A Task Scheduling Algorithm Based on Task Length and Deadline in Cloud Computing

Arnav Wadhonkar, Deepti Theng

Abstract- The cloud computing is the bunch of computing resources which are delivered as a service to the customer or multiple tenants over the internet. The task scheduling is the very important part of a cloud computing. The task scheduling mainly focuses on enhancing the efficient utilization of resources and hence reduction in task completion time. Task scheduling is used to distribute specific tasks to certain resources at a particular time. Different approaches have been presented to overcome the problems of task scheduling. Task scheduling improves the efficient utilization of resource and yields less response time. Task scheduling helps to reduce the completion time of the tasks.

The scheduling algorithm is presented in this paper, which schedules the tasks based on their length and deadline. Results are compared with traditional algorithms and comparative analysis shows a reduction in makespan and average waiting time.

Keywords- Average Waiting Time, Cloud Computing, Makespan, Scheduling Algorithms, Scheduling Criteria, Task Deadline, Task Scheduling.

1 INTRODUCTION

THE cloud computing is simply described as the computing services and resources delivered to the user over the internet. Cloud computing is nothing but the accessing the pooled resources required for computing through your browser's window. In [1] cloud computing is defined as computing services which are served on the general internet protocols in distributed network to the user. Cloud computing is a pay-per-use service rather than a product. And who provides this kind of services are called as cloud service providers. For Example Google, Amazon, Microsoft are the cloud service providers.

In IT industries, the word 'cloud' is commonly used in some networks which setup is not commonly known to us. For example, Internet Service Provider's (ISP) network. The user of the internet is not known to the network setup, but they use it easily. Generally, in networking diagrams, ISP's networks are represented by cloud symbol. So the computing, which is done over the internet without knowing the resource's location can be called as cloud computing.

1.1 Scheduling in Cloud.

The task scheduling is can be described as the procedure of allocating the resources to the particular job in specific time. The most important objective of scheduling is to increase the usage of resources. Minimizing the waiting time is the goal of scheduling. A good scheduling algorithm yields good system performance. In the cloud, there are numerous and distinct resources available. The cost of performing tasks in cloud depends on which resources are being used so the scheduling in a cloud is different from the traditional scheduling. In a cloud computing environment, task scheduling is a biggest and challenging issue. Task scheduling problem is the NP-Complete problem. Many heuristic scheduling algorithms have been presented, but more improvement is needed to make the system faster and more responsive.

The traditional scheduling algorithms like Max-Min, Round Robin, Min-Min, First Come FCFS, SJF are not a better solution to scheduling problems with cloud computing. So we need the better solution to this heuristic problem. In cloud computing, a lot of tasks scheduling algorithms are available, which are presented by various authors. The goal of this paper is to present algorithm in cloud computing which is revised.

- Scheduling procedure in cloud [2] is given roughly in three phases which are as follows
- Resource recognizing and classifying – The resources are recognized by datacenter broker which resides in a system and it also gathers data about the status of resources.
- Choosing resource – The desired resource is chosen as per particular criteria of resource and task. This is an important phase.
- Task capitulation - Resource is allocated to the chosen task.

- *Arnav Wadhonkar is currently pursuing Master of Technology in Computer Science and Engineering at GHRCE, Nagpur, India. E-mail: arnavpw@gmail.com*
- *Deepti Theng is currently working as Assistant Professor in Department of Computer Science and Engineering at GHRCE, Nagpur, India. E-mail: deepti.theng@gmail.com*

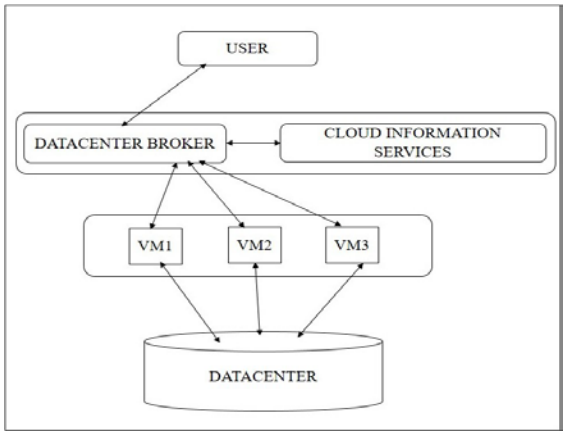

Fig. 1: Scheduling Procedure in Cloud [2]

1.2 Types of Scheduling

The scheduling is distinguished as Static Scheduling and Dynamic Scheduling.

Static Scheduling: In Static Scheduling all information are known to scheduler about tasks and resources before execution. It has less runtime overhead. [3]

Dynamic Scheduling: In Dynamic Scheduling information about task components is not known before execution. Task execution time may not be known. It has more runtime overhead. [3]

2 Related Work

Various scheduling algorithms are available in cloud computing. Two main criteria are taken into consideration for scheduling are task length and deadline in the presented approach. There are some other parameters which tempt the utilization of resources and scheduling of tasks.

Improved Cost Based Algorithm [4]: This algorithm ameliorates the traditional cost-based scheduling algorithm for making appropriate resource allocation. The tasks are grouped as per processing power of resources.

Earliest Feasible Deadline First [5]: The goal of this algorithm to minimize the time complexity of EDF algorithm. This algorithm takes deadline as scheduling criteria. The process migration is reduced to minimize the complexity of the algorithm. The real-time application can be possible if algorithm improved further.

A Priority based Job Scheduling Algorithm [6]: The approach is presented here, is for job scheduling by using mathematical calculations. In this algorithm for scheduling the priority is considered and each job request for resources with some priority. This paper discusses problems linked to the algorithm for example consistency, complexity as well as makespan. As per author, makespan can be reduced further by improving the algorithm.

A Scheduling Algorithm based on Priority for VM Allocation [7]: The objective of the presented algorithm is to provide more advantage to the service provider. Since the present resources are not adequate to practice all the requests. The paper proposes a priority algorithm to find the best choice. This technique can increase the benefits than applying typical FCFS strategy. If more information can be made available, e.g. the regular pattern of the usage the algorithm can be improved further.

Greedy Based Job Scheduling Algorithm [8]: This algorithm focuses on QoS, as the cloud computing is a business-oriented service. The goal of the algorithm is to reduce completion time and to give a faster solution to a scheduling problem. This algorithm classifies tasks based on QoS and then as per the task category, the appropriate function is assigned. The results of the algorithm are compared with another existing algorithm that are, the algorithm based on Berger model and existing scheduling strategy of CloudSim tool.

Priority Based Earliest Deadline First Scheduling Algorithm [9]: In this method there is two task scheduling algorithms are used, one is Earliest Deadline First and other is priority-based scheduling algorithm. This algorithm focuses on resource allocation and memory utilization. This algorithm will minimize the completion time of preempted jobs and it will improve the efficiency of scheduling. This algorithm overcomes the waiting time problem of preempted tasks. The waiting queue is introduced which processes the preempted tasks.

Credit Based Scheduling Algorithm [10]: This algorithm has two scheduling parameters that are user priority and task length. This is a credit based algorithm, which means credits are calculated on the basis of priority and the length. And finally it decides the sequence of tasks by sorting that credits in descending order. That means the first task which will be scheduled is having the highest credit value. This algorithm works more efficiently than previous methods.

3 Presented Approach

The presented algorithm has two scheduling criteria first one is task length and the second one is task's deadline. The algorithm is deployed as scheduling policy of broker.

The following figure explains the workflow of the project. First up all cloud environment is simulated using CloudSim [11] simulator in NetBeans IDE 8.0.2. In next phase, an algorithm based on task length is implemented. And the analysis of result is carried out. And in the next phase, an algorithm based on task's deadline is implemented and analysis of result is done. And in the last phase presented algorithm is implemented and final comparative analysis of results are done.

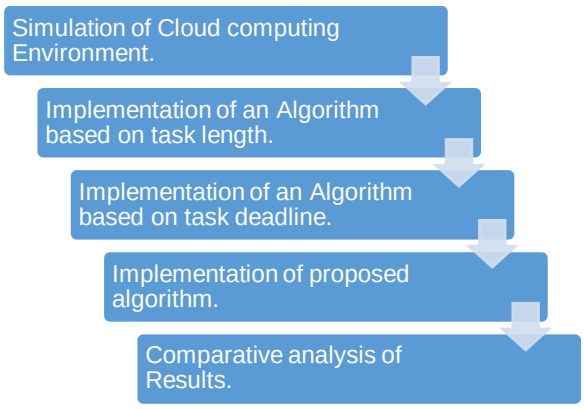

Fig 2: Workflow Diagram

The following table shows the symbols and their meanings in the algorithm.

TABLE 1
SYMBOLS AND THEIR MEANINGS.

Symbol	Meaning
len	List of cloudlet length
dl	List of deadline of cloudlet
lenavg	Average of length.
dlavg	Average of the deadline.
len_d	List of cloudlet length difference
dl_d	List of cloudlet deadline difference
len_div	List of the division of cloudlet lengths.
dl_div	List of division on cloudlet deadline.
n_cloudlet	Number of cloudlets
T	List of final sorted cloudlets.

The very first step in the algorithm is to get length and deadline of cloudlet then it find the average of the length and average of the deadline of all cloudlets and the lists are sorted in ascending order. And then the difference is calculated for each cloudlet from the average and the length of each cloudlet and same is done for each deadline. Here the average difference is calculated to choose the cloudlets for scheduling from the middle of the list not from the very beginning or the last of the list. Then the both lists are sorted and division is performed on each cloudlet. And then the addition is performed on both lists by index number. That leads to proper scheduling of cloudlets and no starvation for smallest or biggest cloudlet. The detailed algorithm is given below.

3.1 Algorithm

Step 1. Get the length of each task. (leni)
Step 2. Get the deadline of each task. (dli)
Step 3. Calculate the average of task length (lenavg) and an average of task deadline (dlavg).
Step 4. Sort len and dl in ascending order.
Step 5. Calculate Task Length Difference (len_d) for each task length and each task deadline (dl_d)

len_di = | lenavg – leni |

Same for task deadline

dl_di = | dlavg – dli |

Step 6. Sort len_d and dl_d in ascending order.
Step 7. For each task length difference (len_di) divide each (len_di) by the number of cloudlets and after every successive division decrement number of cloudlet by1.

For Each len_di {

len_divi = len_di/ n_cloudlet

cloudlet - - }

Same step will be followed for dl_d for each task deadline difference.

For Each dl_di {

dl_divi = dl_di / n_cloudlet

cloudlet - - }

Step 8. For each task T.

T_i = len_divi + dl_divi

Addition is not done one to one. It's done by index numbers.
Step 9. Finally sort the T in ascending order.
Step 10. The first task in sorted order will be sent on highest configuration virtual machine and second on second large VM and continues for all remaining cloudlets.

4 RESULTS AND DISCUSSION

The presented approach is tested in the simulated cloud environment. For simulation "CloudSim" [11] simulator is used. The comparison of completion time and average waiting time of tasks is done with two algorithms where the scheduling criterion of the first algorithm is only task length and second algorithm's scheduling criterion is only task's deadline. The results of the presented algorithm are improved. The following table shows the configuration of simulation of the cloud.

TABLE 2
THE BASIC CONFIGURATION OF SIMULATED CLOUD

No. of Datacenter	1
No. of Cloudlet	10
No. of Broker	1
No. of Virtual Machines	3

The specific configuration of Datacenter is given in next table.

TABLE 3
DATACENTER CONFIGURATION

Architecture	x86
RAM (MB)	4096
Hypervisor	Xen
Storage (MB)	10000
MIPS	15000
Bandwidth (MBps)	10000

The following figure shows the simulation of the datacenter in a cloud environment. And shows the Datacenters configuration.

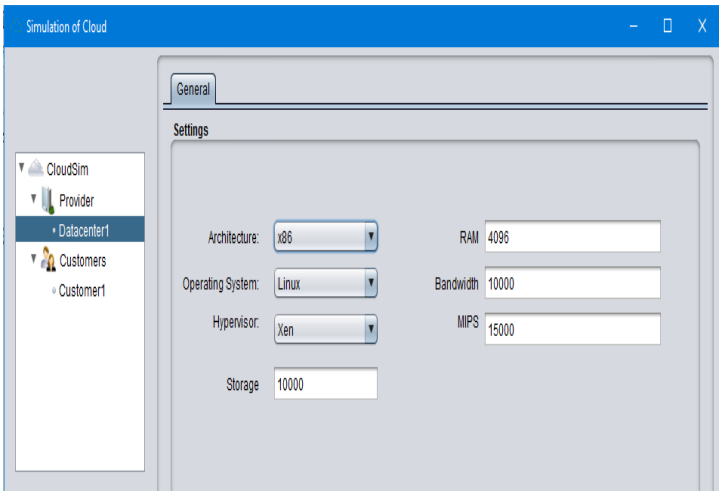

Fig 3. Simulated Datacenter and its configuration.

Three Virtual Machines are created on data center having a different configuration. Three VMs having RAM (MB) 1300, 1340, 1400. And having MIPS 200, 500, 1000 respectively. The following figure shows how the inputs are taken for the presented algorithm to be tested. All three algorithms are tested on same inputs.

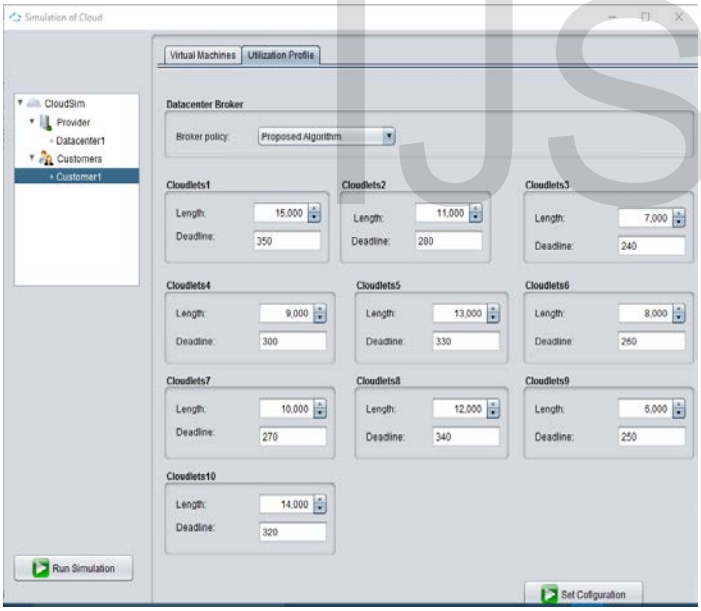

Fig 4: Taking Inputs for Presented Algorithm

After running the simulation with the presented algorithm the following output is generated. The following figure shows the output of the simulation. It shows how cloudlets are scheduled in the desired sequence. The makespan and average waiting time are calculated from the output.

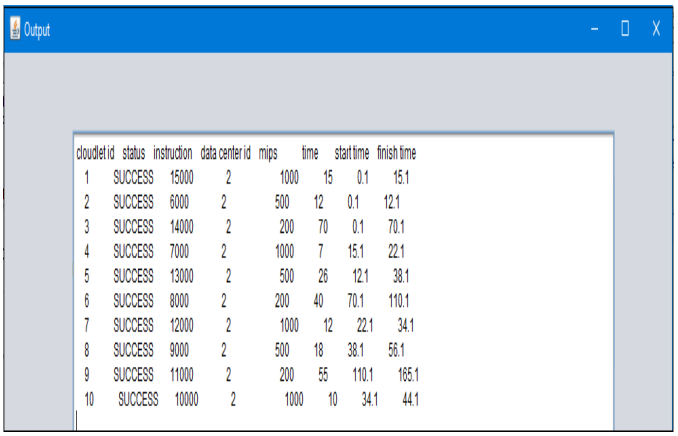

Fig 5: The output of presented algorithm.

The comparison of the result of the presented algorithm is done with existing traditional algorithms. Two algorithms are used for comparison first algorithm have only task length as scheduling criteria and another one have the only deadline as scheduling criteria. Makespan is total time taken to complete the task. The following figure shows the graph of a comparison of all the three algorithms with respect the makespan. . Comparative analysis shows the result that makespan of the task is reduced as compared to existing algorithm.

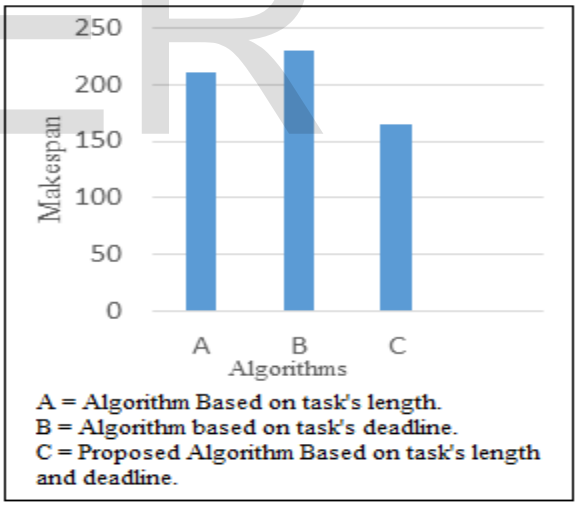

Fig 6: Comparison of Each algorithm with respect to Makespan

Average waiting time is the calculated average of waiting time required by every cloudlet. The following figure shows the graph of a comparison of all the three algorithms with respect the average waiting time. Comparative analysis shows the result that average waiting time of the task is reduced as compared to existing algorithm.

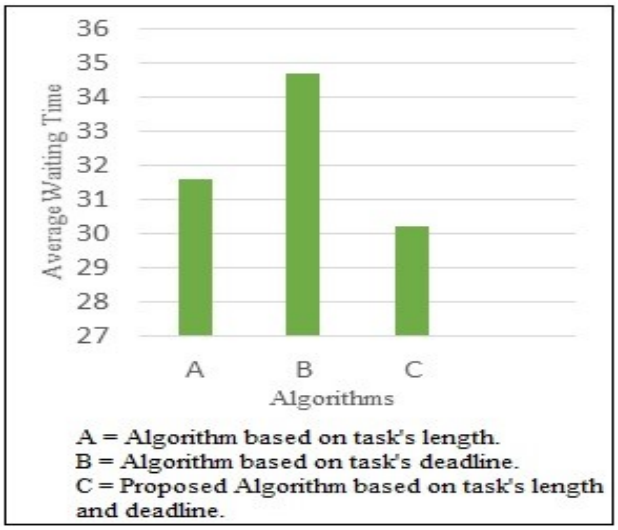

Fig 7: Comparison of Each algorithm with respect to Average Waiting Time.

5 Conclusion and Future Work

In this paper, the scheduling algorithm is presented to schedule a task on two parameters that are task length and its deadline. Algorithm is implemented and tested in the simulated environment using CloudSim simulator. The results are compared with two algorithm one which is having only task length as scheduling criteria and another one which have only task's deadline as its scheduling criteria. After Comparative analysis, we came to know that the Makespan and Average Waiting Time can be reduced by implementing presented algorithm in simulated Cloud Computing environment. In future, the presented algorithm can be advanced by taking task length, deadline and priority into consideration as parameters for scheduling. And may be further improved by making presented algorithm preemptive.

References

[1] Barrie Sosinsky, "Defining Cloud Computing" in Cloud Computing Bible, Indianapolis, Indiana: Wiley Publishing, Inc. 2011

[2] Choudhary & Peddoju "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment", International Journal of Engineering Research & Applications, 2564 – 2568, Issue 3, Vol. 2, May – June 2012 [2248 - 9622].

[3] Henzinger, Thomas A., et al. "Static scheduling in clouds." memory 200.o1 (2011): i1.

[4] Selvarani, S., and G. Sudha Sadhasivam. "Improved Cost-Based Algorithm For Task Scheduling In Cloud Computing." Computational Intelligence And Computing Research (ICCIC), 2010 IEEE International Conference on. IEEE, 2010.

[5] Jagbeer Singh, Bichitrananda Patra, Satyendra Prasad Singh, " An Algorithm to Reduce the Time Complexity of Earliest Deadline First Scheduling Algorithm in Real-Time System" (IJACSA) International Journal of Advanced Computer Science and Applications, February 2011.

[6] Ghanbari, Shamsollah, and Mohamed Othman. "A priority based job scheduling algorithm in cloud computing." Procedia Engineering 50 (2012): 778-785.

[7] Xiao, Jing, and Zhiyuan Wang. "A PriorityBased Scheduling Strategy for Virtual Machine Allocations in Cloud Computing Environment." Cloud and Service Computing (CSC), 2012 International Conference on. IEEE, 2012.

[8] Li, Ji, Longhua Feng, and Shenglong Fang. "An greedy-based job scheduling algorithm in cloud computing." Journal of Software 9.4 (2014): 921-925.

[9] Gupta, Gaurav, et al. "A simulation of priority based earliest deadline first scheduling for cloud computing system." Networks & Soft Computing (ICNSC), 2014 First International Conference on. IEEE, 2014.

[10] Thomas, Antony, G. Krishnalal, and VP Jagathy Raj. "Credit Based Scheduling Algorithm in Cloud Computing Environment." Procedia Computer Science 46 (2015): 913-920.

[11] Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and Experience 41.1 (2011): 23-50.