Time and Load based Cloud Scheduling Algorithm

Parampreet Singh Jaura
M.Tech Student
Lovely Professional University
Phagwara, India

ABSTRACT

Due to the fast development of internet, a huge amount of load increases over data centers every second. This causes scheduling overhead, huge memory demand at data centers. Thus increases overhead effects the load balancing at data centers. So, there is a need of mechanisms which will decrease overhead and provide effective load balancing. Today, every load balancing scheduling algorithm balances the load on data centers that reside in the same region. They give birth to same problems like scheduling overhead, huge memory demand. This paper proposes a Load balancing scheduling algorithm which is based on load and time. This algorithm balances the load over the Data centers which reside in different regions. This mechanism will maximize hardware utilization, decrease huge memory demand and decrease cost.

Keywords

Cloud Computing, Load Balancing, Time based, Scheduling

1. INTRODUCTION

Cloud Computing is the hottest field in today's world. Every large and small organization eagerly wants to adopt the cloud computing services. Because it helps reduce the carbon footprint and also the extra space. Cloud Computing is denoted as a blend of interdependent and virtualized computers that are forcibly provisioned and represented as single or multiple integrate computing assets [1].Cloud computing works on Pay per use (i.e. you have to pay only for that what you have needed and used) [2]. In elastic computing, the workload is handled dynamically by variation in cloud resources. Its aim is to provide the exact resources which are required by the end user and the amount he paid for [3] [4]. Elasticity implicit the capability to shift and pool resources among dissimilar infrastructure so that data requirements and resource availability can be kept more in synchronization, avoiding the lavish practice of over-provisioning [5] [6]. Amazon Elastic Compute Cloud (EC2) is a component of Amazon web services and Amazon's cloud computing platform. It provides virtual computers on rent basis to the users in which they can develop and run their own applications. It provides most valuable service i.e. scalability [7]. A user can scale his/her resources anytime by paying for the scaled resources what he/she wished for. EC2 provide web service with the help of which consumer can boot an Amazon device image to construct virtual machine [8] [9].

1.1 Load Balancing

Load balancing is a method in which assignment is distributed over different systems to accomplish optimal assets utilization, minimalize response period and evade overload [10] [13]. Load balancing algorithms are used to distribute load on various systems. These algorithms can divide into two major categories:-

1.1.1 System Load

- Centralized approach: A particular node manages dispersal within whole scheme.
- Distributed approach: In this, multiple nodes manages distribution by collecting load information from each other and builds their own load vectors. This method is most appropriate in cloud computing.
- Mixed approach: In this, combination of centralized and distributed approach is used [13].

1.1.2 System Topology

- Static approach: It is clear at proposal phase of scheme (i.e. not change dynamically during load balancing)
- Dynamic approach: It focuses on present state of a scheme throughout load balancing. It is most fit in cloud computing.
- Adaptive approach: In this method, when system state changes it adapts the load distribution according to it [13].

1.2 Scheduling

Scheduling is a method in which tasks are schedule for achieving high performance with minimum effort. In distributed systems, scheduling algorithms effort on exploiting the utilization of resources while lessening the task execution time by spreading loads on processors [14]. There are different types of scheduling algorithm that occur in distributed structures. Some of them can be practical in cloud computing location but outdate scheduling algorithms are unable to deliver programing in cloud [11] [12]. In cloud, job scheduling algorithms are divided into two main categories:-

1.2.1 Batch mode heuristic scheduling algorithms

In this, jobs are queued in a system and then scheduling algorithm starts afterward a static period of time. For e.g. FCFS and RR [14].

1.2.2 Online mode heuristic scheduling algorithms

In this, tasks are scheduled when they attain at the system. There is no queuing of jobs and no fixed time for a scheduling algorithm to start. For e.g., MFTF [14].

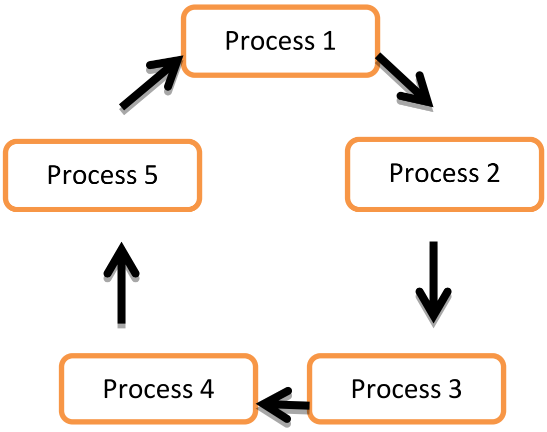
The rest of this paper is organized as follows: Section 2 discusses background and related work. Section 3 describes methodology. Section 4 describes simulation and result analysis. Section 5 concludes paper and proposes future research directions. Section 6 describes the References.

2. BACKGROUND AND RELATED WORK

Load balancing is used to avoid excess overload on the resources and also to reduce maximum response time. Until now, existing load balancing mechanisms balances load within the same region. These mechanisms work well during normal hours of traffic but during peak hours, when traffic come in huge bursts then existing mechanisms performance reduces to poor level [13] [15]. Following are some existing load balancing techniques that used in cloud environment.

2.1 Round Robin Algorithm (RR)

It is the simplest and easiest used algorithm. In this, jobs or procedures are separated into all processors on the basis of round robin order [13]. Workload distribution is equally divided between processors but job processing time varies according to job size. The following figure shows how round robin works. Figure shows that each user request is assisted by every processor in particular time quantum. When the time slice is over, the succeeding line up user request will arise for execution. If the user request ends in time quantum then user would not wait else user has to wait for its next period [15].



Execution of processes within Time Quantum in Circular Queue

Figure 1: Round Robin Scheduling Process

2.2 Equally Spread Current Execution Algorithm (ESCE)

It is the most preferred load balancing technique now days. In this, jobs are distributed according to the priorities via Load Balancer [13]. It transfer burden to that practical mechanism which is free or idle or handle task easily. Following figure shows that different clients job requests will queued in a job pool. There is a Virtual Machine Load Balancer that chooses the jobs according to the priority and provide that jobs to virtual machines which are free or idle [15].

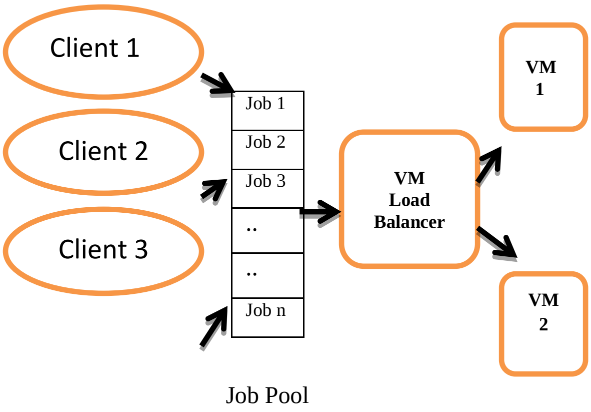


Figure 2: ESCE Scheduling Process

2.3 Throttled Load Balancing Algorithm(TLB)

In this, the load balancer maintains an index table of virtual machines with their state (i.e. Available or Busy). It will distribute jobs to that specific virtual machine which is most suitable for processing that job [13]. Following figure shows that the process first begins by keeping a list of every VMs available. Each row is separately indexed to speed up the lookup method. If a match is found on the base of size and accessibility of the machine, then the load balancer takes the request of the client and assigns that VM to the client. If there is no VM existing that matches the standards then the load balancer returns -1 and the demand is queued [15].

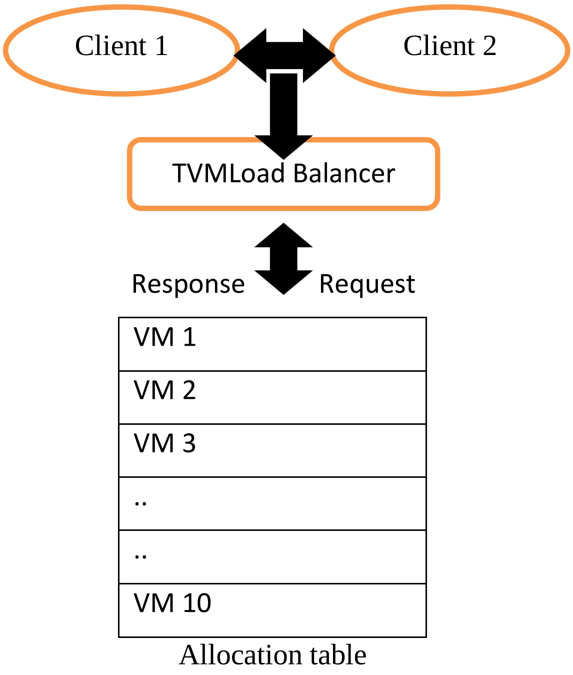
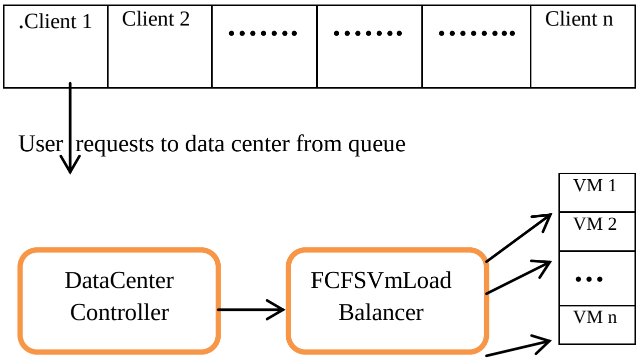


Figure 3: Throttled Scheduling Process

2.4 First Come First Serve (FCFS)

It is a simple task ordering strategy and used in parallel task processing. It selects and processes tasks according to the right order of jobs getting into system [13] [15]. With this scheme following figure shows that the user request which arises first to the datacenter controller is allotted the virtual machine for implementation first. The datacenter controller examines for virtual machine which is in idle state. Then the very first request from the queue is eliminated and forward to unique VM through the VMLoadBalancer



Allocation of virtual machines according to arrival time

Figure 4: FCFS Scheduling Process

3. METHODOLOGY

3.1 Existing Logic and Algorithm

In this phase, different load balancing algorithms i.e. Round Robin (RR), Equally Spread Current Execution (ESCE), Throttled was examined. These existing load balancing algorithms balance the load between different datacenters within same region.

3.1.1 Logic

Existing load balancing algorithms based on only one parameter i.e. Load and they shift the load within same region. These algorithms works well in average and minimum hour when traffic is moderate. But during peak hours when traffic becomes burst then at that time Scheduling Overhead occurs, Response Time increases, Data Centre Processing Time and Cost increases [11] [12]. At that time there is a need of setting up new data centre within the region that can balance the load during peak hours of traffic. But setting up new data centre takes huge cost and time.

3.1.2 Equally Spread Current Execution Algorithm

1. Equally Spread maintains an index table of virtual machines and number of requests currently assigned to a VM. It also maintains state of the VM (Busy or Available).
2. If Current VM is available then it will assign the task to it for processing and if VM is in Busy state then it will try to Find the next available VM.
3. If available VM is not allocated create a new one
4. Count the active load on each VM
5. Return the id of those VM which is having least load.
6. The VM Load Balancer will allocate the request to one of the VM.
7. If a VM is overloaded then the VM Load Balancer will distribute some of its work to the VM having least work, so that every VM is equally loaded.
8. The datacenter controller receives the response to the request sent and then allocates the alignment requests from the job pool/queue to the available VM & so on.
9. Continue from step-2.

3.2 Proposed Logic and Algorithm

3.2.1 Logic

Proposed logic works on two parameters i.e. Load and Time. It mainly focuses on time based scheduling of tasks. In this, load will be shifted on datacenters which resides in another region on the basis of time zone. This world is divided into time zones and most of the countries reside in different time zones. For e.g. Time difference between India and U.S is 10 hours 30 minutes. So, when there is a huge traffic on Indian datacenters during peak hours then some of the load will be shifted on U.S datacenters which are idle or free during those hours. This logic provides an effective and efficient load balancing technique. Following data proves that during those hours U.S datacenters are idle or have light traffic as compared with Indian datacenters.

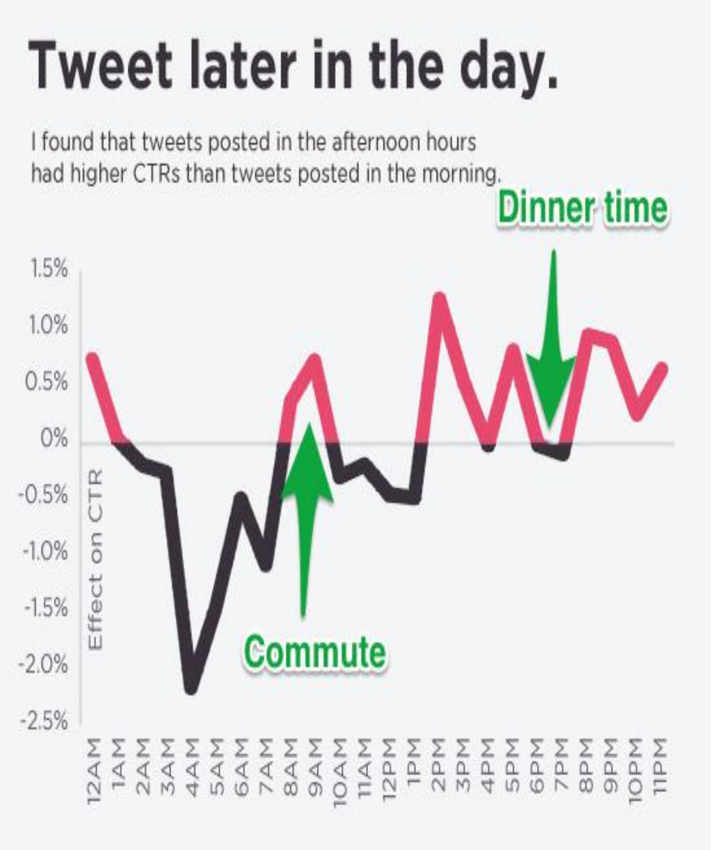


Figure 5: Time Graph of Twitter [16]

As above describes clearly that from 1 AM to 8 AM, 10 AM to 1 PM (i.e. Black line) there is very less traffic and from 8AM to 10 AM, 1 PM to 6 PM (i.e. Red line) very huge traffic on datacenters. So, consider Red line as Indian working hours (or office hours) during high traffic and Black line as U.S hours during same period of time. Timing difference is 10 hours 30 minutes between both the countries. Load will be easily balanced by applying this logic. It will reduce the extra cost for setting up new datacenter because with this logic a load can be shared with the existing datacenters of another region.

3.2.1 Proposed Algorithm

1. If in a region no of users increasing rapidly then load over datacenters also increases.
2. When load reaches at 60% in that region then it will start finding datacenters in another region which is lightly loaded.
3. The datacenters in another region must be at a difference of 5 and half hour from heavily loaded datacenters.
4. When suitable datacenters found whose load is less than heavily loaded datacenters, then some of load from heavily loaded datacenters will be shifted to other region's lightly loaded datacenter.
5. This will balance the load properly and effectively.

4. SIMULATION AND RESULT ANALYSIS

For testing scenarios, Cloud Analyst simulator has been used and results will be produced on the basis of data collected by using this tool. Only the average values of results are presented in this paper because scenario runs several times. So, following results contains average values comparison b/w equally spread current execution algorithm and proposed algorithm. Datacenter Response Time, Overall Response Time by Region and Cost in Region are three parameters used in comparison. A minor variation is observed in values during result comparison b/w existing and proposed algorithm.

4.1 Cloud Analyst

It based on GUI environment that is established on CloudSim framework. CloudSim is a toolkit that permits performing simulation and observations. The chief difficulty with CloudSim is that all the work essential to be complete programmatically. It permits the consumer to do frequent simulations with minor alteration in constraints very simply and rapidly. Cloud analyst permits setting position of operators that are producing the request and also the place of data centers. In this configuration constraints can be customize like amount of operators, amount of demand produced per operator per hour, amount of virtual machines, amount of processors, quantity of storage, network bandwidth and other essential constraints. Based on the constraints the tool calculates the simulation result and displays in graphical method. Result contains response time, processing time, cost. After executing several simulations the cloud vendor can define the finest method to allot assets, on the basis of request which data center will be selected and can enhance cost for giving services [15].

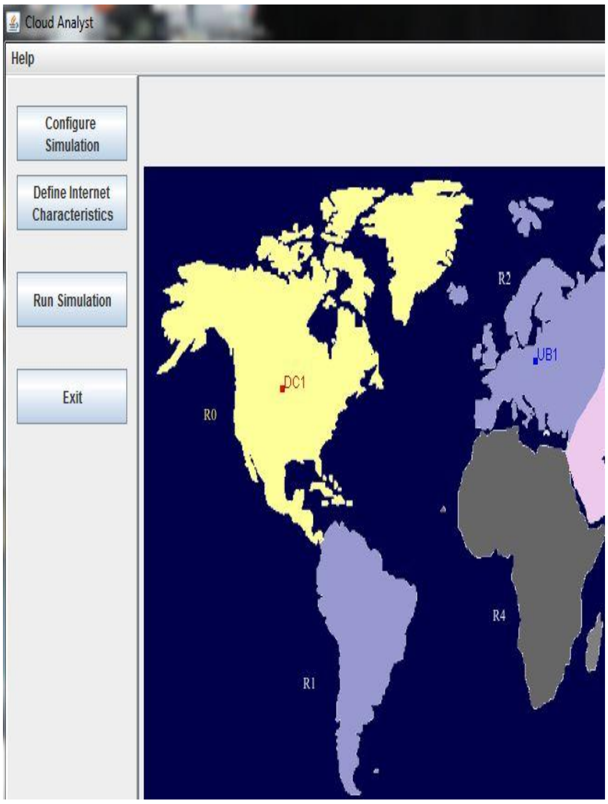


Figure 6: Graphical User Interface of Cloud Analyst

4.2 Scenarios

4.2.1 Scenario A

Table 1: Parameters and values used in Simulation

Regions Used	No. of Datacenters	No of Usebase	Request/user/hour in each UserBase
R3	5	10	200
R4	3	--	--

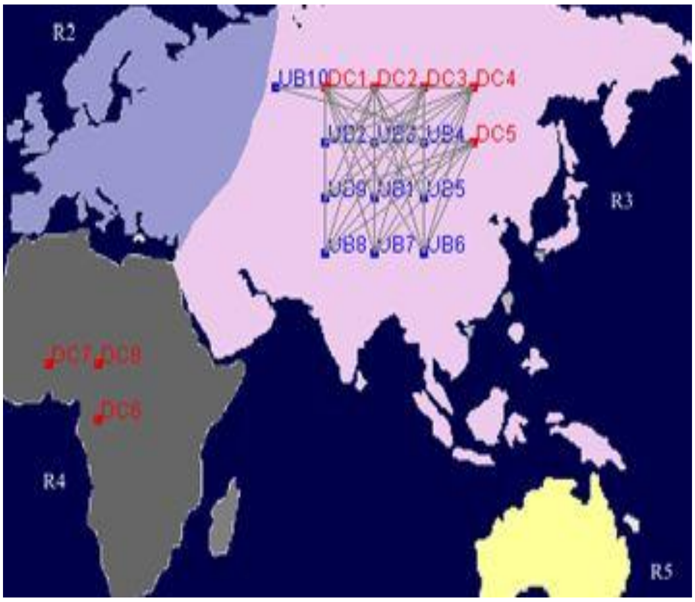


Figure 7: Simulation running by using ESCE Algorithm

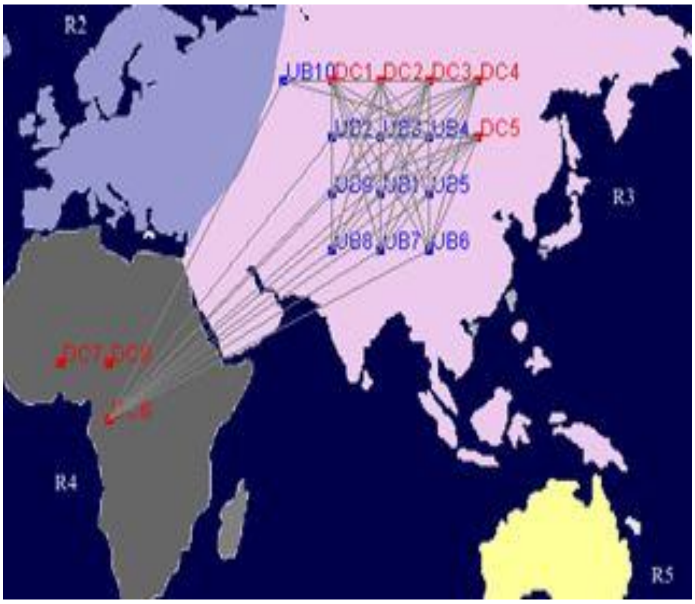


Figure 8: Simulation running by using Proposed Algorithm

Table 2: Comparison of Maximum Datacenter Response Time

Algorithm	Maximum
Equally Spread (ESCE)	1.07
Proposed Algorithm	0.93

Table 3: Comparison of Maximum Overall Response Time in Region

Algorithm	Maximum
Equally Spread (ESCE)	65.65
Proposed Algorithm	64.15

Table 4: Comparison of Cost

Algorithm	VM Cost ($)	Data Transfer Cost	Total
Equally Spread (ESCE)	4.00	2.12	6.12
Proposed Algorithm	4.40	1.54	5.94

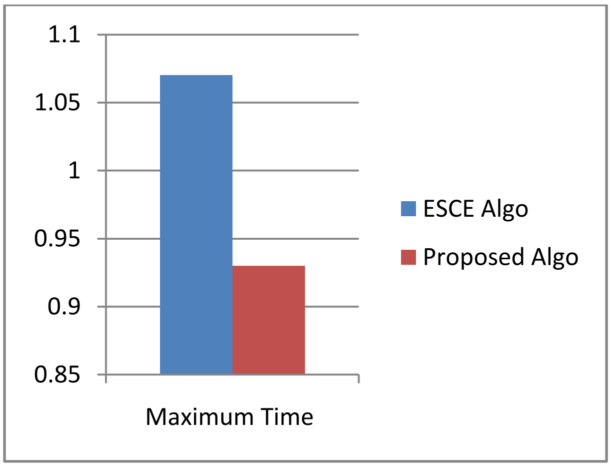


Figure 9: Analytical Comparison of Maximum Datacenter Response Time

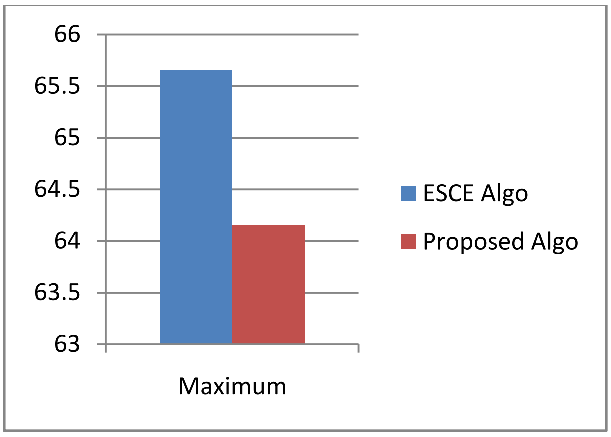


Figure 10: Analytical Comparison of Maximum Overall Response Time

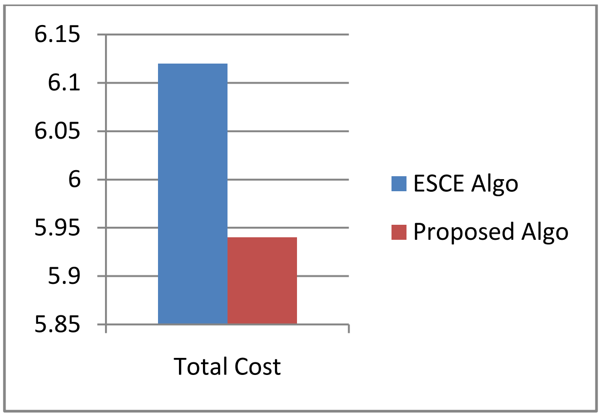


Figure 11: Analytical Comparison of Cost

4.2.2 Scenario B

Table 5: Parameters and values used in Simulation

Regions Used	No. of Datacenters	No of Usebase	Request/user/ hour in each UserBase
R3	5	10	1000
R4	3	--	--

Table 6: Comparison of Maximum Datacenter Response Time

Algorithm	Maximum
Equally Spread (ESCE)	1.07
Proposed Algorithm	0.96

Table 7: Comparison of Maximum Overall Response Time in Region

Algorithm	Maximum
Equally Spread (ESCE)	66.65
Proposed Algorithm	64.42

Table 8: Comparison of Cost

Algorithm	VM Cost ($)	Data Transfer Cost	Total
Equally Spread (ESCE)	4.00	10.52	14.52
Proposed Algorithm	4.40	9.94	14.34

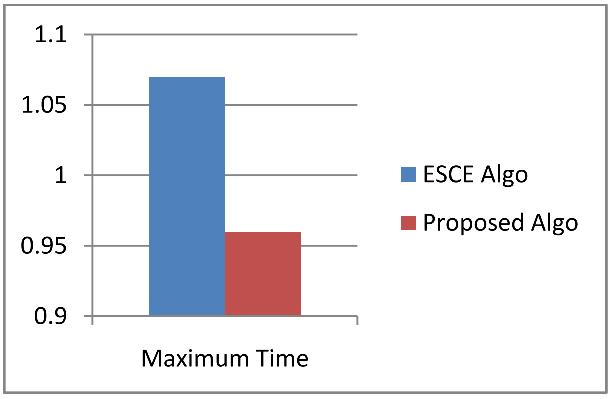


Figure 12: Analytical Comparison of Maximum Datacenter Response Time

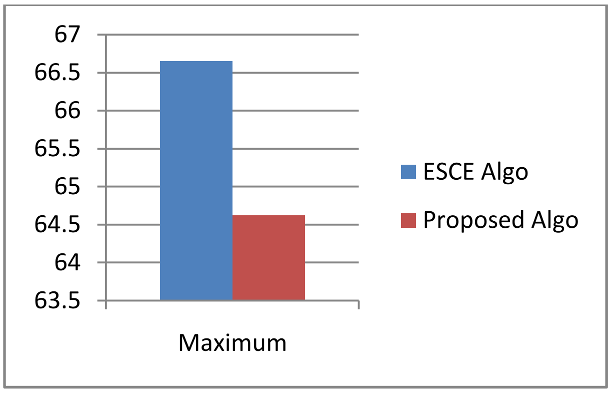


Figure 13: Analytical Comparison of Maximum Overall Response Time

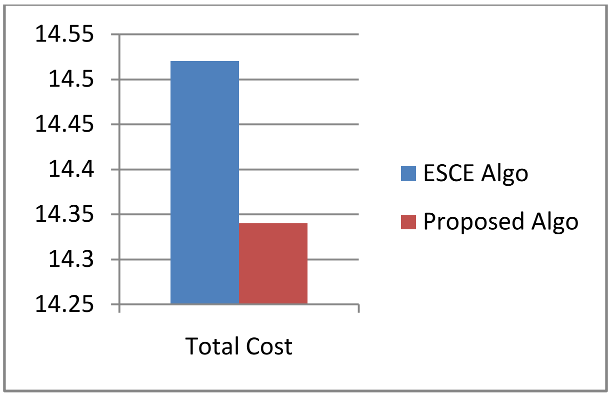


Figure 14: Analytical Comparison of Cost

5 CONCLUSION AND FUTURE WORK

The volume of internet users growing continuously every minute. So, the load on network devices also increases rapidly. There is a need of load balancing algorithms that cannot compromise response time. Providing QoS to users are main focus and this can be achieved when load is balanced properly between data centers. Existing algorithms shared load with those datacenters which resides in the same region. So, during peak hours of traffic load will be shifted only those datacenters. This will increase scheduling overhead, response time, datacenter processing time. There's only one solution left that establish a new datacenter for proper balancing of load during peak hours but there is also a problem with this. Establishing a new datacenter just for balancing load during peak hours is extremely costly. Proposed algorithm solves all the problems like huge memory demand and scheduling overhead during peak hours of traffic. It shares load with data centers in another region. It improves the maximum overall response time and maximizes hardware utilization. It reduces data transfer cost and data center processing time as compared with existing algorithm.

There's no doubt that Google, Amazon provides extremely good link quality for their services but when timing difference is more between different regions datacenters then this will lead to poor link quality issue. So in future, for that type of issue work can be done on improving the link quality between different datacenters that have poor link quality due to very large timing difference.

6 REFERENCES

[1] Buyya, Rajkumar., Broberg, James., Goscinski, Andrzej. "Cloud Computing Principles and Paradigms" (1st ed.). Hoboken, New Jersey, USA: Wiley, 2011.

[2] Ranjan Dinesh, Canino Anthony, Izaguirre A Jesus and Douglas Thain "Converting a High Performance Application to an Elastic Cloud Application" 3rd IEEE International Conference on Cloud Computing Technology and Science.

[3] Mao Hong, Zhang Zhenzhong, Zhao Bin, Xiao Limin and Ruan Li "Towards Deploying Elastic Hadoop in Cloud", International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.

[4] Zhang Fan, Cao Junwei, Hwang Kai, Wu Cheng "Ordinal Optimized Scheduling of Scientific Workflows in Elastic Compute Clouds", IEEE, 2011.

[5] Lin Cui and Lu Shiyong "Scheduling Scientific Workflows For Cloud Computing", IEEE 4th International Conference on Cloud Computing, 2011.

[6] Genaud Stephane and Gossa Julien “Cost-wait Trade-offs in Client-side Resource Provisioning with Elastic Clouds”, IEEE 4th International Conference on Cloud Computing, 2011.

[7] Ishii Atsushi and Suzumura Toyotaro “Elastic Stream Computing with Clouds”, IEEE 4th International Conference on Cloud Computing, 2011.

[8] Tommaso Cucinotta, Konstanteli Kleopatra and Varvarigou Theodora “Probabilistic Admission Control for Elastic Cloud Computing”.

[9] Marshall Paul, Tufo Henry and Kate Keahey “Provisioning Policies for Elastic Computing Environments”, IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, 2012.

[10] Indukuri Raju Krishnam R, P Varma Sureshand Moses Jose G “Performance measure of multi stage scheduling algorithm in cloud computing”, IEEE IntemationalConference on Cloud Computing, Technologies, 2012.

[11] Yu Hongbo, Lan Yihua, Zhang Xingang and Liu Zhidu) “Job Scheduling Algorithm In Cloud Environment”, IEEE International Conference On Computational and Information Sciences, 2013.

[12] Antony Simy, Antony Soumya, A S Beegom Ajeena and M S Rajasree “Task Scheduling Algorithm with Fault Tolerance for Cloud”, IEEE International Conference on Computer Sciences, 2012.

[13] Kaur Amandeep and Kinger Supriya “Analysis of Load Balancing Techniques in Cloud Computing”, International Journal of Computers & Technology, 2013.

[14] Vijayalakshmi M. and Kumar Venkatesa V “Investigations on Job Scheduling Algorithms in Cloud Computing”, International Journal of Advanced Research in Computer Science & Technology, 2014.

[15] Mohapatra Subasish, Rekha K.Smruti, Mohanty Subhadarshini “A Comparison of Four Popular Heuristics for Load Balancing of Virtual Machines, 2013.

[16] Bufferapp, Available: https://blog.bufferapp.com/best-time-to-tweet-post-to-facebook-send-emails-publish-blogposts.