International Journal of Computing Academic Research (IJCAR)
ISSN 2305-9184, Volume 4, Number 6 (December 2015), pp.182-190

http://www.meacse.org/ijcar

# Application of Min-Min and Max-Min Algorithm for Task Scheduling in Cloud Environment Under Time Shared and Space Shared VM Models*

Sanjay Kumar and Atul Mishra
Department of Computer Engineering, YMCA University of Science and Technology, Faridabad, Haryana, India

## Abstract

Cloud Computing is one of the latest emerging technology in which services are provided by geographically distributed resources by accessing these remotely. Task scheduling is the most challenging issue in cloud computing environment. In task scheduling, tasks are mapped to allocated resources according to their requirement and characteristics. The scheduler should adapt its scheduling strategy according efficient execution of tasks. In this paper, Min-Min and Max-Min, task scheduling algorithms have been taken for simulation analysis. Both algorithms use static approach and task independent scheduling. These two algorithms are scheduled on both, space shared and timeshared manner, one by one, on virtual machines to be scheduled on actual hosts. These scenarios are tested using a cloud simulator, called CloudSim. These scenarios and experimental result have been compared to show under different load scenario how do they perform.

**Keywords:** Cloud Computing, Task Scheduling, CloudSim, Min-Min, Max-Min

## Introduction

Cloud computing [1] technology provides on-demand services to the end users by retrieving scalable and virtualized computing resources from the internet using different web tools. It has emerged from focus development of grid computing, parallel computing, virtualization and web technologies. In its internet based computing model [2] it delivers Software as a services (SaaS), Platform as a service (PaaS) and Infrastructure as a service (IaaS) . In SaaS, software application is software distribution model which is made available by the cloud provider or vendor and services are provides to end user over the internet. In PaaS development tools and platforms are supplied by provider to allow the developer to create and run the various web based application. In IaaS virtualized hardware and software as computing components are provided to the cloud user in the form of Virtual Machine (VM) to develop the platform and software for the end user. For providing the service by the cloud service provider and using the services by the cloud user, an initial agreement called the Service Level Agreement (SLA) [3] has to be made between the cloud users and the cloud service provider. While resource allocation is made to the cloud user, SLA violation should be avoided as much as possible or SLA violation should be minimal without compromising Quality of Service (QOS) [4] parameters like performance, availability, response time, security, throughput , reliability etc.

These services are provided by the cloud host on the basis of pay as-you-use model to the cloud users. In cloud environment, scheduling is the major issue. Scheduling is responsible for efficient utilization of the resources. How to utilize Cloud computing resources expertly to achieve the maximum gain with task scheduling system is Cloud computing service provider's ultimate target.

Resource Provisioning is combination of both Resource allocation and Task Scheduling. Available resources on the cloud must be setup in such a manner that their computing power are utilized efficiently

* **The paper was first presented in International Conference on Recent Trends in Computing and Information Technology organized by B.S Anangpuria Institute of Technology and Management, Faridabad, India.**

and effectively available to the users on real time mechanism to completion of tasks assigned by the cloud users.

Scheduling refers to the strategy to control the work order to be performed by a computing system. The goal of scheduling algorithms in cloud computing is disturbing the load on processors and maximizing their utilization while minimizing the total execution time of task. Task scheduling is one of the most famous optimization problems, plays an important role to improve potential power of systems. The main purpose is to assignment of tasks to the allocated resources in accordance with time, which involves discovering out a suitable order in which tasks can be executed under transaction logic constraints. There have been various kinds of scheduling algorithm existing in cloud computing system, and task scheduling is one of them. There have been various kinds of scheduling algorithm existing in cloud computing system, and task scheduling is one of them. The main benefit of task scheduling is to obtain a high performance from computing system which leads to best system throughput. Scheduling regulates availability of resources and a good scheduling policy provides maximum utilization of computing resource.

There are two mode of the task scheduling

a. Space-shared
b. Time-shared

There are broadly two categories of scheduling algorithm.

1. Static scheduling algorithm
2. Dynamic scheduling algorithm

Both of have their own advantages and limitations. Dynamic algorithm performance is better than static algorithm but has more overhead compare to it. The performance of task scheduling depends upon the various parameters like execution time, response time and completion time. Execution time depends upon the nature of the task whether they are dependent or independent tasks. The primary aim to focus on minimizes the overall makespan of tasks on machines in such a way that it provides better performance to cloud providers and cloud users.

**Related Work**

There is a lots related work are done in task scheduling to achieve the minimum makespan. Generalized Priority algorithm [5] proposed for efficient execution of task and compare with Round Robin Scheduling and FCFS. In the proposed algorithm, the tasks are prioritized according to their MI size on the basis of one having highest MI value has highest rank. The Virtual Machines are prioritized according to highest MIPS value has the highest rank. Thus, the key factor for prioritizing tasks is MI and for VM is MIPS. This algorithm provides better performance compared to other conventional scheduling algorithm. The limitation of this algorithm is there are limited task in this task scheduling. There is a possibility to take more tasks and try to minimize the execution time.

Jangra et al. [6] define computation regarding the user's tasks is done with the help of various parameters and conditions to be encountered in the simulation of tasks. The user's tasks are classified on the accordance of data and resources requested by the task and then prioritized. Selection of resources is done on the accordance of its turnaround time and cost using greedy approach. A priority formula is used for the task selection. This way of task selection and resource selection gives better results over sequential scheduling. This method gives a way to more future findings in the scheduling techniques in a cloud environment.

Acharin et al. [7] proposed novel scheduling mechanism, , first priorities the tasks and Virtual Machines and then defines a tree based data structure called Virtual Machine Tree (VMT) in which every nodes of a tree represents a Virtual Machine. The clustering of task is done based on number of leaves nodes in the VMT. The modified DFS algorithm identify the suitable Virtual Machines, for which the assigned tasks be executed. Comparison of this algorithm with the FCFS and priority based algorithm result represent that the proposed algorithm is much efficient than priority based algorithms and FCFS.

R. Vijayalakshmi [8] focuses on minimizing the makespan to utilization of allocated resources more efficiently. The user's tasks first got prioritized and then submit to the virtual machine on the basis of the prioritization. The highest prioritized task is scheduled on the largest processing powered virtual machine. Comparison of this algorithm with the FCFS result shows that the proposed algorithm is more efficient than FCFS algorithm.

Mishra et al. [9] proposed a selective algorithm which uses certain heuristics to decide between the two algorithms (Min-Min, Max- Min) so that overall makespan of tasks is minimized. Comparison of this algorithm with FCFS algorithm shows that this algorithm is more efficient to minimize the makespan .

Mousumi Paula et al. [10] defines a new probabilistic measurement or credit of the tasks for allocation of the resources. This measurement depends upon the arrival time and the execution time of the task over the available resource. The consequence of this parameters leads to the minimum completion time. The cost matrix is used for the assigning the tasks to the resources. The task which have high probability to get resource and resource which fits better for the task are allocated.

Pardeep Kumar et al. [11] proposed an improved genetic algorithm in which two conventional algorithms Max-Min and Min-Min are merged with the standard Genetic algorithms. The idea for generating the initial population randomly in genetic algorithm is replaced by Min-Min and Max-Min which increases the chances of produces better child. The experiment results shows improved genetic algorithm is much better compare to the standard genetic algorithm.

Poonam Devi et al. [12] implements short job scheduling in which user requests is define in the requirement query which includes process time, deadline, arrival time and the input output requirement of the processes. After that cloud is prioritized in which available memory and load are defined for every virtual machine. All the process requests are arrange according to memory requirements. The migration process is also defined for the overload conditions. The implementation of the proposed work carried in the MATLAB Toolkit. In future the other parameters for migration can be used.

Santhosh et al. [13] modified improved Max-Min algorithm to define two new algorithms based on the average execution time. In this largest task just greater than the average execution time is selected and assigned to the resource which gives minimum completion time. The average execution time is calculated using arithmetic mean for independent tasks and geometric mean for dependent tasks. The experimental comparison of these two algorithms with the Max-Min and Improved Max-Min represents that the proposed algorithms are more efficient for minimization of the makespan.

**Scheduling of Tasks**

The main motive of task scheduling algorithms in Cloud computing systems is distributing the load on processors and maximizing resource utilization while minimizing makespan .

Task Scheduling in cloud computing can be broadly classified into three phases namely-

**Resource detection and filtering** - Broker selects the resources availability in the cloud computing system and gathers the respective status information of them.

**Resource allocation** – Required resource is picked up on the basis of the parameters of task and resource.

**Task submission** –Task is assigned to allocated resource.

To minimize the overall makespan of the tasks on machines and provide the better performance min-min and max-min algorithms are designed. These algorithms are simulated under space-shared and time-shared mode to get minimum makespan.

Min-Min[9] algorithm has following procedure:

Phase 1: First computes the completion time of every task on each machine and then for every task select the machine which processes the tasks in minimum possible time.

Phase2: Among all the tasks in Meta task the task with minimum completion time is selected and is assigned to machine on which minimum execution time is expected. The task is removed from the list of Meta Task and the procedure continues until Meta Task list is empty.

Max-Min[9] algorithm follows the following procedure:
Phase1: First computes the completion time of every task on each machine and then for every task chooses the machine which processes the tasks in minimum possible time
Phase2: Among all the tasks in Meta Task the task with maximum completion time is selected and is assigned to machine. The task is removed from the list of Meta Task and the procedure continues until Meta Task list is empty.

If x denotes the number of machines and y denotes the number of meta-tasks then Min-Min algorithm has the time complexity of O ($y^2x$) and Max-Min algorithm has time complexity of O ($y^2x$).

To better understand the working of these scheduling algorithms with the aim to achieve minimum makespan in cloud computing environment a simulated environment was prepared using CloudSim software tool. In software simulation tool CloudSim [14,15] tasks are resembled as Cloudlets and machines are resembled as VMs and cloud itself is resembled as a Datacenter. Cloudlets are schedule on the VMs where each VM is virtual machine on which cloudlets runs. VMs are schedules onto hosts (Physical Server) based on hardware requirements (memory, processing elements, bandwidth and storage). Processing power of host is measured using MIPS (Million Instruction per Second). MI is used as the primary unit to measure the expected execution of the task (cloudlet).

In space-shared scheduling, a cloudlet is executed to completion before another cloudlet acquire the VM for execution. In time-shared scheduling more than one cloudlet may be run on same VM in different time slice by consistent acquire of VMs from cloudlets.

## Experimental Results

**Performance Metrics** The performance metrics of the task scheduling in cloud environment is makespan of the cloudlets on given virtual machines. Makespan is the basic unit of throughput in heterogeneous system.
Makespan is calculated using following terms and their relations:

$T_{start}$= represent the starting time
$T_{exe}$ = represent the execution time
$T_{finish}$ = represent the finishing time
$T_{comp}$= represent the completion time
$T_{finish} = T_{start} + T_{exe}$

MakeSpan=Completion time
Completion time= $\max(T_{finish} = T_{start} + T_{exe})$

**Simulation Environment** The experiment is conducted under following environment parameters:
a) Resource computing power are defined in MIPS (Million Instructions per Second).
b) Cloudlets are attributed by their MI (Million Instructions).
c) Resources and cloudlets nature may be heterogeneous.
d) The cloudlets can be exectued on VMs in timeshared or spaceshared manner.
e) Allocation of VMs on host can be in timeshared or spaceshared manner.
f) The environment is static such that list of Meta Tasks and resources and values of their respective parameters is to be assigned before simulation.

## Experimental Data and Results

The experimental testing is used following Input Data

Table [1]: Input Data

| | |
|---|---|
| Number of Vms | 3 |
| Number of Cloudlets | 4 |
| Number of Datacenter | 1 |
| Number of Hosts | 1 |
| RAM required for VM | 512 (MB) |
| MI of cloudlets | 200000, 80000, 120000, 160000 |
| MIPS of VMs | 250 ,425, 500 |

Table [1] data was used to calculate the performance of max-min and min-min algorithms. The completion time of every task is evaluated in the simulated environment.

CloudSim[8] toolkit is used to setup simulation environment , the completion time of every cloudlet is figure out in ms(millisecond) and extra time of few milliseconds is consider for initialization of each entity of Cloud Computing System(Cloulets , VMs, Hosts and Datacenters) is added to completion time of cloudlets on their allocated VMs.

The testing of algorithms is performed in following scenarios

Scenario 1: Overall Makespan of five tasks on three VMs in spaceshared mode using Min-Min Algorithm , Fig 1.

Scenario 2: Overall Makespan of five tasks on three VMs in timeshared mode using Min-Min Algorithm , Fig 2.

Scenario 3: Overall Makespan of three tasks on three VMs in spaceshared mode using Max-Min Algorithm , Fig 3.

Scenario 4: Overall Makespan of five tasks on three VMs in timeshared mode using Max-Min Algorithm, Fig 4.

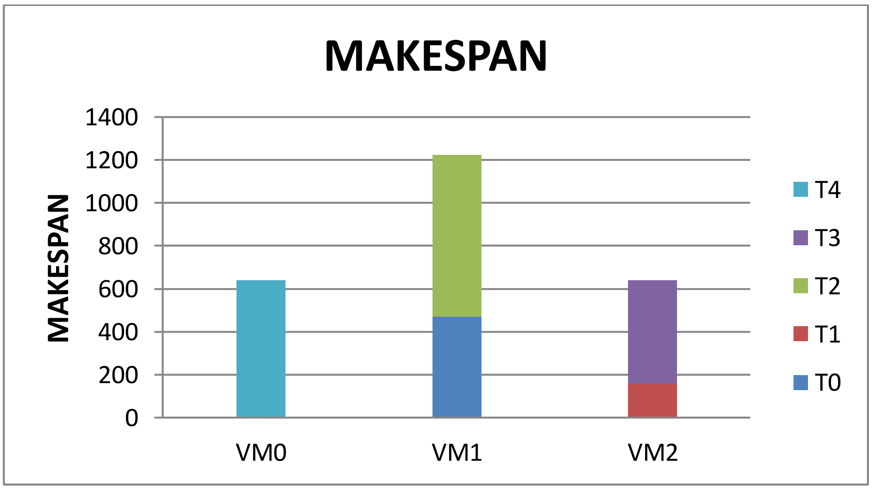


Fig 1: Overall Makespan of 5 tasks on 3 VMs in space-shared mode using Min-Min Algorithm

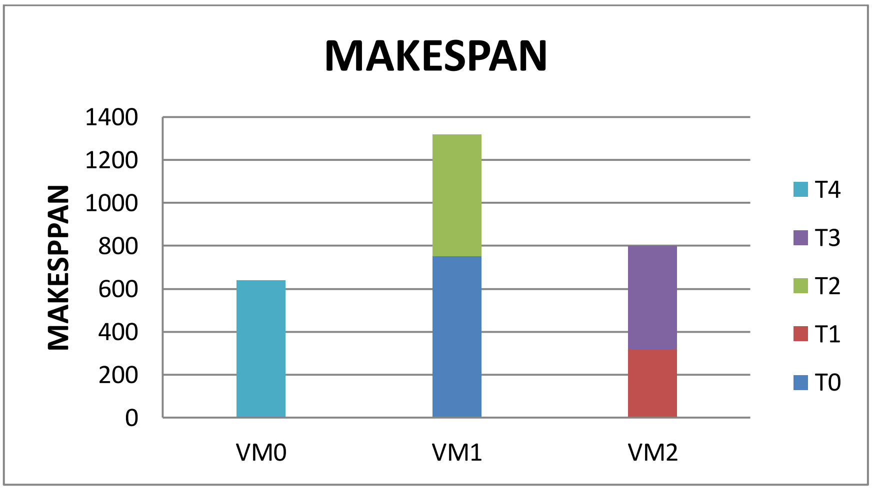


Fig 2: Overall Makespan of 5 tasks on 3 VMs in timeshared mode using Min-Min Algorithm

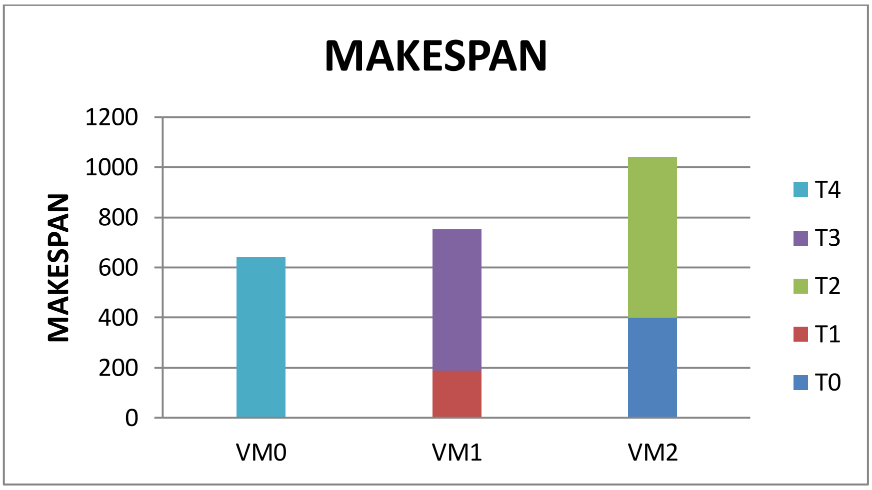


Fig 3. Overall Makespan of 5 tasks on 3 VMs in space-shared mode using Max-Min Algorithm

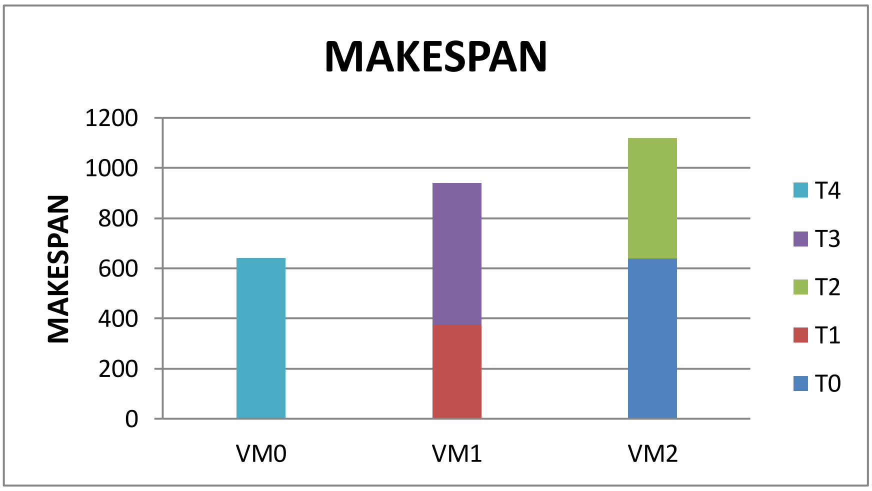


Fig 4. Overall Makespan of 5 tasks on 3 VMs in timeshared mode using Max-Min Algorithm

In Fig 1 and 2, show overall makespan of five cloudlets (T0, T1,T2,T3,T4) on three VMs(VM0, VM1, VM2) in space-shared and time-shared scheduling mode using min-min algorithm for resource allocation. The overall Makespan of five cloudlets on three VMs using min-min algorithm in space-shared and time-shared mode is 1223 and 1317 respectively. In Fig 3 and 4, show overall makespan of five cloudlets (T0,T1,T2,T3,T4) on three VMs(VM0, VM1, VM2) in space-shared and time-shared scheduling mode using max-min algorithm for resource allocation . The overall makespan of five cloudlets on three VMs using max-min algorithm in space-shared and time-shared mode is 1040 and 1120 respectively. Table [2] represents the comparison of four different scenarios evaluated in given environment.

Table [2]: Comparison of makespan of tasks on VMs in Min-Min and Max-Min Algorithm

| Mode/ Algorithms | Using Min-Min | Using Max-Min |
|---|---|---|
| Space Sharing | 1223 | 1040 |
| Time sharing | 1317 | 1120 |

Fig 5 and 6 show the comparison of the min-min and max-min algorithm in time shared and space shared in the same simulation environment. Fig 5 represents that, as the number of cloudlets increase makespan also increases exponentially but the makespan of the max-min in timeshared mode growing rate is less than the min-min in timeshared.

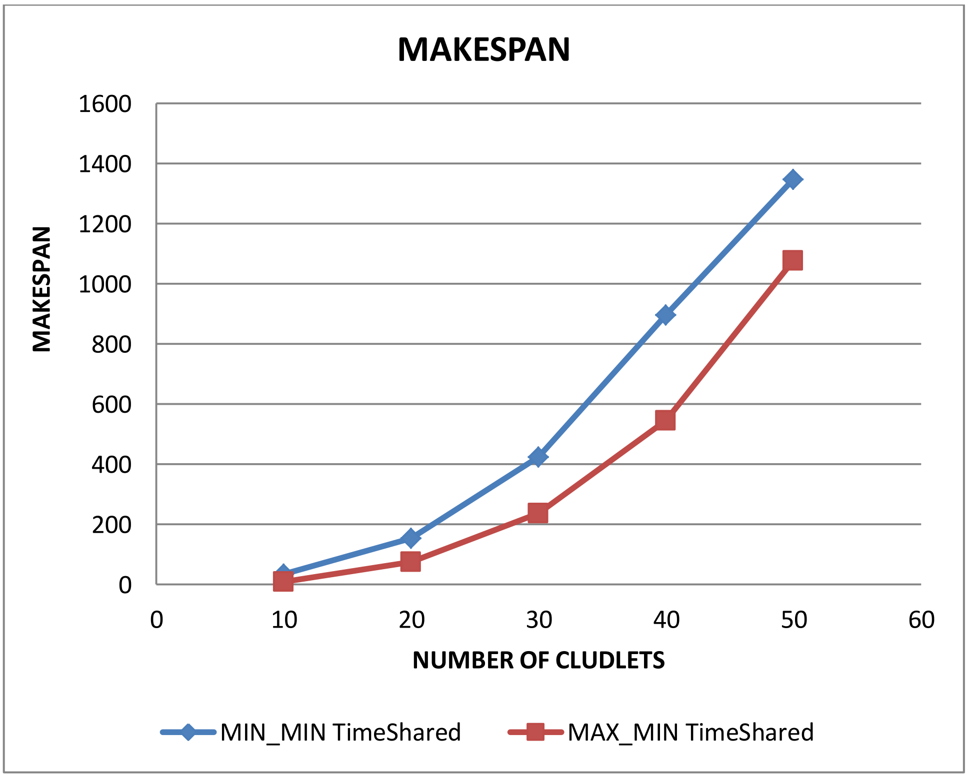


Fig 5. Comparison of Min-Min and Max-Min algorithm in Time shared mode

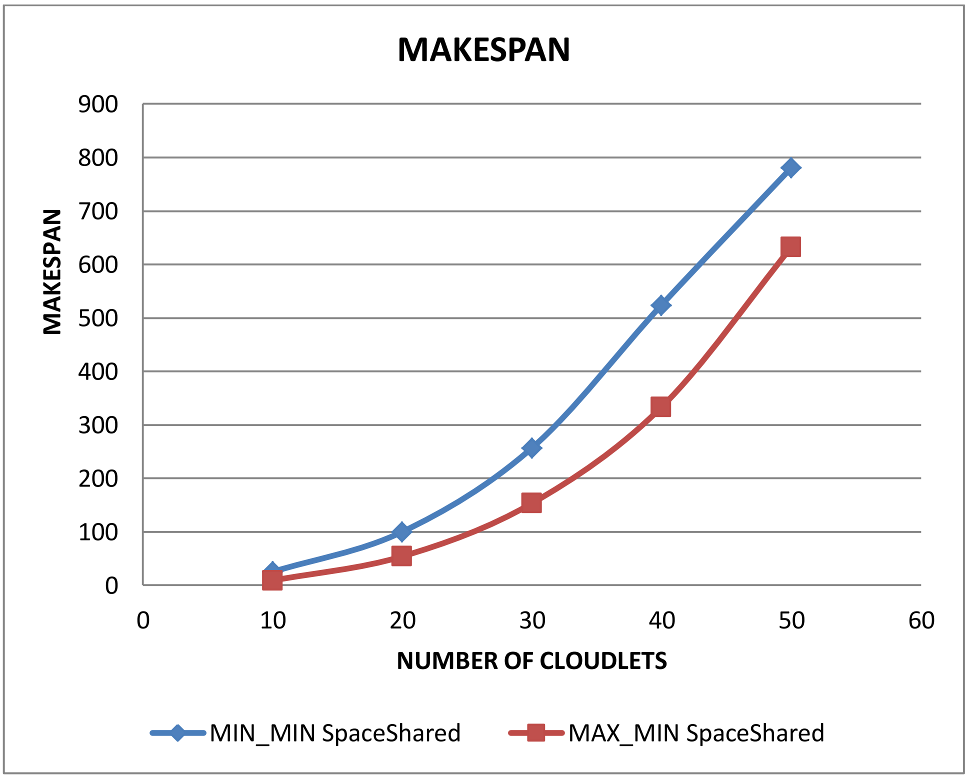


Fig 6. Comparison of Min-Min and Max-Min algorithm in Space shared mode

In Fig 6, there is also the makespan growing rate is less in min-min as compare to max-min in space shared mode. As the number of cloudlets increase in the period of 10 the difference between the two curves also varies rapidly.

When the Fig 5 and 6 is compared, it is concluded that for the same simulation environment the makespan of max-min algorithm in space shared mode is increases less exponentially whereas in the other scenarios makespan increases rapidly as the number of cloudlets increases.

## Conclusions

Scheduling is one of the most important aspects in cloud computing environment. In this paper, analyses of various scheduling algorithm which efficiently schedule the computational tasks in cloud environment as been carried out. Min-min and Max-min use both the static approach and independent tasks scheduling. The result shows that the Max-min algorithm in spaceshared mode is more efficient than other scenarios. Min-Min algorithm provides minimum delays in processing of tasks. Max-Min provides the proper utilization of the VM and load balancing.

## 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.

## References

[1] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud Computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer System.25, 599–616 (2009)

[2] V. R. Srinivasa, N. K. R. Nageswara and E. K. Kumari, ―Cloud Computing: An Overview,‖ Journal of Theoretical and Applied Information Technology, 2009

[3] Foued Jrad, Jie Tao and Achim Streit "SLA Based Service Brokering in Intercloud Environments" in proceedings of 2nd International Conference of Cloud Computing (CLOSER 2012) SciTePress
[4] Sanjay Agrawal, Sonal Dubey "QoS Driven Task Scheduling in Cloud Computing" International Journal of Computer Applications Technology and Research Volume 2– Issue 5, 595 - 600, 2013.
[5] Dr. Amit Agarwal, Saloni Jain "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment" International Journal of Computer Trends and Technology (IJCTT) volume 9 number 7 , Mar 2014
[6] Dr Ajay Jangra , Tushar Saini "Scheduling Optimization in Cloud Computing" International Journal of Advanced Research in Computer Science and Software Engineering ISSN: 2277 128X ,Volume 3, Issue 4, April 2013
[7] Raghavendra Achar, P. Santhi Thilagam "Optimal Scheduling of Computational Task in Cloud using Virtual Machine Tree" Third International Conference on Emerging Applications of Information Technology (EAIT) , IEEE 2012
[8] R. Vijayalakshmi, Mrs. Soma Prathibha "A Novel Approach for Task Scheduling in Cloud" IEEE – 31661 , July 2013,
[9] Atul Mishra, Mayanka Katyal "Application of Selective Algorithm for Effective Resource Provisioning In Cloud Computing Environment" International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol. 4, No. 1, February 2014
[10] Mousumi Paula, Goutam Sanyalb "Task-Scheduling in Cloud Computing using Credit Based Assignment Problem" International Journal on Computer Science and Engineering (IJCSE), ISSN 0975-3397 Vol. 3 No. 10 October 2011.
[11] Pardeep Kumar, Amandeep Verma "Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm" International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), ISSN: 2277 128X Volume 2, Issue 5, May 2012
[12] Poonam Devi, Mr. Trilok Gaba "Implementation of Cloud Computing By Using Short Job Scheduling" International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), ISSN: 2277 128X , Volume 3, Issue 7, July 2013
[13] Santhosh, Dr. Manjaiah D.H "An Improved Task Scheduling Algorithm based on Max-min for Cloud Computing" International Journal of Innovative Research in Computer and Communication Engineering(IJIRCCE) Vol.2, Special Issue 2, May 2014
[14] Rizwana Shaikh , M. Sasikumar "Cloud Simulation Tools: A Comparative Analysis" International conference on Green Computing and Technology, 2013
[15] B. Wickremasinghe, R. N. Calheiros, R. Buyya. "CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications"24th International Conference on Advanced Information Networking and Applications (AINA) IEEE Computer Society, 2010, pp. 446-452. DOI: 10.1109/AINA.2010.32