

In associate rule, two parameters of job length and priority jobs are thought-about.

If the rule wish to run primarily based on the length of the roles, it initial arranges the tasks so as of length of jobs, then calculates the common length of the roles, and then calculates the distinction within the length of every jobs with the mean. This rule selects tasks from the centre and gives credit to every tasks, and queues for execution. If you want to run per the priority, per the priority given by the user to the task, credit is applied and also the task’s square measure run on the premise of their validity. [5]

In a dynamic resource programming formula, a time delay algorithm is projected within which the calculation of the arrival time of missive of invitation for numerous tasks is taken into account as a basic concept. Once it involves programming tasks, the work that has more work (execution time) is at first set. If multiple work loads are an equivalent, they'll be indiscriminately chosen. That is, if a little job is entered, it'll be delayed. Therefore, the delay formula analyzes and assigns requests consistent with the priority and point. This formula helps to allot resources to users at once in process. [6]

A rule is projected for prioritizing tasks, which are arranged within the initial stage per the time of arrival. In the second step, the tasks area unit prioritized by the point in time and placed within the prepared queue. The task that takes priority is performed earlier. If when the queue type, a brand new priority task is added to the queue, it'll be placed at the start of the queue andTherefore, the queue is reordered. In fact, they use the EDF (Earliest Deadline First) rule. If one thing goes in the middle of another task and gets a better priority, the queue will return to the queue and therefore the queue are organized according to the priority, and what is running is that the resource at the end of the roles. [7]

An advanced max-min scheduling algorithm is proposed for cloud computing. Which is a correction in the advanced Max-min algorithm. First, it obtains an average runtime from all available jobs, then selects the closest run-time distance to the average number. Sometimes the biggest job is too big and it causes a system imbalance. Therefore, this method always chooses a task whose length or execution time is closer to the average length or the time it runs all the tasks. This method reduces Makespane and performs load balancing over the Max- min method.[8]

In the strategy planning of algorithm, PPIA (Priority Based Performance Improved Algorithm) is proposed that takes into account the priority of the user. Users who regularly use cloud services can experience better service. Of course, the top priority is given to normal user tasks. Other tasks are normal priority tasks. The Meta function set must be divided into two groups for important tasks and ordinary priority tasks. In the first step, the tasks of the task group with a high priority should be scheduled using the Min-Min algorithm. Then, the normal priority tasks must be scheduled using the Max-Min algorithm. This algorithm generates the average time for priority tasks as the Min-Min algorithm. Compared to the Min-min algorithm, Makespane reduces it and increases the utilization rate of resources. [9]

In the other algorithm, the three algorithms RR (Round Robin), SJF (Shortest Job First), and Priority are used, with all processes that is in priority queues. The priority range is low, medium to high. Regarding the RR algorithm, a fair quantum is given to all processes. Therefore, low priority processes are less time-consuming than initial processes, and moderate priority processes have less time than high priority processes. Given the RR algorithm, every job has a time slice for using the processor. Quantum time should not be low because it creates a large number of switches. It should not be too high because it causes starvation. In the SJF algorithm, a process that is less time-consuming in the queue takes the CPU first. Although this algorithm has the best ability, it does cause the processes to go starvation at a longer time. The priority algorithm also places the CPU on top priority tasks, and if the two jobs have the same priority, then use the First Come First Service (FCFS) algorithm.[10]

Regarding the defects of the two popular Min-Min and Max-Min algorithms, it is proposed to overcome these algorithmic errors, first before processing the ETC matrix (Expected Time to Compute) using Min-Maxnormalization to scale the values in the range Characterized. After normalizing, the new values of the NETC are matrix between 0 and 1, which is most used to select tasks. After normalizing, the last completion time and the first time you finish each task are found on existing virtual machines. The task that takes maximum time difference and planned on the virtual machine. Here, the desirable virtual machine for each task is defined as the virtual machine, which takes the minimum time to perform this task. [11]

In First come first serve algorithm, given in processes with their burst times, the task is to search out average waiting time and average rotate time mistreatment FCFS programming rule. First in first out (FIFO), conjointly referred to as 1st come back, 1st served (FCFS), is that the simplest programming rule. FIFO merely queues processes within the order that they arrive within the prepared queue. In this, method the method that comes 1st are going to be dead 1st and next process starts solely when the previous gets absolutely dead. Here we have a tendency to area unit considering that point for all processes is zero. [12]

With the aim of improving the SJF scheduling algorithm, an algorithmic cloud computing algorithm is proposed to obtain the best result of scheduling, lower execution time, and decreasing Makespane. The main modified SJF main idea is to sort out tasks in ascending order based on the length of jobs and calculating the average of the total length of the tasks. Then the algorithm checks for the length of all jobs, if it is less than the average job length, and the number of tasks in VM1 is less than the number of tasks in VM2, then the job is sent to VM1 instead of sent to VM2. In this paper, the SJF job scheduling algorithm is modified to achieve the scheduled tasks with the minimum Makespane. It focuses not only on the completion time of the jobs, but also on the completion of all tasks. We demonstrated with empirical evidence that MSJF is better than SJF and FCFS. Experimental results showed that MSJF (Modified Shortest Job First) was more efficient in improving response time and Makespane compared to SJF and FCFS. [13]

The algorithm was presented with Priority Aware Longest Job First (PA-LJF). In those users, they send things through the web interface. The tasks are with an assistant receiving the request. The user who sent the job is authenticated if the job sent from him is sent to the priority queue, otherwise the user will be known as the attacker and his job will not be performed. After confirming the user, the tasks are arranged by their size and descending order, and the virtual machine that can do it faster will take the job. This is done by the resource regulator, which has many scheduling algorithms. Including min-min, PSO, ACO, greedy resource allocation, honey bee, round robin and more. Each of these algorithms has advantages and disadvantages. Therefore, according to the user's need or choice, the scheduling algorithm is selected. After the end of the task, the algorithm's performance is minimized,minimum run-time, high efficiency, response time, bandwidth used and user satisfaction is checked. [14]

In optimization algorithms, optimization can be used to improve scheduling parameters such as cost, energy, resource usage, rotational time, and runtime. An algorithmic resource was proposed to reduce Makespane and increase resource productivity. In this algorithm, it provides scheduling for a large number of independent tasks to examine the performance of the Makespane and CPU parameters based on the particle optimization integration and the Max-Min algorithm. Therefore, this algorithm combines the advantages of both algorithms with good accuracy. The traditional PSO algorithm takes a lot of time to find the optimal solution. But by integrating with the Max-Min algorithm, it outputs for possible sequences of tasks that can be assigned to a virtual machine. This output is then given as input to the Particle Swarm Optimization (PSO) algorithm. As a result, it helps to find the optimal solution sooner than the traditional one. Experimental results show the positive performance of this algorithm with other algorithms. [15]

PROPOSED WORK

In this paper, we use the following algorithms for comparison to find as to which algorithm performs better.

- MAX-MIN

Time taken by all the tasks is computed and the one with the maximum time is paired along one that takes minimum time. So, simultaneously many tasks perform together.

- MIN-MIN

Time a task takes is computed for all the tasks. The task that takes minimum time is first computed and so on.

- DAG

A rooted tree is one type of Directed AcyclicGraph and a DAG is one of directed graph. A DAG is used to represent the most common sub-expressions in an optimizing compiler.

- FCFS

Given n processes with their burst times, the task is to seek out average waiting time and turnaround time by using FCFS scheduling algorithm. 1st in, 1st out (FIFO), is the simplest planning. It merely queues processes within the order that they arrive and executes it and later goes to next process.

- MCT

  Monte Carlo Tree gives the most promising moves based on random sampling.

- DHEFT

It involves heterogeneous environments hence the name Deadline Heterogeneous Earliest Finish Time. It works in 2 phases, one being set the priority to each task and select tasksbased on them and second being load the task on to a processor.

The time taken for execution is recorded and the following is computed for the above given algorithms.

1. Turnaround time = Task Completion time – Arrival Time
2. Waiting time = Turn Around Time – Burst Timeanda graph is plotted to compare the performance of each algorithm.

To enhance the security as we are performing tasks in cloud, ECC algorithm is used for encryption.

ECC: Elliptic-curve cryptography is an approach to public-key cryptography which is based on the algebraic structure of elliptic curves over finite number of fields. ECC needs smaller keys than that compared to non-EC cryptography to provide equivalent security. Hence the documents in Cloud gets secured with ECC.

PERFORMANCE RESULT

In this part, we check for the comparison between the algorithms incorporated. Below given is a graph drawn.

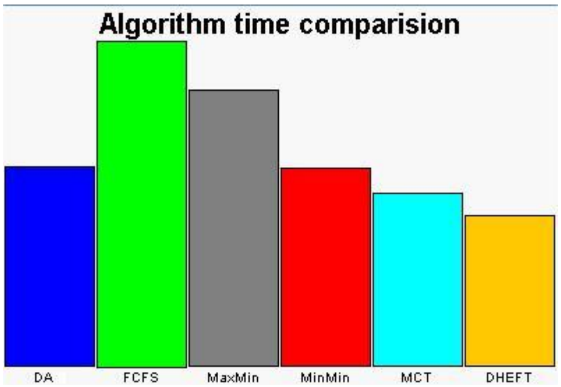

Figure 1: *Graph.*

The time taken by the various algorithms is plotted which gives us better understanding as to which algorithm perform betterand also by encrypting the documents the documents are encrypted stored in parts as to how a document was split in task scheduling so that the security is enhanced as the documents are stored in unreadable format and that too in bits makes it more complex for an hacker to break the code.

CONCLUSION

In this paper, the main aim is to compare various task Scheduling algorithm, which provides better understanding about the best algorithm that aids in optimizing resource efficiency, which helps in reducingthe user's waiting time and task's waiting time. Here thesecurity is enhanced by providing various layers of security. We are selecting few of the best algorithms which holds good for prioritizing the tasks in additionwith min-min and max-min algorithms. In this algorithm, priority is firstly selected for tasks based on a prioritization algorithm, and then using the median number to decide which one of the Min-Min or Max-Min algorithms is to be used. ECC encryption is used for enhancing the security in the project.

REFERENCES

1. Amalarethinam, D.I.G.,S.Kavitha (2017), "Priority based PerformanceImproved Algorithm for Meta-task Scheduling in CloudEnvironment",*SecondInternationalConference OnComputing andCommunications Technologies(ICCCT'17), IEEE.*
2. Ajay Thomas S., Santhiya. C (2017), "Dynamic Resource Scheduling using Delay Time Algorithm in Cloud Environment", *Second International Conference On Computing and Communications Technologies(ICCCT'17) IEEE.*
3. Wadhonkar, A., D. Theng(2012),"A survey on different schedulingalgorithms in cloud computing", *InternationalConference onAdvances in Electrical,Electronics, Information,Communication and Bio-Informatics(AEEICB16)*,2016Ghanbari , Shamsollah, and Mohamed Othman. "A priority based job scheduling algorithm in cloud computing",*Procedia Engineering*, Volume 50, pp.778–785.
4. Pankajdeep Kaur, Parampreet Singh, "Priority basedScheduling Algorithm with Fast Task Completion Rate in Cloud.
5. Xiaofang Li, Yingchi Mao, Xianjian Xiao, YanbinZhuang (2014),"An Improved Max-Min Task-Scheduling Algorithmfor Elastic Cloud" *IEEE International Symposium on Computer, Consumer and Control.*
6. Er. Rajeev Mangla, Er. Harpreet Singh, (Feb. 2015), "Recovery and User Priority Based Load Balancing in Cloud Computing", *International*

Journal Of Engineering Sciences &Research Technology, ISSN: 2277-9655.

7. Dr. M. Dakshayini, Dr. H. S. Guruprasad (October 2011), "An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment",*International Journal of Computer Applications.*
8. Jianguang D (2014),"Research on Task Scheduling Strategy of Cloud Computing", South China University of Technology.
9. Gupta, G., et al. (2014), "A Simulation of Priority Based Earliest Deadline First Scheduling for Cloud Computing System", *First International Conference on Networks & Soft Computing, IEEE.*
10. J. Tsai, J. Fang, J. Chou (2013), "Optimized taskscheduling and resource allocation on cloud computing environment using improved differential evolution algorithm", *Computers & Operations Research,*Volume 40, Issue 12, pp. 3045−3055.

Cite this article as: *Harshitha N, Devi T, Harshitha S, Vaishnavi N, & Kambalapalle Sravani. (2019). Maximizing Resource Efficiency using Task Scheduling in Cloud. Journal of Data Mining and Management, 4(2), 15–20. http://doi.org/10.5281/zenodo.3270575*