Execution Time Enhancement Through Efficient Task Scheduling Algorithm in Cloud Computing Environment

Shital Sharma[1] Dr. Chirag Suryakant Thaker[2]
[1]P.G. Student [2]Assistant Professor
[2]Department of Information Technology
[1]GTU PG School, Gandhinagar [2]L.D. College of Engineering, Ahmedabad

Abstract— Cloud computing is depending on sharing of computing resources not on local servers or personal device to handle application. Cloud computing is made up of data-center's which handles large number of users. To handle all the client requests scheduling is most important. The scheduling of tasks in cloud means choose the best suitable resource available for execution of tasks or to allocate computer machines to tasks in such a manner that the completion time is minimized as possible. The aim of this paper is to represent the efficient scheduling technique which enhances the performance of scheduling and enhance the parameter value in cloud computing. For implementation purpose, we create the virtual environment in Oracle VM Virtual Box Manager.

Key words: Cloud Computing, Task Scheduling, Makespan, Max-Min Algorithm, Improved Max-Min Algorithm

I. INTRODUCTION

Lots of researchers try and outline cloud computing from completely different application aspects, but there is not an accord definition thereon. Cloud computing is a model for conveying and hosting services on the internet. Cloud computing can continuous process and multiple jobs from different users by providing scalable and virtualized resources as a service. The basic plan of cloud computing is to divide the pc program into several little subprograms through the IAAS to complete its task, so generate the results. The number of tasks in the cloud is huge and also the system is handling large tasks all the time, therefore the task programming is incredibly vital and tough. According to oxford schedule can be defined as a plan for carrying out a process or procedure. The person or a machine that organizes or maintains schedule is known as scheduler. Thus, scheduling can be defined as the process of defining the order in which the set of jobs to be completed. Scheduling contains a set of procedures to control the work load to be performed by the computer system. The tasks which are submitted by the cloud users need to be schedule properly on the resources for maximum utilization of the available resources. Scheduling is the process of arrangement of multiple tasks on particular time. Scheduling is the process of denotation of order in which multiple jobs are completed. It is defining that when an activity should start or end on its time. Using the different technique of scheduling we will able to arrange the task at their time and able to measure different parameters and get their better result [6] [9].

The main aim of this paper is to enhance the parameter value using proposed algorithm. Due to this we can reduce the parameter value and get better results of task scheduling in cloud computing environment. If task scheduling is done properly then the performance of cloud computing is also improve. So here we use are using a new algorithm to improve the performance of the cloud computing.

In this paper, we are going to discuss following sections, section II shows the work which is related to our work. In section III we are going to describe the proposed work flow and proposed algorithm. In section IV we derive the experimental results of proposed work and analyze the results and compare proposed work with existing one. At last section V concludes the whole paper.

II. RELATED WORK

Saeed Parsa et al. have proposed a new algorithm. In this the algorithm is built through a complete study and analysis of two popular task scheduling algorithm, Max-Min algorithm and Min-Min algorithm. Proposed algorithm will overcome the disadvantage of the both algorithm and generates a new better result. The proposed RASA (Resource Awareness Scheduling Algorithm) firstly calculate the completion time of the task of all available resources and then applies the Max-Min algorithm and Min-Min algorithms alternatively. RASA uses the Min-Min strategy to execute small tasks before the larger ones and Max-Min strategy to execute larger tasks to avoid delay and to support concurrency [1].

S. Devipriya and C. Ramesh have prepared a new improved algorithm which is based on concept of Max-min algorithm and RASA algorithm. Here the proposed improved Max-min algorithm is based on expected execution time instead of complete time. So, the scheduling of the tasks using Improved Max-min can achieve lower makespan rather than original Max-min. When the smaller task is higher than the larger task in meta-task, due to this problem the task completion time and makespan will increase. To overcome this problem author proposed the improved Max-min scheduling algorithm. In this algorithm, larger tasks are applied to the slower resources and smaller tasks are applied to faster resources. So that using this algorithm one can able to reduce the makespan [2].

O. M. Elzeki, M. Z. Reshad and M. A. Elsoud, have prepare a new algorithm to reduce the parameter value. Here the proposed algorithm which is improved max-min algorithm is focuses on task execution time not in task completion time as a selection basis. This algorithm proves that using improved max-min algorithm we can generate the lower makespan compared to RASA and Max-Min algorithm [3].

Upendra Bhoi and Purvi N. Ramanuj, have the main aim of this paper is modification of Improved Max-min task scheduling algorithm. This algorithm calculates the expected completion time of the submitted tasks on each resource. Then the Largest Task is assigned to a Slowest Resource. Then the scheduled task is removed from meta-tasks. All calculated times are updated and then apply max-min algorithm on remaining tasks. Before the larger task will apply to smaller resources, the average of the all tasks is calculated. Using that average value author will decide that which task will go first [4].

Bhavisha Kanani and Bhumi Maniyar, have study and explain about Max-min algorithm, issues related to this algorithm and solves the algorithm related problems. Proposed algorithm is used to reduce the makespan for important jobs and increase the resource utilization. User-priority was considered in this paper to fulfil user’s requirement quickly for the important job. This algorithm will first make a group of tasks and arrange them as their priority so the most important task will execute first. The proposed algorithm gives the priority as highest and lowest and normal priority tasks. In this paper, we calculate the resource cost to allocate the tasks to their resource according to their priority [5].

Santosh B. and Manjaiah D. H., have presents a new task scheduling algorithm which takes the advantages of MaxMin algorithm and perform based on Enhanced MaxMin algorithms. Here the resource set is divided into two groups. After the division of the resource, author will calculate the average of the tasks. The tasks which are just greater than the average length is selected those are belongs to the first division of the resources. The other tasks which has the maximum length are assigned to the other division of the resource. Here if the selected resource is belonging to the first set, selecting the largest task is same as MaxMin and else selecting the task greater than the average length of the task is same as Enhanced MaxMin Algorithm [6].

III. PROPOSED METHODOLOGY

Here we are introducing a new algorithm for task scheduling in cloud computing environment. Here is the flowchart and algorithm is given.

The algorithm is EETSAlgorithm (Execution time Enhancement Task Scheduling Algorithm). In this algorithm, first we take multiple tasks and create a set of it. Then we add the details of tasks in task information table. Now the new thing in this algorithm is we take a median value of the given tasks. So, using the median value we can able to divide the tasks in two divisions. The first division is small tasks and other one is larger tasks.

Now we have a median value of the given tasks so we divide the tasks into two parts like small tasks and large tasks and put them in descending order. After that scheduler bifurcate the resources according to its MIPS speed and assign the tasks to the resources according to task size. Now scheduler checks if there any task is remain in meta task or not, if there is any task is remains then scheduler assign it to the resource else this process will end.

Then scheduler selects the tasks from the small and large tasks alternatively and gives it to the resources. Here we bifurcate the resources like faster resource and slower resource according to its MIPS speed. So, the scheduler assigns the larger tasks to the faster resources and small tasks to the slower resources.

At last after assigning the task to the resources we remove the data of the tasks from the task information table. This is how proposed algorithm is works.

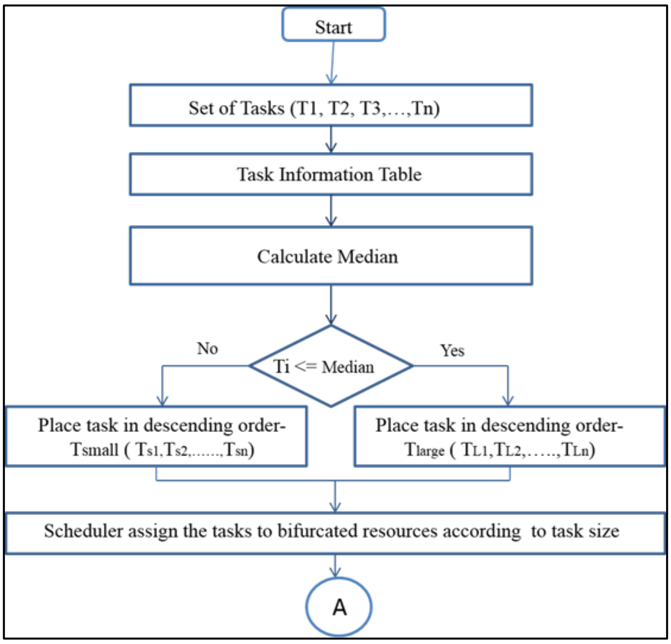
Start
Set of Tasks (T1, T2, T3,…,Tn)
Task Information Table
Calculate Median
Ti <= Median
No
Yes
Place task in descending order- Tsmall (Ts1,Ts2,......,Tsn)
Place task in descending order- Tlarge (TL1,TL2,…..,TLn)
Scheduler assign the tasks to bifurcated resources according to task size
A

Fig. 1: Proposed algorithm flow

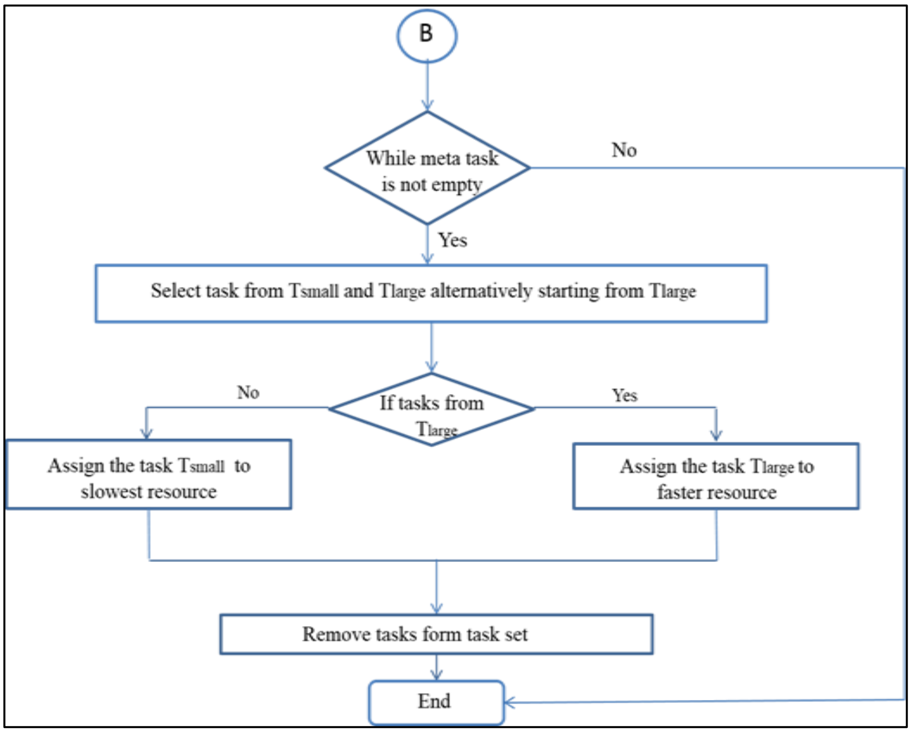
B
While meta task is not empty
No
Yes
Select task from Tsmall and Tlarge alternatively starting from Tlarge
If tasks from Tlarge
No
Yes
Assign the task Tsmall to slowest resource
Assign the task Tlarge to faster resource
Remove tasks form task set
End

Fig. 2: Algorithm

The EETSAlgorithm (Execution time Enhancement Task Scheduling Algorithm) is here,

```
Prepare Task Meta Data
For All Submitted Task Ti, Select Gmed.
        Gmed = Median(Ti)
Find all Gmed <=TAvg and place them in descending order in TSmall ArrayList
Find all Gmed >TAvg and place them in descending order in Tlarge ArrayList
Select task Tk from TSmall ArrayList and TLarge
  ArrayList alternatively, starting from TLarge ArrayList first,
  If Tk from Tlarge:
        Assign it to faster resources with minimum completion time
  Else if Tk from Tsmall:
        If Tlarge is not empty:
             Assign it to slower resources with minimum completion time
        Else:
             Assign it to resources with minimum completion time
        Remove Tk from Task meta data
If task meta data is empty
  End
Else go to step 5
```

IV. Experimental Result

According to the proposed Parameter Based Task Scheduling Algorithm here is the theoretical analysis is done which shows the performance of the system. Assume that the task scheduler has six meta-tasks and four resources as given below,

Task	T1	T2	T3	T4	T5	T6
Inst. Vol.	620	400	350	700	560	680

Table 1: Volume of Instruction in Task

Resources	R1	R2	R3	R4
Processing Speed	100	200	300	50

Table 2: Processing Capabilities of Each Resources

To calculate the median value, first we sort the task in ascending order, so we get T(median) = 590.

Task	T3	T2	T5	T1	T6	T4
Inst. Vol.	350	400	560	620	680	700

Table 3: Sorted volume of instruction in task

Using above given data in tables we calculated expected execution time of the task on each of the resource,

	R1	R2	R3	R4
T3	3.5	1.75	1.17	7
T2	4	2	1.33	8
T5	5.6	2.8	1.86	11.2
T1	6.2	3.1	2.06	12.4
T6	6.8	3.4	2.26	13.6
T4	7	3.5	2.33	14

Table 4: Expected execution time of each task on each resource

According to table-4 we are going to divide the task into two parts like T(large) and T(small),

T(large)	R1	R2	R3	R4
T4	7	3.5	2.33	14
T6	6.8	3.4	2.26	13.6
T1	6.2	3.1	2.06	12.4

T(Small)	R1	R2	R3	R4
T5	5.6	2.8	1.86	11.2
T2	4	2	1.33	8
T3	3.5	1.75	1.17	7

Table 5: Division of the task based on the Median value of the task

According to the proposed algorithm, the task nearer larger than or equal to the median value that is task T4 is allocated to the faster resource that is resource R3 as per the example. Now the scheduler will alternatively select the task from both the ArrayList T(large) and T(small) then starts from the T(Large) first, so it selects the task T4 and allocates it to the resource which gives minimum execution time that is resource R3. Secondly T(small) task is assign to the slowest resource. Similarly, all the tasks will be assigned to the resources as per the algorithm.

Oracle VM VirtualBox (formerly Sun VirtualBox, Sun xVM VirtualBox and Innotek VirtualBox) is a free and open-source hypervisor for x86 computers currently being developed by Oracle Corporation. We implemented our project in Oracle VM VirtualBox.

So, the experimental results show that using proposed algorithm we reduce the makespan comparing it with the MaxMin algorithm.

Proposed_algo_makespan	52.64000001049
Maxmin_algo_makespan	54.9800000191

Table 6: Comparison of Max-Min algorithm and proposed algorithm

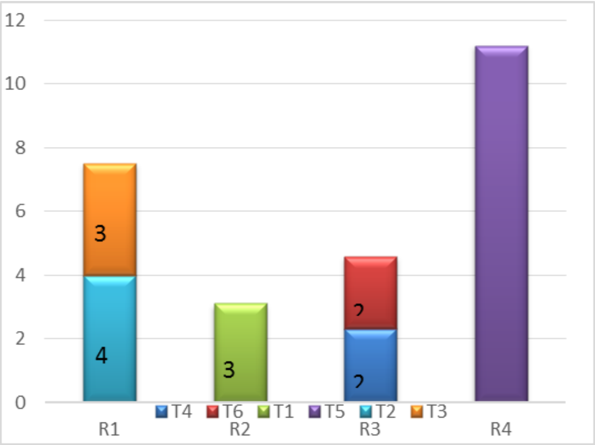

Fig. 3 : Chart 1: Allocation of task to the resources with the method of proposed algorithm

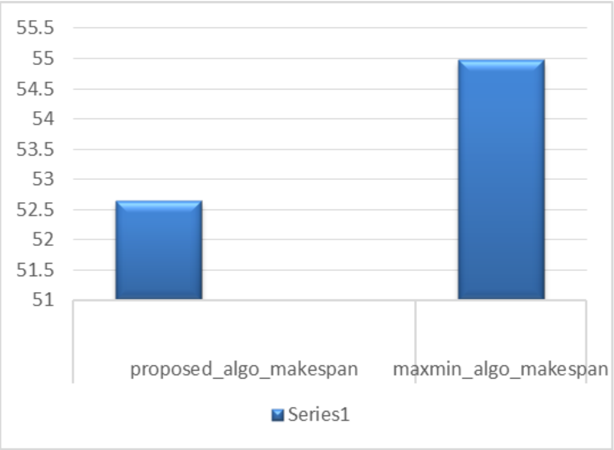

Fig. 4 : Chart 2: Comparison of (Proposed Algorithm and Max- Min Algorithm based on makespan

V. Conclusion and Future Work

After analyzing the various scheduling algorithms for cloud computing environment here is the new technique is generated. We proposed the new technique based on Max-min algorithm and Improved Max-min algorithm. When the number of large tasks is more than number of the smaller tasks in a meta-task, the Max-min algorithm schedules tasks. Due to this it will increase the parameter value (here we take make-span). So, we proposed a new algorithm for this situation, after using this algorithm, we enhance the parameter value and increase the performance of the task scheduling in cloud computing environment. The performance of the proposed algorithm is also compared with the other existing scheduling algorithm like Max-min algorithm. In future, this algorithm can be implemented in different cloud computing tools and the other parameters can be measured.

ACKNOWLEDGEMENT

The author would like to thank to his thesis guide Dr. Chirag Suryakant Thaker for their great effort and perfect guideline. I also want to thank Mr. Nayan Patel who helped me during my research work and all other peoples who help me for this research work.

VI. REFERENCES

[1] Saeed Parsa, Reza Entezari-Maleki, “RASA: A New Task Scheduling Algorithm in Grid Environment” in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009 ISSN 1818.4952 ©IDOSI Publications, 2009.
[2] O. M. Elzeki, M. Z. Reshad and M. A. Elsoud, “Improved MaxMin Algorithm in Cloud Computing”, International Journal of Computer Applications (0975 – 8887) Volume 50 – No.12, July 2012.
[3] Upendra Bhoi and Purvi N. Ramanuj, “Enhanced Max-min Task Scheduling Algorithm in Cloud Computing” in International Journal of Application or Innovation in Engineering and Management (IJAIEM).
[4] Upendra Bhoi and Purvi N. Ramanuj, “Enhanced Max-min Task Scheduling Algorithm in Cloud Computing” in International Journal of Application or Innovation in Engineering and Management (IJAIEM).
[5] Santhosh B, Manjaiah D H, “A Hybrid avgtask-Min and Max-Min Algorithm for Scheduling Tasks in Cloud Computing” in 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT).
[6] Pinal Salot, “A survey of various scheduling algorithm in cloud computing environment” in IJRET: International Journal of Research in Engineering and Technology ISSN: 2319-1163.
[7] S. devipriya and C. Ramesh, “Improved max-min heuristic model for Task scheduling in cloud” in 2013 IEEE.
[8] Soheil Anousha, Mahmoud Ahmadi, “An Improved Min-Min Task Scheduling Algorithm in Grid Computing” in Springer-Verlag Berlin Heidelberg 2013.
[9] Arnav Wadhonkar, Deepti Theng, “A Survey on Different Scheduling Algorithms in Cloud Computing” in International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB16).