International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

An Improved Longest Approximate Time to End Algorithm using Dynamic Cloud Sim

Kirandeep Kaur[1], Khushdeep Kaur[2]

[1, 2]Bhai Maha Singh College of Engineering, PTU

Abstract: ***Longest approximate time to end scheduling algorithm sorts all the cloud jobs in queue according to their time taken in execution and other time base performance parameters. However, in many ways for scheduling analysis, planning makes a lot of difference in performance and in removing bottlenecks before final execution. But for these additional parameters are needed to consider. Therefore, in this research work parameters like inter node bandwidth, traffic, congestion have been used for building a plan for scheduling, simulated results show significant improvements in performance is concerning of time performance parameters.***

Keywords: Planning, Workflow, Cloud Computing, LATE Algorithm, Improved LATE

1. Introduction

Planning is a procedure, in which workflow activities are organized to balance management and performance of a process. Planning strategy performs before the scheduling. Workflow Sim [7] is an open source workflow simulator that was firstly developed by Weiwei Chen. It extends Cloud Sim by applying a workflow level support of simulation.

It contains three layers: Workflow planner, Workflow engine, Workflow scheduler. Algorithms that help to plan the workflow are known as planning algorithms or planners. Planning algorithms display the global view of workflows to the cloud users that can tie any task to any resource. Planning of workflow is required to plan correct, consistent workflows achieve the better result performance in workflow designing and execution, which makes the cloud computing more effective and latest popular technology developing sharply. Workflow engine releases free tasks to workflow scheduler that matches these free tasks to condor virtual machine in Workflow Sim and submit them for execution. Workflow scheduler can only bind free tasks to available resources.

The four constituents of cloud that are Cloud user, Broker, Physical Machines and VM's. The cloud users can make their service requests from any location in the world to cloud to fulfill their service requirements. A cloud data centers include physical machines. VM's are created on the top of physical machines by using virtualizing of technology. The broker works as a mediator between cloud data centers and cloud users. It assigns cloud resources to client's workflow implementations [2].

There are different existing planning algorithms for workflow designing and execution. But many of these existing algorithms avoids inter node traffic, congestion, inter node bandwidth while plans, whether there is smooth flow of traffic and congestion between VM's, it has not been considered which is crucial or simply, most of the algorithms work in a few parameters, especially for the analysis before execution. There are a large number of chances to enhance the reliability of algorithms to increase their performance. To refine the performance of previous algorithms, there should need to work on inter node bandwidth, traffic, congestion like parameters. Hence, in the next section, a survey of this issue is conducted.

2. Related Work

There are large numbers of existing planning algorithms to design and execute the workflow in distributed cloud environment. But these planning algorithms consider a limited number of factors while taking decisions. Hence, many of these can be enhanced further to increase productivity.

Some of the works done in this area are discussed below:

In this paper, [Chen Weiwei et.al., 2011] the authors have used workflow planning, execution logs gathered from Pegasus and Condor to analyze overheads for set of workflow runs on cloud and grid platforms.

In this paper, [Marc Bux, Ulf Leaser et.al, 2013] the authors introduced a new simulation toolkit Dynamic Cloud Sim extension of Cloud Sim. They enhanced characteristics like instability, Dynamic changes of performance at run time, Inhomogeneity and failure during task execution. But they unable to find the issues like data locality.

In this paper, [Sergio Esteves, Luis Veiga et al, 2014] the authors introduced a new scheduling model like cord model for a cloud workflows to remove the challenges like low data processing rates, low resource efficiency rates and also propose the novel service-oriented scheduler planner for the continuous data processing workflow. They also developed WaaS (Workflow as a service) workflow coordinator system for the cloud to share the data. But it does not take care of parameters like memory, job size, internode bandwidth etc.

In this paper, [M Zaharia et. al., 2008] the authors introduced a LATE scheduling algorithm extension of the Hadoop default scheduling algorithm, which works for the alteration of default queue. LATE sort all the jobs in the queue according to their time taken to reduce the time. But there is no consideration for planning and analysis before execution.

In this paper, [Ewa Deelman et. al., 2014] the authors introduced a PEGASUS system having capabilities like to

achieve reliable and scalable workflow for the evaluation of complex science systems. This paper describes how Pegasus attains scalable, reliable workflow execution across a wide variety of computing environments.

3. Simulation Parameters

Number of Virtual Machine/ Host	5
Internode Bandwidth	1.5e7
RAM	512MB
Workload Type	Scientific
Hard disk	1000MB

4. Research Gap

Existing algorithm like HEFT [4] LATE [4] ignores to building a solution for internode traffic, congestion, internode bandwidth and task (memory to data processing) while planning workflow distribution to virtual machines whether there is congestion or smooth flow traffic in between virtual machines. Hence there are large numbers of chance to improve the reliability of algorithm to increase its smooth flow. So there is a need to work with parameters that will responsible to improve result performance by making the improved algorithm in the concept of the HEFT planning algorithm.

5. Problem Formulation

The LATE scheduling algorithm basically works on principle of sorting jobs based on long time taken by job for execution or farthest in the future. This algorithm can be implemented as a planning also, where before execution, pre analysis along more parameters may be used.

6. Scope of work

Based on the research gap and problems, the discussed scope of work may be defined as follows.
Develop LATE [4] based scheduling algorithm based [4] and conduct/record observations.
Developed [4] improved algorithm based [4] and conduct /record observations.
Evaluate performance of the algorithm based on observations recorded.

7. Methodology

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

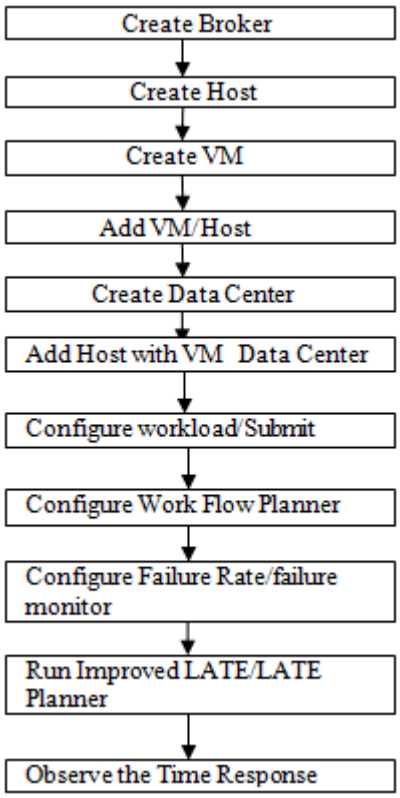

Figure 7: Block Diagram

7.1 Create Broker

In that step, firstly we build a datacenter broker. The Net Datacenter Broker represents a broker acting on behalf of Datacenter provider. It hides virtual machine management, as virtual machine creation, submission of cloudlets to this virtual machine and destruction of virtual machines. In our case the data broker will be submitting scientific workload. It works on behalf of a provider not for users. One has to implement interaction with user broker to this broker.

It is an object that acts as an intermediate service provider and cloud user. It helps to the cloud users in the selection of services of cloud. Broker submits the request to the data center, which cloud user wants to use. In our case the data broker will be submitting scientific workload. It facilitates the work distribution between the distinct cloud service providers. It provides the facility to the customers about how to use the cloud computing services to attain the appropriate business targets.

7.2 Create Host

In that step, we have added a host (physical machine). The network host class expanded host to assist simulation of networked data centers. It completes activities concerned with management of packets (send and receive) other than that of VM's (e.g., creation and destruction). A host has a defined policy for providing memory and bandwidth, as well as an allocation policy for processing elements to virtual machines.

7.3 Create Virtual Machine

In that step, create a virtual machine. Network virtual machine class extends virtual machine to support simulation of networked data centers. It executes activities concerned with packet management. It works as a helper for the small and medium organizations to that requires reliable and a number of services at lower cost rates. It permits the scalability without adding the physical resources and can be extended more easily than in the physical machines. These deliver the appropriate solutions for backup and recovery.

7.4 Add VM/Host

In that step, virtual machine and host are added which are already created.

7.5 Create Data Center

In this step, the Datacenter developed. Datacenter consists of host and virtual machine, whose host lists are virtualized and networked. It consists of the information of internal networks. It stores the data and delivers services**.** Datacenter class is a Cloud Resource whose host Lists are virtualized. It concerned with processing of VM queries (i.e., handling of VMs) rather than of processing cloudlet related queries. It is the centralized repository that connects applications, servers and storage services. Enterprises depend upon their data centers to derive the business operations with greater efficiency. Data centers want to be planned and managed carefully to fulfill the user objectives. Data center diminishes the demands for hardware by "time sharing" clients on the same hardware platform with the use of virtualization.

7.6 Add Host with VM Data Center

In that step, host added with datacenter of virtual machine.

7.7 Configure workload/Submit Workload

In this step path of files is configured and submits the workload to the planner. It is a set of processes that can be componentized individually performed upon and evolve a determine result with the abstraction being above the network, hardware and evaluate the performance of the planner to get the optimum results.

7.8 Configure Work Flow Planner

Parameters.Planning Algorithm pln_method = Parameters. Planning Algorithm. ILATE;

7.9 Configure Failure Rate

It collects the failure occurs during the running of algorithms.

7.10 Run Improved LATE/LATE Planner

Pseudo code of previously existing LATE algorithm:

1. For each job (J1, J2, J3)
2. Estimate time ⟶ T1 as a time parameter
3. Sort ⟶ jobs in descending order
4. Pick farthest job ID
5. Plan longest job
6. Execute each job as their time taken for execution.

Working of LATE:
Virtual Machine 1

Table 1: LATE

Jobs	Time taken for execution (in nanoseconds)
Job1	**2.5ns**
Job2	**2.1ns**
Job3	**2.7ns**
Job4	**1.9ns**
Job5	**2.9ns**
Job6	**3.0ns**
Job7	**3.1ns**
Job8	4.3ns

Virtual Machine 2

Table 2: LATE

Jobs	Time taken for execution (in nanoseconds)
Job1	2.3ns
Job2	2.6ns
Job3	2.9ns
Job4	1.1ns
Job5	2.3ns
Job6	3.1ns
Job7	3.2ns
Job8	4.4ns

LATE always try to complete the jobs first, which takes the longest time for their execution. It speculatively executes the task which are farthest in the future. For example: We have taken two virtual machines, virtual machine1 and virtual machine2. In the table no. 1 job8 takes 4.3nano seconds to complete its execution. Hence, LATE first try to sort job8 in descending order. After the execution of job8, it executes the job7 which takes 3.1 nanoseconds to complete its execution. After that it executes job6, which takes 3.0 nanoseconds to complete its execution. Then it sorts the remaining jobs in descending order according to their time taken for execution.

Pseudo code of proposed algorithm:
1) Take jobs (J1, J2, J3)
2) For each job in job planner
   {Estimate internet bandwidth (IBw)}
   {Estimate memory 'm' in MIPS}
   {Estimate ratio values}
3) Calculate score of jobs in descending order.
4) Execute planned job, having the highest score

Working of ILATE:
Virtual Machine1

Table 3: ILATE

Job	Longest time taken (in nano seconds)	Internode bandwidth (in Mbps)	Memory load/Bits (512*RAM/megabits)	Congestion/ traffic (Bandwidth/megabits)
Job1	1.1	220	512*8/2= 2048	220/2=110
Job2	3.2	412	512*8/1=4096	412/4=103
Job3	3.1	250	512*8/4=1024	250/3=83.33
Job4	3.0	315	512*8/5=819.2	315/5=63
Job5	4.1	415	512*8/6=682.7	415/7=59.3

Job score: 4.1*415*4096*110=766627840

Virtual Machine2

Table 4: ILATE

Job	*Longest time taken (nano seconds)*	*Internode bandwidth (in Mbps)*	*Memory load/Bits (512*RAM/ megabits)*	*Congestion/traffic (Bandwidth/ megabits)*
Job1	2.1	240	512*4/3=682.66	240/1=240
Job2	4.2	350	512*4/4=512	350/2=175
Job3	4.5	215	512*4/1=2048	215/4=53.77
Job4	1.2	218	512*4/4=1024	218/3=72.66
Job5	4.0	308	512*4/8=256	308/6=51.33

Job score: 4.5*350*2048*240=774144000

In this illustration two virtual machines, virtual machine1 and virtual machine2 are taken as shown above table3 and table4. As per the new algorithm, new parameters are calculated for calculation. Job1 up to Job5 entries in the given tables3 and 4 are computed. The calculation of the factors like longest time taken, Internode bandwidth, memory load, traffic and congestion are done. As per simulation configuration, the value of the internet bandwidth varies from 100 GB-1TB for virtual machines. The RAM configuration varies.

8. Results

The proposed longest approximate time to end algorithm implemented on Intel core3 with 3GB RAM on 32 bit operating systems. The observations are carried out by Cloud Sim3.0 simulator. The speed of PE's (processing elements) indicated in MIPS (Million Instructions per Second) and the length of cloudlets indicated as the number of instructions to be executed. The algorithms are tested by varying the number of cloudless and also randomly varying the length of cloudlets. Also, the number of VMs used to execute the cloudlets, are varied accordingly. Comparisons of our improved algorithm to previously existing algorithm show that proposed algorithm has better results performances and more reliable.
Following graphs are examined for results analysis:

Response Time: It is time when the task is ready to execute to the time when it finishes the task.
Response Time= Arrival Time- Execution Time

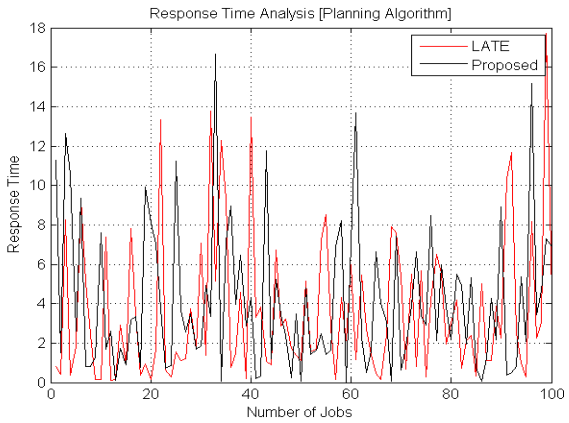

Figure 8(a)

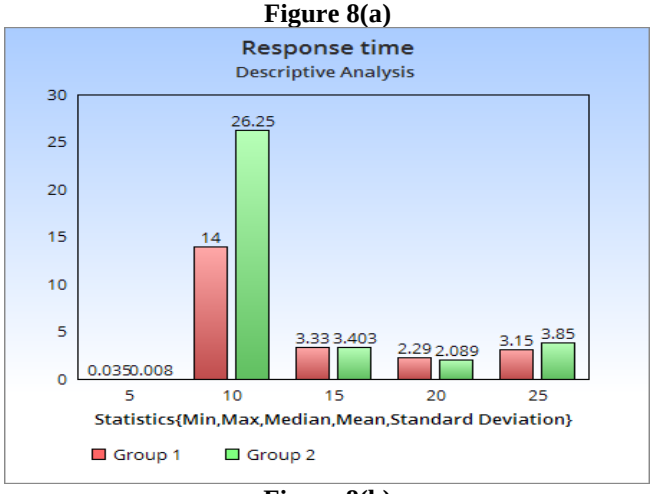

Figure 8(b)

Analysis: The above graphs show the comparisons of the improved algorithm with an existing algorithm by increasing the number of jobs with respect to the response time. It shows that the proposed algorithm's response time less than the previous existing algorithm. It is clear from the range LATE and ILATE in line graph and minimum, maximum values and their mean, median, standard deviation values is shown in the bar graph of the response time given above.

Waiting Time: Waiting time is the time when action is required or it occurs.

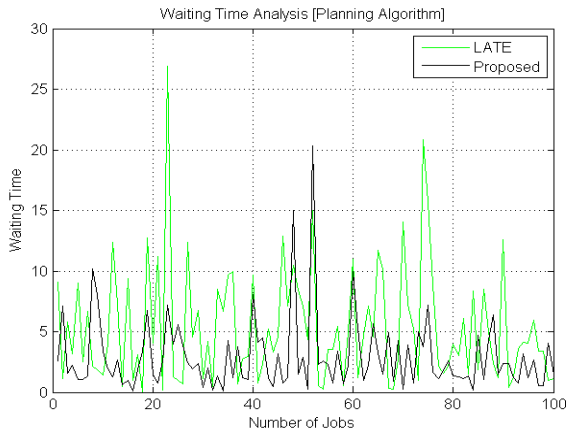

Figure 8(c)

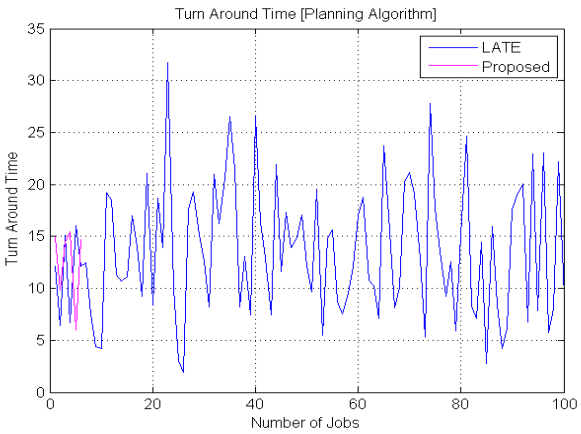

Figure (8e)

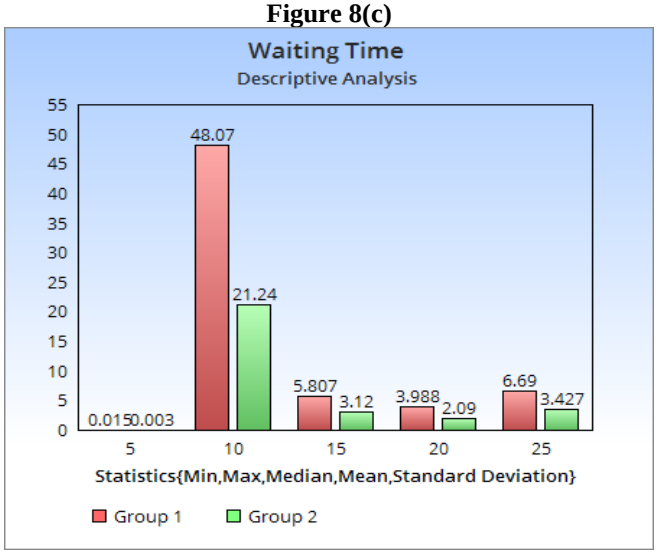

Figure 8(d)

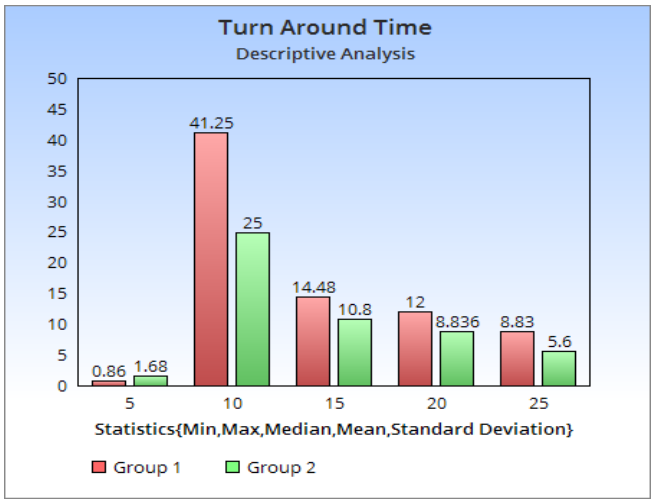

Figure 8(f)

Analysis: The above graphs show the comparative analysis of the proposed algorithm with existing algorithms. Observations are performed between the number of jobs and waiting time by increasing the number of jobs. From the above illustration it is clearly shown that improved planning algorithm's waiting time lower than the previous algorithm and has better results performances and more reliable. It is clear from the LATE and ILATE range in line graph and the minimum, maximum values and their mean, median, standard deviation values shows in the bar graph of the waiting time given above.

Turnaround Time: It is the total time taken between the task submission for execution and the return of the complete output to the user.

Turnaround Time= Submission Time + Waiting Time + Execution Time

Analysis: The above comparative analysis shows that the proposed algorithm's turnaround time less than the existing LATE scheduling algorithm. It is clearly shown from the LATE and ILATE range in line graph and the minimum, maximum values and their mean, median, standard deviation values shows in the bar graph of the turnaround time given above.

9. Discussion and Conclusion

In this research paper, we conclude that LATE algorithm [6], assured that it has a number of imperfections. The research work proposed to implement their scheme. The improved planning algorithm based on LATE algorithm takes a number of parameters into account such as inter-node bandwidth, memory to processing ratio, traffic, congestion, etc. Results show that proposed algorithm works well as compared to the previous in most cases of time, more reliable. The proposed algorithm uses the custom sort criteria for tasks is based on execution time and resources. It is shown from the graphs related to response time figure [8 (a), 8 (b)] that improved planning algorithm's response time less than the previous algorithm and has better results performances and more reliable. It is clear range LATE and ILATE in line graph and minimum, maximum values and their mean, median, standard deviation values is shown in the bar graph of the response time. Also clears from the second type of graphs [8(c), 8(d)] based upon waiting time

and third type of graphs [8 (e), 8 (f)] related to turnaround time that that improved planning algorithm's waiting time and turnaround time less than the previous algorithm and work better in most cases of time, has better results performances and efficient reliability respectively. It is clearly shown from the range LATE and ILATE in line graph and minimum, maximum values and their mean, median, standard deviation values are shown in the bar graph of the response time. But in some cases, the average come above.

10. Future Scope

The future work to be carried out under the current research work should involve workload that is partitioned or divided before planner takes on the workload for planning. However, for future scope, we suggest a probabilistic graphical model approach to make a new type of graphical models.

References

[1] WeiweiChen, EwaDeelman, "Workflow Overheads Analysis and Optimizations", 2011.
[2] Renu Bala, Gagandeep Singh, "An Improved Heft Algorithm Using Multi- Criterian Resource Factors", 2014.
[3] Long, W., Yuqing, L., and Qingxin, X. (2013), "Using CloudSim to Model and Simulate Cloud Computing Environment", 9th International Conference on Computational Intelligence and Security (CIS), IEEE Publication, 14-15 Dec. 2013, Leshan, pp. 323-328.
[4] Marc Bux, Ulf Leaser, "Dynamic Cloud Sim: Simulating Heterogeneity in Computational Clouds, IEEE Publication, 2013.
[5] Esteves, Luis Veiga et al, "Planning and Scheduling Data Processing Workflows in the Cloud with Quality-of-Data Constraints", Springer publication, 2014, pp 324-338.
[6] Ewa Deelman, "Pegasus, a Workflow Management System for Science Automation", 2014.
[7] M Zaharia et. al, "Improving MapReduce Performance in Heterogeneous Environments, in: Proceedings of the 8th USENIX Symposium on Operating Systems Design and Implementation, San Diego, USA, pp. 29-42,2008.
[8] Weiwei Chen, Ewa Deelman, "Workflow: A Toolkit for Simulating Scientific Workflow in Distributing Environments", IEEE publication, 2013.
[9] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, J. Myers, Examining the Challenges of Scientiic Workfows, IEEE Computer 40(12):24-32, 2007.
[10] Saeid Abrishami , Mahmoud Naghibzadeh, Dick H.J. Epema, " Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds",2012.