Performance Analysis of Enhanced Max-Min and Min-Min Task Scheduling Algorithms in Cloud Computing Environment

Varsha Soni
Department of CSE, M.B.M Engineering College, Jodhpur
Dr. N.C. Barwar
Professor, M.B.M Engineering College, jodhpur

ABSTRACT

The cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or cloud provider interaction. With the rapid increase in the number of cloud users day by day and their resource requests, there is a need for efficient resource allocation mechanism at host level which aims at proper utilization of cloud resources. In order to process these requests by making efficient use of the available resources, proper scheduling is required. In this study, Enhanced Max-Min and Min-Min algorithms are studied and their performance is evaluated on the basis of makespan by increasing the number of cloudlets in time shared and in space shared mode.Simulation of these task scheduling algorithms have been performed on Cloudsim3.0.3, then a comparative analysis has been done and results have been presented graphically to find out the efficient task scheduling algorithm.

KEYWORDS: Cloud computing, cloudSim, makespan

I. INTRODUCTION

Cloud Computing is the new cost-efficient computing paradigm in which information and computing power can be accessed over the internet by customers. Cloud Computing holds the potential to eliminate the requirements for setting up of high-cost computing infrastructure for IT-based solutions and services that the industry uses[1]. It promises to provide a flexible IT architecture, accessible through internet from lightweight portable devices. It provides 1)Software As a Service, 2)Infrastructure As a Service and 3)Platform As a Service. In a Saas , A single instance of the software runs on the cloud and services multiple end users or client organizations, In a PaaS, an application or development platform in which users can create their own application that will run on the Cloud is provided, In a IaaS, basic storage is delivered and standardized services are computed over the network.

In a cloud computing environment, a cloud user sends a task to a cloud provider. Assignment of these tasks to physical resources is done at datacenter[2]. The resources in datacenters need to be provisioned in an efficient manner so that the services of cloud can be utilized effectively and efficiently. Provisioning involves both allocation and scheduling. The scheduling in cloud determines the best selection of computing resources for efficient execution of the tasks. It is the set of policies to control the order of work to be performed by a computer system. The reliability and performance of cloud services depends a lot upon the scheduling of the tasks.

Scheduling can be done at task level or resource level or workflow level. The Task Scheduling is one of the most important activity that performs in the cloud computing environment where the task are mapped to the virtual machines. Task scheduling is implemented by the task scheduling algorithms[3]. We have studied, simulated and compared two different task scheduling algorithms which are Enhanced Min-Min and Min-Min algorithm.

II. RELATED WORK

Kangkang Li, et al.[1] discussed the problem of task allocation in data centres .That is, given a set of tasks with different makespan, how to schedule these tasks into the data centers to minimize the average makespan.

Dr. A C Subhajini and Shameer A.P[2]presented the study on different scheduling algorithms for cloud computing, they compare the three scheduling algorithms, first-come-first-serve, round-robin, and shortest job first.

Medhat Tawfeek, et al.[3] discussed that a good task scheduler should adapt its scheduling strategy to the changing environment and the types of tasks. In the paper, a cloud task scheduling policy based on Ant Colony Optimization(ACO) algorithms compared with different scheduling algorithms like First Come First Serve and Round Robin.

Pinal Salot [4] compared the existing scheduling algorithms ,The author emphasized that job scheduling is most important task in cloud computing environment because users have to pay for resource used based upon time. Hence efficient utilization of resource must be important and for that scheduling plays a vital role to get maximum benefits from the resource.

C T Lin[5] conducted the comparative based analysis of scheduling algorithms for resource management in cloud computing environments of resource. There are different types of resource scheduling technologies in cloud in Cloud Computing Environment. These are implemented at different levels based on different parameters like cost, performance resource utilization, time, priority, physical distances, throughput, bandwidth, resource availability etc.

III. TASK SCHEDULING ALGORITHMS

A cloud user sends the request to the cloud provider and this request is termed as task. It may include entering data, processing, accessing software, or storage functions[4]. The scheduling at task level means the process in which the tasks are mapped to the vacant resources after looking at the characteristics and requirements of the tasks. It allocates the computer machines to tasks in such a manner that the completion time is minimized as possible. The task scheduling helps in:

(i)increasing the usage of resources.

(ii)it is responsible for mapping jobs submitted to cloud environment onto available resources in such a way that the total response time ,the makespan is minimized.

(iii)balance between improving the quality of services and at the same time maintaining the efficiency and fairness among the tasks.

(iv)to maximize the resource utilization and minimize processing time of the tasks.

To implement task scheduling, there are many scheduling algorithms .We have studied and compared two algorithms as follows:

1) Enhanced Max-Min Algorithm: It selects task with Average or Nearest greater than average execution time (Average or Nearest greater than average task) then assign to be executed by resource with minimum completion time (Slowest resource).Following is a algorithm where T_i is task, R_j is the resources, C_{ij} is completion time and E_{ij} and Execution Time.

Algorithm:

1. For all submitted tasks in Meta-task; Ti

1.1 For all resources; Rj

1.2 Cij = Eij + rj

2. Find task Tk costs Average or nearest Greater than Average execution time.
3. Assign task Tk to resource Rj which gives minimum completion time (Slowest resource).
4. Remove task Tk from Meta-tasks set.
5. Update rj for selected Rj.
6. Update Cij for all j.
7. While Meta-task not Empty

7.1. Find task Tk costs maximum completion time.

7.2. Assign task Tk to resource Rj which gives minimum execution time (Faster Resource).

7.3. Remove Task Tk form Meta-tasks set.

7.4. Update rj for Selected Rj.

7.5. Update Cij for all j.

2) Min-Min Algorithm: It selects task with the overall minimum expected execution time (shortest Task) then assign to be executed by resource with minimum expected completion time (Slowest Resource). Following is a algorithm where T_i is the tasks, R_j is the resources, C_{ij} is completion time and E_{ij} and Execution Time.

Algorithm:

1. For all submitted tasks in Meta-task; Ti

1.1. For all resources; Rj

1.2 Cij = Eij + rj

2. Find task Tk costs minimum execution time (Largest Task).
3. Assign task Tk to resource Rj which gives minimum completion time (Slowest resource).
4. Remove task Tk from Meta-tasks set.
5. Update rj for selected Rj.
6. Update Cij for all j.
7. While Meta-task not Empty

7.1. Find task Tk costs minimum completion time.

7.2. Assign task Tk to resource Rj which gives minimum execution time (Faster Resource).

7.3. Remove Task Tk form Meta-tasks set.

7.4. Update rj for Selected Rj.

7.5. Update Cij for all j.

IV. EXPEERIMENTAL SETUP AND RESULTS

The undertaken scheduling algorithms are simulated using the cloudsim 3.0.3 which helps to model various different cloud applications by creating DataCenter, Host, Cloudlet, Users, Virtual Machines and many other utilities[7]**.** The following parameters has been considered as shown in Table I.

Table I: Parametric Table

No. Of VMs	4
No.of cloudlets	30
No. of data centers	1
No. of PE's	4
MI of cloudlets	2000, 4000, 6000 ,8000, 10000,12000, 14000, 16000, 18000, 20000,22000, 24000, 26000, 28000, 30000, 32000,34000, 36000,38000,40000,42000,44000,46000,48000,50000,52000,54000, 56000,58000,60000
MI of VM's	100, 250, 300, 220
PE's No.	1,2,1,1
RAM required	512 MB

We have considered the two scenarios as follows:-

(i)Space-shared policy: In the space-shared policy , it schedules one task on VM after at a time, after its completion it schedules another tasks on VM. This same policy is used to schedule VM on hosts.

(ii)Time-shared policy: In the time-shared policy, it schedules all the tasks on VM at same time. It shares the time slice among all the tasks and schedules simultaneously on the VM.

Using the above parameters and policy, we have compared all the algorithms and calculated their respective makespan[8]. Makespan means total completion time required for task execution. Table II represents the makespan of both the algorithms in space shared mode.

Table II: Makespan in space-shared mode

No.ofCloudlets →	5	10	15	20	25	30
Algorithms ↓						
EMM	183.51	786.76	2864.87	6791.26	12896.61	23350.51
Min-Min	208.5	1029.33	3476.48	8006.13	14820.57	25300.83

Table III: Makespan in time-shared mode

No.ofCloudlets →	5	10	15	20	25	30
Algorithms ↓						
EMM	183.51	673.09	2067.26	4507.92	8218.68	14273.29
Min-Min	172.5	781.36	2264.51	4954.08	8824.73	14697.80

TableIII represents the makespan of all the four algorithms in the time shared mode.

The above table showing makespan of the scheduling algorithms as the number of cloudlets increasing by five in space-shared and time-shared mode. Graphically representing the above comparative result table we will get the following graph.

IV. COMPARISON OF OVERALL MAKESPAN IN SPACE SHARED AND TIME SHARED MODE

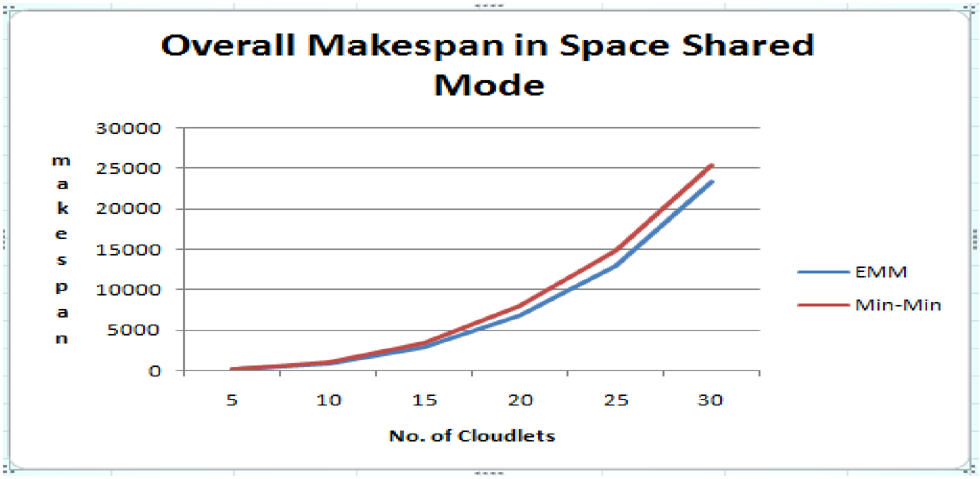

Figure 1: Overall Makespan Analysis in Space- Shared Mode

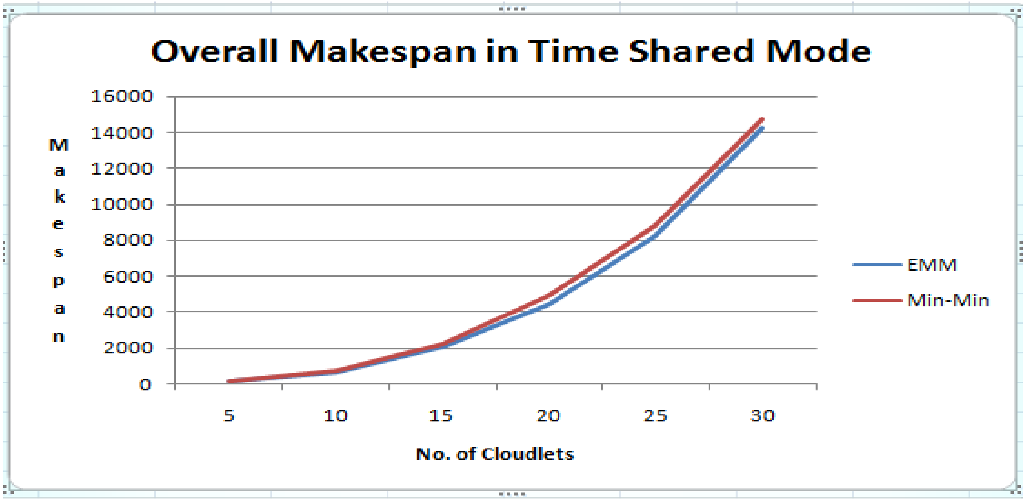

Figure 2: Overall Makespan Analysis in Time- Shared Mode

Figure 1 and 2 show comparison of overall makespan of 30 cloudlets on four virtual machines (V0, V1, V2, V3) in space shared and time shared mode using Enhanced Max-Min and Min-Min algorithm for resource allocation in the same simulation environment. As the number of cloudlets are increasing from 20 cloudlets, the differences in the makespan of both the algorithm is becoming visible.

The results shown above concludes that as the number of cloudlets increases, makespan also increases exponentially but the makespan of the Enhanced max-min in space shared mode growing rate is less than the other algorithms in space-shared and timeshared mode. Thus, Enhanced Max-Min in space-shared proves to be the better algorithm.

V. CONCLUSION

To meet the thousands of service requests while making best possible use of available resources simultaneously satisfying both users and service providers as well is an important aspect in cloud and thus the task scheduling plays the most important role in cloud computing[10]. In this paper, analysis of Enhanced max-min and min-min scheduling algorithm is done using makespan as a criteria. The comparative result shows that as the number of cloudlets increases the makespan of algorithms also increases but the rate of growth of makespan of Enhanced Max-Min algorithm in space shared mode is less than other growth rate of Min-Min algorithm in space-shared and time-shared and Enhanced Max-Min algorithm in time shared mode .Thus Enhanced Max-Min algorithm in space shared mode is more efficient than other scenarios.

VI. FUTURE WORK

In future, our objective is to reduce the execution time of tasks in cloud computing environment with dynamic task allocations. There is more possibility that the grouping or clustering of the dynamic tasks checked on dependencies may minimize the makespan while maintaining the cost and processing power within the limits. In the future, the real time allocation of tasks may be done which help us to maintain the SLA(Service Level Agreement) and manage huge numbers of cloud users.

VII. REFERENECES

[1] CT Lin, “Comparative Based Analysis of Scheduling Algorithms for Resource Management in Cloud Computing Environment”, International Journal of Computer Science and Engineering (IJCSE) – Volume 1 Issue 1, July 2013.

[2] Rodrigo N. Calheiroos, Rajiv Ranjan, Cesar A.F.De Rose and Rajkumar Buyya, “ CloudSim: A Novel Framework for Modelling and Simulation of Cloud Computing Infrastructure and Services”, GRIDS Labortory.

[3] Pinal Salot, “A survey of Various Scheduling Algorithm In Cloud Computing Environment”, (IJRET) International Journal of Research In Engineering and Technology, ISSN: 2319-1163.

[4] Deepika Saxena,Dr. R.K. Chauhan and Shilpi Saxena, “ Evaluating Makespan of Min-Min and Max-Min Scheduling Techniques in Cloud Computing: Concepts and Comparison”, International Journal of Trend in Research and Development, Vol.3(6), No.4, ISSN: 2394-9333.

[5] Atul Mishra, Mayanka Katyal “Application of Selective Algortihm for Effective Resource Provisioning In Cloud Computing”, International Journal on Cloud Computing Service and Architecture(IJCCSA), Vol. 4, February 2014 .

[6] KangKang Li, Ziqi Wan, Jie Wu and ADAM Blaisse , “ Towards Makespan Minimization Task Allocation in Data Centers”,GRIDS Laboratory.

[7] Medhat Tawkeef,Ashraf El-Sisi,Arabi Keshk and Fawzy Torkey , *IAJIT* ,“Cloud Task Scheduling Based on Ant Colony Optimization”, VOL. 12, NO. 2, March 2019.

[8] Upendra Bhol, Purvi N. Ramanuj, “Enhanced Max-Min Task Scheduling Algorithm in Cloud Computing”, IJAIEM, Volume 2, Issue 4,April 2013.

[9] Monica Gahlawat, Priyanka Sharma “Analysis and Performance Assessment of CPU Scheduling Algorithm in Cloud using CloudSim”, IJAIS, International Journal of Applied Information System, Issues, Vol. 5, No 9, July 2013.

[10] Harbinder Singh Sidhu,“Comparative Analysis of Scheduling Algorithm of CloudSim in Cloud Computing”, International Journal ofComputer Applications, Volume 97, No. 16, July 2014.