IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. 1 (May – Jun. 2015), PP 42-49
www.iosrjournals.org

Improved Max-Min Scheduling Algorithm

Navdeep Kaur[1,] Khushdeep Kaur[2]
(*Student (Mtech.CSE), Bhai Maha Singh College Of Engineering/PTU, INDIA)*
(Assistant professor (Mtech.CSE), Bhai Maha Singh College Of Engineering/PTU, INDIA)

Abstract: *In this research paper, additional constrains have been considered to progress a holistic analysis based algorithm based on Max-Min algorithm, which work on principle of sorting jobs(cloudlets) based on completion time of cloudlets. The improved algorithms here also reviews the job characteristics in method of size, pattern, payload ratio and available storage blocks in particular cluster of contribution of file systems. The observations show no significant overload due to addition of these constrains, as sorting operation remains same and efficient. Storage allocation helps in getting better performance.*
Keywords**:** *Cloud based computing, Max-Min algorithm.*

I. Introduction

Internet users want to make utilization of services of best quality at minimum expenditure in minimum time effort. However, in the computer technology computer resources like as operating system, bandwidth and processing powers are different in sizes and proportions[1]. Hence, main issue is lies in satisfying request of users for better service and using computer resources efficiently in minimum time. Hence scheduling is better option for mapping each task to suitable resources to satisfy some type of criteria. Scheduling can be defined as set of some rules to maintain the order of work directed by a computer environment [12]. Scheduler is used to map jobs to nodes based on selected algorithm selected by users of cloud services [8].There are so many scheduling methods and algorithms used in maintaining workload.

To be more specific, scheduling is done in according to gain maximum profit and maximizing their usage. Hence this paper tries to the capture focus on scheduling algorithms used in cloud environment. There are mainly two types of scheduling. Static based scheduling and dynamic based scheduling . In static scheduling homogeneous resources are used. Prior knowledge is important like number of nodes, processing power, memory, etc. In the dynamic environment, heterogeneous resources are used. In this case resources used are flexible and prior knowledge of resources is not required for best results such as no. of nodes, processing power, memory etc[4]. In cloud computing Quality of Service and Load Balance (LB) are main conditions [6].

Scheduling algorithms can be categorized into two main types of algorithms: Batch mode heuristic algorithm and online mode heuristic algorithm. Main exiting heuristic scheduling algorithms are Max-Min, Min-Min, FCFS etc. Max-Min performs better than other algorithms because it chooses to assign longer jobs to better resources and reduces the total runtime. However, there are still few limitations in Max-Min, therefore there is requirement to overcome the previous Max-min algorithm.

II. Review on related work

Scheduling is one of the most challenging tasks in a cloud computing environment. Many heuristic algorithms have been recognized for achieving efficient task scheduling in the cloud environment. Conventional job scheduling algorithms provide limited functionalities and performance optimal efficiency. There is an acute need of developing some scheduling policies for complete allocation of tasks to their hardware and software resources. To solve this problem, many authors have work and following is the description of their work.

Weiwei Chen et.al. [8] have evaluated the performance of four heuristics Max-Min, Min-Min, FCFS and MCT with the montage workflow load in overhead robustness of DAG scheduling heuristics experiment and influence of workflow engine delay. In this paper conducted experiments show that overheads plays significant influence on the overall runtime and they behave in different nature. Results conducted show that Max-Min performs better than MCT and FCFS algorithms because it assigns longer jobs to better resources and reduces the overall runtime in task. It does not consider availability like storage capacity and only regard to time.

E. Ullah Munir et. al. [7] have introduced a task scheduling algorithm for grid environments called QoS Sufferage . This algorithm takes network bandwidth and schedules tasks based on their bandwidth requirement. In this paper comparision with the Max-min, Min-min, QoS guided Min-min and QoS priority grouping algorithms has been done. It was found QoS Sufferage acquires smaller make spans. To make effective use there is a need to distribute the submitted tasks amongst the grid resources in such a way that the total response time can be minimized.

Pardeep K**.** et. al. [11] in this research paper, the authors have used a hybrid scheduling algorithm which is an improved version of Genetic algorithm applied to scheduling algorithm . The paper uses genetic scheduling algorithm to reach at optimal conditions used for scheduling. The authors are using Min-Min, Max-Min and using completion time parameters The Min-Min and Max-Min scheduling methods are merged in standard Genetic Algorithm. This paper schedule multiple jobs on multiple machines in an efficient manner such that the jobs take the minimum time for completion, however ,they can further improve algorithm by adding hardware constrains.It also does not consider other parameters like as storage capacity, early execution time, workload type, job size.

S.Devpriya C.R. et.al [9] in this research paper, the authors have presented a modification in Max-Min algorithm. This algorithm is based on RASA (Resource Aware Scheduling Algorithm) algorithm and idea of Max-min strategy. The modified Max-min algorithm is developed to perform scheduling process of RASA (Resource Aware Scheduling Algorithm) in chance of total complete time for all submitted jobs. Their Max-Min algorithm is based on expected execution time but not considers completion time. Their results found that the scheduling tasks using Improved Max-min can achieve lower make span rather than their original Max-Min. It does not consider factors like scalability, availability, stability and others.

Tarun Kumar Ghosh et.al [10] in this research paper, the authors have used Load balanced static grid scheduling using Max-Min Heuristic approach. In their paper, for using unutilized resources two phases are used, in first phase Max-Min is executed and in second phase rescheduling of tasks is done. It was examined that the load balanced Max-Min technique is able to manage the load if there are large number of non-waiting jobs. Their research study is only considered limited number of resources and task execution time parameters, However they can further improve by adding factors like number of hosts, job size.

Rajwinder Kaur et.al [6] In this research paper, the authors have used hybrid improved Max-Min ant algorithm a load approach for load balancing. Their paper uses hybrid Improved Max-min ant optimization technique. The results indicated the comparison between improved Max-min and new hybrid improved Max-min ant approach. It mainly uses total processing time and processing cost. Job size, no. of hosts, workload type and storage requirement are not considered.

III. Research gap

Max-Min tries to allocate longer jobs to better resources and reduces the overall task runtime. However, there are few defects, which need to overcome and hence modify the Max-Min algorithm. This present study need to work on some parameters in order to improve algorithm. In traditional Max-Min main focus is on time parameter, but factor like storage capacity are not consider.

IV. Problem formulation

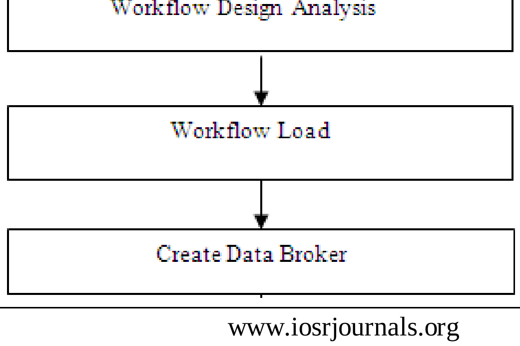
Exiting scheduling algorithm like Min-Min[8], Max-Min[9] always tries to distribute longer jobs with longer completion time to better resources and small jobs assigned to remaining resources. Time taken in this process is main focal point but other factors are ignored. Hence to improve the algorithm job size, no. of hosts, memory, bandwidth and storage capacity should also be considered.

V. Objectives

Conventional existing scheduling algorithm like Max-Min[9] works only on limited range of factors .This type of traditional algorithm can be enhanced by adding some more factors for optimization that may include combination hardware and software. Parameters may be required in process are memory, bandwidth, job size etc. by covering limitations of existing scheduling algorithm.

VI. Methodology

This section explains each step carried out, in process of achieving research gap mentioned in problem formulation. The steps explained below also lead to the process of evaluation of the implementation steps also.



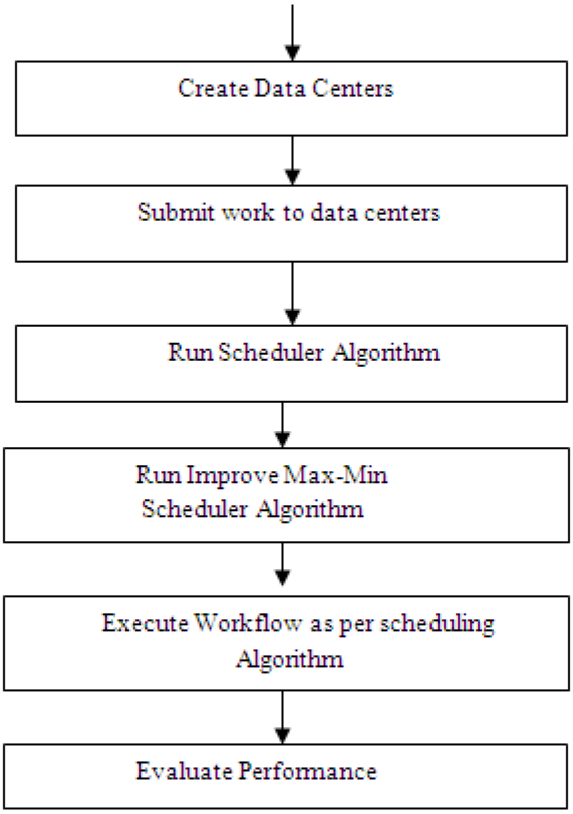


Fig. 6 Block Diagram

Simulation Parameter

Storage block	64 MB	512 MB
Job size	50 KB	500 MB
No. of hosts	2	5
Workload type	Scientific	Scientific

6.1 Step –I Workflow design analysis Firstly scientific workflows are created. Created workflow consists of three tasks communicating to each other. Two tasks send packet and then third task receives them. It selects a scheduling algorithm based on the designing. In this step an environment of analysis, sorting, simulation and visualization for processing tasks is given. Improved Max-min scheduling algorithm will use this simulator for task. Scientific workflow composed of many tasks and modules.

6.2 Step – II Workflow load In this step workload is loaded. Workload initiates a list of jobs that can be send off to a destination resource. Servers or database systems act as components are assigned for handling workload. The workload consists of application programming or executable codes which are processing in the computer system and further computer's application are used by user. Two types of workload are designed.

- Global workload: Global workload that is coming from multi-sites across
- Dynamic workload: Dynamic workload executes the codes of scheduling performed by virtual machine. Virtual machine supposes that there is only one cloudlet for online processing and creating utilization among all PE (Processing Elements).

6.3 Step –III Create Data broker In this step,the data broker is created after workflow load. Data broker acts as a broker on the part of a consumers. It hideaway virtual machines system , responsible for creation of virtual machine, submission of cloudlets (jobs) to the virtual machines and virtual machines destruction . It processes on the basis of of a provider not for users. Brokers submit the request to data center to which services user wants to use. Brokers are responsible for purchasing and selling cloud services.

6.4 Step – IV Create Data Center The data center is designed. In data center jobs or cloudlets are executed. Datacenter is sort of class in which hosts are type of virtualized and connected. It is responsible for handling of virtual machine (VM) queries rather than handling cloudlet queries. Cloudlet scheduler deals with cloudlet and VM (Virtual Machine) allocation policy handles processing of virtual machine. It also contains information about entire crowd network.

6.5 Step -V Submit work to data center Cloudlet loads information and ID of running virtual machine. Cloudlets are submitted in data center .In this step,the user send request to broker for machine, then broker provides virtual machine to user for execution of cloudlets. Broker places ID of virtual machine with its own ID, so that cloudlet resource can return to broker after the implementation. Cloudlet resources are scheduled by cloudlet scheduler. Cloudlet Scheduler is type of class that can be entitled as the programme of scheduling executed by a virtual machine. Hence, extended classes executing cloudlets and implementing a interface.

6.6 step- VI Run Algorithm Scheduler

6.6.1 Working Of Exiting Max-Min scheduling algorithm:

Algorithm [9]

1.for all submitted tasks in meta-task list Ti
2. for all resource Rj
3. compute Cij.
4. While meta-task is not empty
5. find the task Tk consumes maximum completion time.
6. assign task Tk to the resource Rj with minimum execution time.
7. remove the task Tk from meta-tasks set.
8. update rj for selected Rj
9. update Cij for all i

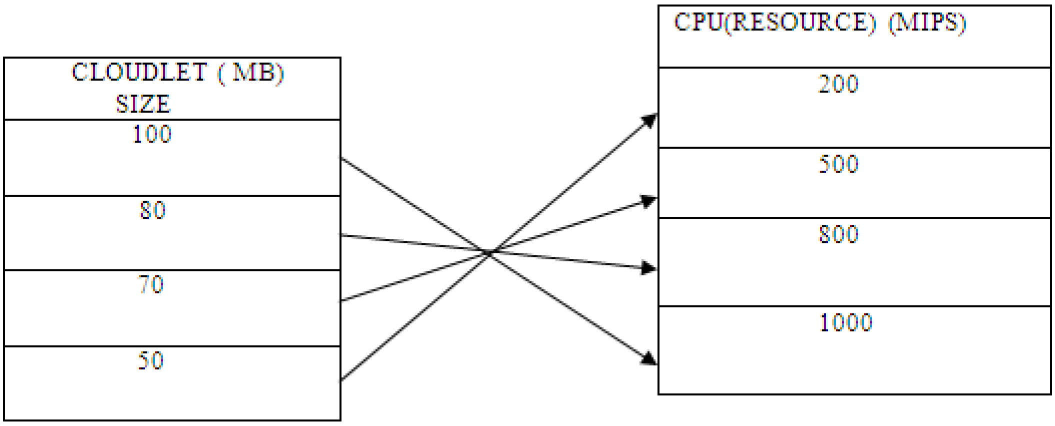


Fig.6.6.1 Working Of Exiting Max-Min scheduling algorithm

6.6.2 Illustration Max-Min scheduling algorithm: Max-min scheduling algorithm finds all unassigned jobs and then sorts jobs in order of job size and then find out the job with maximum size and assigns it to largest resource. It computes long job first after that small jobs are executed. For exp. In given figure 2 job size as cloudlet and CPU as resources are referred. It will use selection sort algorithm for assigning cloudlet to resources. It will pick the maximum job 100 MB and will assign to the resource of maximum mips 1000 and after that another cloudlet of size 80 MB will be picked up and assigns tos the next maximum resource of 800 mips. According to this other remaining maximum jobs(cloudlets) will be selected and allocated to remaining resources. At last cloudlet of minimum size of 50 will be assigned to minimum resource of 200 mips.

6.7 step- VII Improved Max-Min Scheduling Algorithm:

6.7.1 Algorithm:

1. Take all unassigned jobs or cloudlets (Ji1,Ji2….Jin).
2. For each cloudlet in job scheduler queue.
3. Estimate Cloudlet size (J).
   Estimate Completion time (CT).
   Estimate RAM (Memory) (M) .
   Estimate Payload Storage Rate (T)
4. Calculate resources proportional matching job (P).

Improved algorithm works on multiple parameters rather than single parameter as in case of conditional Max-Min algorithm. The improved algorithms also considers the completion time statistics as well as storage requirement as this algorithm is being optimized for data intensive specific application. The new algorithm designs weight to each parameters based on its proportion of contributing as a resource for excluding a particular jobs. The completion time statistics, however, are based on a particular virtual machine.

$: P=T^{\alpha} / R^{\beta}$

T denotes the payload storage rate potentially achievable for a particular virtual machine in the present time slot.
R denotes the heuristic average of payload storage of a particular virtual machine
α, β denotes the time the fairness schedule.
By adjusting α, β in the above equation, the balance between serving best virtual machine is found. The completion time is also considered, and calculated along with equation.

$: P=T^{\alpha}/R^{\beta}+CT$

All this is done, for the job/task/cloudlet having maximum size as it is based in Max-Min algorithm. It is also called Prioritization coefficients.

Working of Improved Max-Min Scheduling Algorithm

CLOUDLET NO.	CLOUDLET SIZE IN G bit	COMPLETION TIME (CT)IN SEC	PAYLOAD STORAGE (T) IN G bit/sec	RAM(R) IN GB
1.	5	5	250(mb)*1024	500
2.	4	3	130(mb)*1024	250
3.	15	12	530(mb)*1024	1000
4.	8	7	370(mb)*1024	750

Where α , β are variables.

Proportional fairness for cloudlet 1, α =.25, β = 1

$P=T^{\alpha}/R^{\beta} +CT$

Score 1 $=(250*1024)^{.25} \div 500+5=5.00$

Score 1=5.00

Proportional fairness for cloudlet 2, α =.75, β = .45

$P= (130*1024)^{.75} \div (250)^{.45} + 3=630.855$

Score 2=630.855

Proportional fairness for cloudlet 3, α=.45, β = 1

$P= (530*1024)^{.45} \div 1000+12=12.380$

Score 3=12.380

Proportional fairness for cloudlet 4, α= .65, β = .35

$P= (370*1024)^{.65} \div (750)^{.35} +7=423.652$

Score 4 = 423.652

6.7.2. Illustration for improved Max-Min algorithm: Improved Max-Min scheduling algorithm also considers completion time with storage block requirement. From given table cloudlet 1 of 5 G bit collects its proportional fairness (P) is 5.044 as given in improved Max-Min scheduling algorithm by applying formula of proportional fairness as given in improved Max-Min scheduling algorithm.

$: P= T^{\alpha}/R^{\beta}+CT$

Hence, score for cloudlet1 is 5.044.
Similarly, for cloudlet 2 of size 4 G bit proportional fairness (P) is 630.855
Hence, score for cloudlet for 2 is 630.855,
For cloudlet 3 of size 15 G bit proportional fairness (P) is 12.380
Hence, score for cloudlet 3 is 12.380
And, cloudlet 4 of size of 8 G bit proportional fairness (P) is 423.652
Hence, score for cloudlet 4 is 423.652

Priority table

Cloudlet size(G bit)	Cloudlet no.	Priority fairness(P)
4	1.	630.855
5	2.	423.652
15	3.	12.380
8	4.	5.044

6.8 Results and graphs:

The Improved Max-Min algorithm is implemented on an Intel core 5 on machine with 320 GB HDD and 4 GB Ram on 64 bit environment. CloudSim 3.0 is used for organizing the experiments. The processing speed of each cloudlet is indicated in mips (Million Instructions Per Second). The length of each cloudlet is termed as the number of instructions to be implemented. The two algorithms are observed by differ in the number of cloudlet and also arbitrary differ in the length of cloudlets. The number of VMs (Virtual Machine) used to implement the different cloudlets. Relative study of Improved Max-Min scheduling algorithm with existing algorithm present that the proposed Max-Min scheduling algorithm have better results and is more genuine scheduling algorithm. The following three graphs are observed.

6.8.1 Response time:

The response time of a cloudlet or job is described as the time taken when task is ready to execute to the time when it finishes its job.

Response Time = Arrival Time – Finish Time

The response time is compared with the existing scheduling algorithm MAX-MIN. The comparison is done with the number of cloudlets and by increasing the number of cloudlets slowly with response time. The results observed are shown in form of graph in figures:

(a)

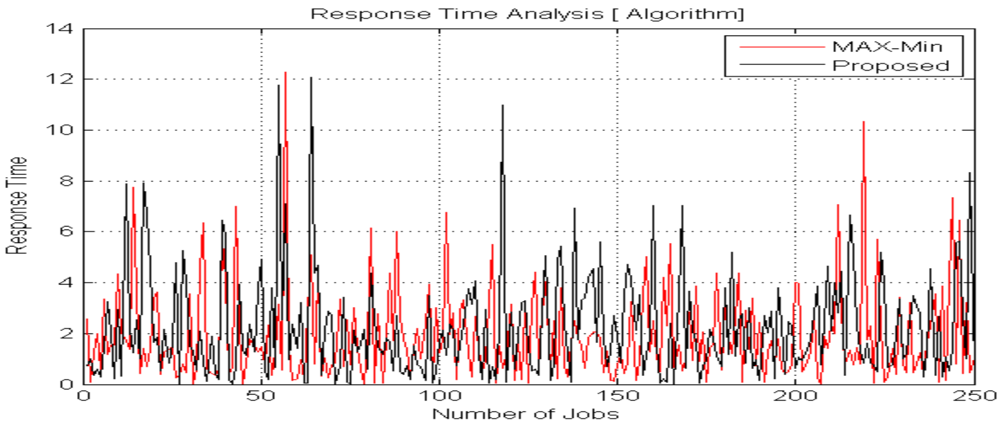


Fig.6.8.1(a) Line Graph

(b)

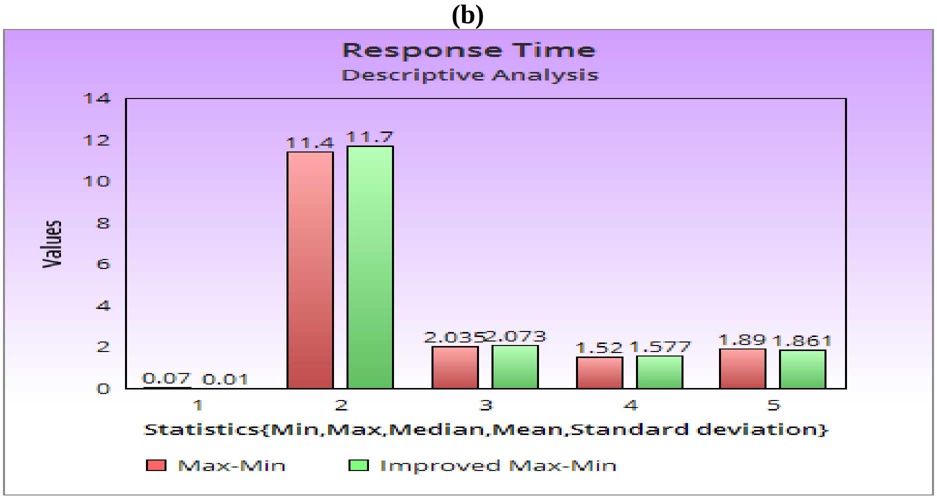


Fig.6.8.1(b) Bar Graph

Interpretation:

From the above graphs it is observed that response time of Improved Max-Min scheduling algorithm is lower than previous Max-Min scheduling algorithm. It is proved from the result values of above two graphs (Line graph and Bar graph).

6.8.2 Waiting Time:

It is the time taken when action is required or performed.

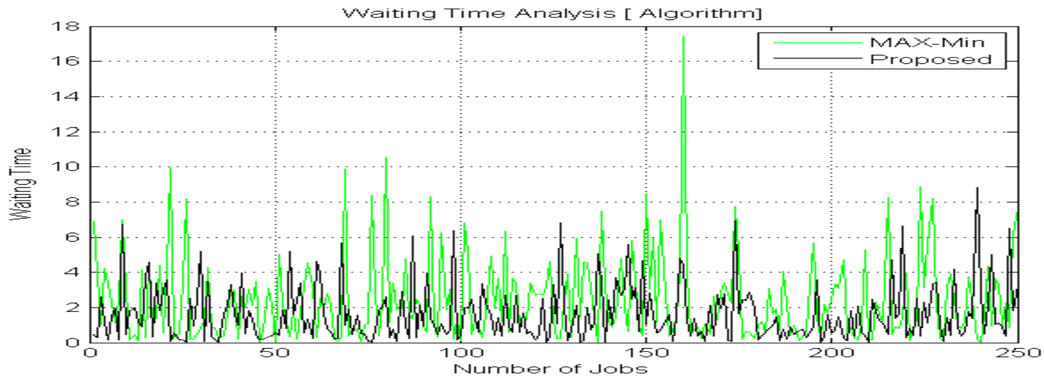


Fig. 6.8.2(a) Line Graph

b)

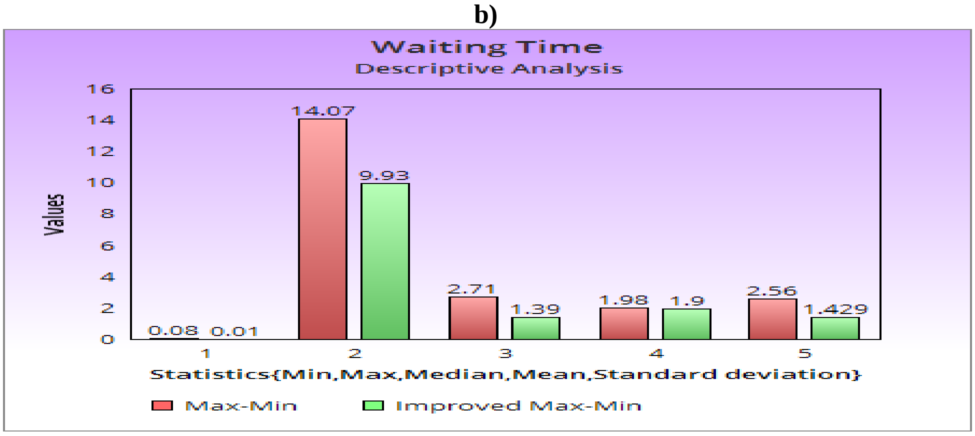


Fig. 6.8.2(b) Bar Graph

Interpretation**:** From the above two graphs of waiting time it is examined that waiting time of improved Max-Min scheduling algorithm is less than previous Max-Min scheduling algorithm. It is proved from the result values of two graphs of Line graph and Bar graph. Previous Max-Min scheduling algorithm is shown by red color and Improved Max-Min is shown by green color in bar graph.

6.8.3 Turn Around Time:

The turnaround time is the total time taken between the submission of a cloudlet for execution and the return of the complete output to the customer .Turnaround time mainly handles the total time it takes for a program to provide the required output to the user after the program is started.

Turnaround Time= Submission Time + Waiting Time + Execution Time

a)

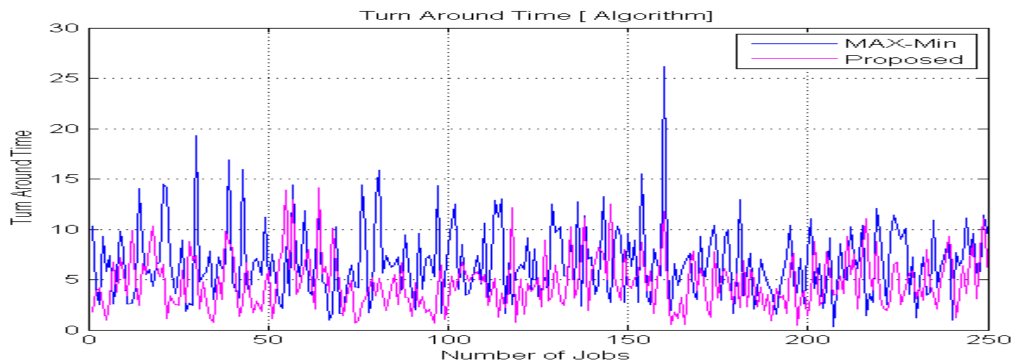


Fig.6.8.3(a) Line Graph

b)

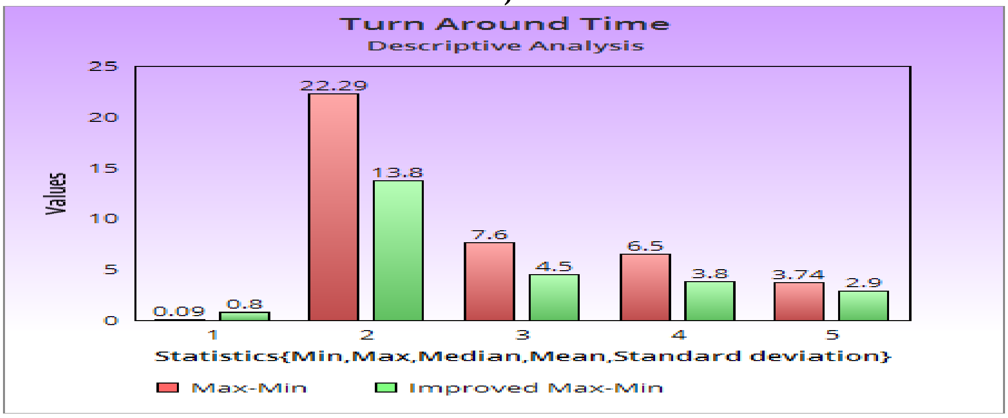


Fig.6.8.3(b) Bar Graph

Interpretation**:** Above two graphs for Turnaround time are shown. One is Line Graph and other is Bar graph. In Line graph previous Max-Min scheduling algorithm is shown by blue color and Improved algorithm is shown by pink color. Similarly, in Bar graph previous Max-min scheduling algorithm is shown by red color and improved algorithm is shown by green color. It is clear from the above values of two graphs that Turnaround time of improved Max-Min scheduling algorithm is less than as compared to previous Max-Min scheduling algorithm.

VII. Conclusion and Discussion

In this experiment work, CloudSim is used as a simulator for better performance of Improved Max-Min scheduling algorithm. In this research study, we have analyzed the exiting Max-Min algorithm [9] and it is experienced that it has some mistakes. Certain factors such as Completion time, RAM memory, Storage, Number of virtual machines, job size etc are examined by Improved Max-Min scheduling algorithm. Outcome shows that Improved algorithm performing better than previous Max-Min [9] in most of cases as shown in above from graphs [3],[4],[5],[6],[7] and in some cases of graphs value observed is above than average . Comparative study of our proposed algorithm with existing algorithm [9] show that the proposed algorithm have better choice and is more better scheduling algorithm.

Future scope: In research work, we need a workload that can be sub-divide before scheduling process taking place. There are many previous clustering algorithm which used to perform assembling to achieve this objective. Since, we need a different scheme to do a more advanced type of cloud assembling. Hence, to improve this research work in future, workload may use Neural Network and Artificial Intelligency.

References

[1]. Er. Shimpy and Mr. Jagandeep Sidhu," Different scheduling algorithm in different Cloud computing", In International Journal of Advanced Research in Computer and Communication Engineering Issue 9, September 2014 .

[2]. Yogita Chawla and Mansi Bhonsle ," A Study on Scheduling Methods in Cloud Computing", In International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) September – October 2012.

[3]. Pinal Salot ," A Survey of various scheduling algorithm in cloud computing environment",In International Journal of Research in Engineering and Technology Feb-2013.

[4]. Yogita Chawla and Mansi Bhonsle ," A study on scheduling methods in cloud computing", In International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) September – October 2012.

[5]. Shah Mihir and Asst. Prof. Yask Patel," A Survey Of Task Scheduling Algorithm In Cloud Computing",In International Journal of Application or Innovation in Engineering & Management (IJAIEM) January 2015.

[6]. Rajwinder Kaur and Navtej Ghumman," Hybrid Improved Max Min Ant Algorithm for Load Balancing in Cloud",In International Conference on Communication, Computing & Systems (ICCCS–2014).

[7]. Ullah Munir, E., J. Li and Sh Shi, 2007," QoS Sufferage Heuristic for Independent Task Scheduling in Grid", In Information Technology Journal, 6 (8): 1166-1170.

[8]. Weiwei Chen and Ewa Deelman ," WorkflowSim: A Toolkit for Simulating Scientific Workflows in Distributed Environments", In 8th International Conference on E-Science, IEEE Publication, 8-12 Oct. 2012.

[9]. S. Devpriya and C. Ramesh ," Improved Max-Min Heuristic model for task scheduling in cloud.", In International Conference on Green Computing, Communication and Conservation of Energy (ICGCE),IEEE ,2013.

[10]. Tarun Kumar Ghosh, Rajmohan Goswami, Sumit Bera and Subhabrata Barman," Load Balanced Static Grid Scheduling Using Max-Min Heuristic",In 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012.

[11]. Pardeep Kumar and A mandeep Verma, " Independent task scheduling in cloud computing by improved genetic algorithm, " In International Journal of Advanced Research in Computer Science and Software Engineering, May,2012.

[12]. Jing Wang, Gongqing Wu, Bin Zhang and Xuegang Hu," A heuristic algorithm for scheduling on grid computing environment ",In Seventh ChinaGrid Annual Conference, IEEE,2012,13.