# Robust and Flexible Algorithm of Load Balancing for Cloud Computing

*** Priyanka Mahobia**
**** Sanjay Kumar Brahman**

## ABSTRACT

Cloud Computing became very talked-about within the previous couple of years. As a part of its services, it provides a versatile and straightforward way to keep and retrieve data and files. Load Balancing is crucial for efficient operations in distributed environments. It helps in allocation and de-allocation of instances of applications without failure. Scheduling in cloud computing could be a technique which is employed to boost the general execution time of the task. A decent scheduling algorithm can help in load balancing also.

In this paper we proposed a brand new method for load balancing. We focused on devising an algorithm to schedule jobs and allocate servers in cloud systems. The algorithm is efficient because it provides optimal allocation. It maximizes the quantity of job requests which will be processed in unit time while conserving energy and keeping the prices low. The said optimal allocation is achieved by reducing the idle time of nodes of active servers and reducing the overall number of servers used.

**Keywords:-** Load Balancing, Cloud Computing, Divisible Load Scheduling Theory (DLT), Virtual Machine**.**

* Priyanka Mahobia, Dept. of CSE, RKDF College of Engineering, Bhopal, birasini.co.914@gmail.com
** Sanjay Kumar Brahman, Dept. of CSE, Bhabha Engineering Research Institution, Bhopal, sanjushukla2007@gmail.com

## I. INTRODUCTION

Cloud computing is an on demand service during which shared resources, information, software and other devices are provided consistent with the clients requirement at specific time. It's a term which is usually utilized in case of Internet. the entire Internet are often viewed as a cloud. Capital and operational costs are often cut using cloud computing.

Load balancing in cloud computing systems is basically a challenge now. Always a distributed solution is required. Because it's not always practically feasible or cost efficient to take care of one or more idle services even as to satisfy the specified demands. Jobs can't be assigned to appropriate servers & clients individually for efficient load balancing as cloud may be a very complex structure and components are present throughout a good spread area. Here some uncertainty is attached while jobs are assigned. Figure 1 show the cloud used in network and over all structure of the cloud.

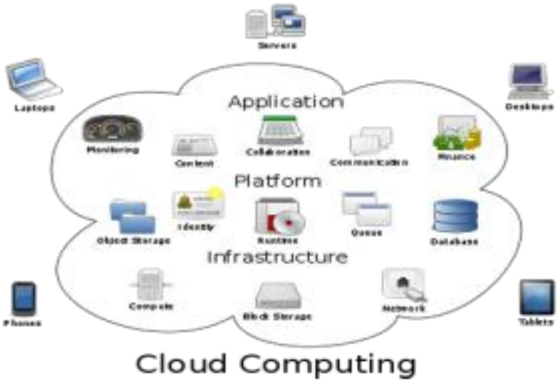

Figure 1. A cloud is used in the diagrams to depict the Internet.

## II. PREVIOUS WORK

In this paper [1], Authors proposes BalanceFlow, which could handle controller load leveling entirely at the controller level. We've a bent to introduce AN extension for OpenFlow switches: CONTROLLER X action. Upon controller load imbalance, the super controller runs partition algorithm and reallocates the load of varied controller by distributing allocation rules to switches. supported the analysis of their BalanceFlow design, they show that BalanceFlow will flexibly regulate the work of each controller, and reach finding a balance between controllers' load and average propagation latencies of the entire network.

In this paper [2], Authors created a hybrid image delivery system victimisation the distributed cloud and heritage servers, and operated as a public internet site throughout August 2010 and August 2011. The user-side server choice mechanism create image server switch quicker, consequently, the distributed cloud and heritage servers area unit well integrated. Distributed datacenters and multiple net gateways (map646 servers) alter the network load leveling and wide-area live migration.

There system worked nearly stably, as results of user-side server choice mechanism and geo-distributed load leveling worked well, and conjointly this system have geo-distributed redundancy. keep with authors they got insight that the wide-area live migration of high loaded VM are going to be fail typically.

A novel load leveling algorithm to affect the load rebalancing drawback in large-scale, dynamic, and distributed file systems has been bestowed during this paper [3]. There proposal strives to balance the masses of nodes and reduce the demanded movement value the utmost amount as potential. Within the absence of representative real workloads (i.e., the distributions of file chunks in an exceedingly large-scale storage system) within the ownership, they have investigated the performance of their proposal and compared it against competitory algorithms through synthesized probabilistic distributions of file chunks. The synthesis workloads check the load leveling algorithms by making variety of storage nodes that area unit heavily loaded. The performance results with theoretical analysis, laptop simulations and a real implementation area unit encouraging, indicating that their planned algorithm performs alright. Their proposal is cherish the centralized algorithm within the Hadoop HDFS production system and dramatically outperforms the competitory distributed algorithm in terms of load imbalance issue, movement value, and algorithmic overhead.

In this paper [4] authors surveyed the progressive of load leveling in cloud system. They establish the state of the art load leveling within the cloud system, giving a definition of this term, its classification and samples of its implementation in classical distributed systems and within the cloud system key technologies likewise as analysis directions and cases study of search.

In this paper [5] authors study the matter of dynamic grouping in cloud computing. To at an equivalent time reach value potency, load leveling, and strength, they propose 2 sorts of grouping strategies: mathematic grouping and heuristic grouping. Thorough experiments are performed to verify the effectiveness of their ways.

Cloud computing could also be a relatively new IT paradigm that provides great quantity of resources at cheap value. The special characteristics of cloud environments and also the dynamic nature of its virtual infrastructure imply economical load leveling solutions that area unit capable of maintaining low values for interval and server masses. Among the important factors that have an impact on the performance of a load balancer is its design which could be decentralized, centralized or hierarchical. During this paper [6] authors administered a comparative study between the three architectures and therefore the way they need an impact on the cloud performance.

A simulated model for a public cloud has been engineered for this purpose at totally different scales and also the system performance was measured underneath the three potential load leveling architectures. The experimental results illustrated the dominant performance of the hierarchical design for load balancers as a results of its ability to separate the load leveling overhead among several load balancers running numerous algorithms which can supplement each other whereas maintaining some nature of centralized management over the cloud.

In this paper [7] authors proposes a totally unique load leveling algorithm along side the matter setting. Within the planned algorithm consumer submits the necessity or characteristics of the duty to cloud supplier. Supplier stores the necessity within the repository in xml format. The last word choice of the resource relies on the resource occupancy matrix, period of the duty and repair charge. The entire algorithm has been developed by Jdk seven.0. It conjointly explains the algorithm with totally different potential statements and assumption, with the flow of the operating method of the planned algorithm through interfaces.

In this paper [8], authors bestowed a totally unique load-balancing algorithm to affect the load rebalancing drawback in large-scale, dynamic, and distributed file systems in clouds. There proposal strives to balance the masses of nodes and reduce the demanded movement value the utmost amount as potential, whereas taking advantage of physical network neighborhood and node nonuniformity. Within the absence of representative real workloads (i.e., the distributions of file chunks in an exceedingly largescale storage system) within the ownership, they have investigated the performance of their proposal and compared it against competitory algorithms through synthesized probabilistic distributions of file chunks. The synthesis workloads check the load-balancing algorithms by making variety of storage nodes that area unit heavily loaded. the pc simulation results area unit encouraging, indicating that there planned algorithm performs alright. There proposal is cherish the centralized algorithm within the Hadoop HDFS production system and dramatically outperforms the competitory distributed algorithm in terms of load imbalance issue, movement value, and algorithmic overhead. Significantly, their load-balancing algorithm exhibits a fast convergence rate. The potency and effectiveness of their style area unit more valid by analytical models and a real implementation with a small-scale cluster setting.

In this paper [9] authors planned a mean athletic game theoretical framework for cloud computing systern. Wherever the player interacts through the interval provided by the system and decides volitionally to chop back their load to act in steady state. The system of coupled SDE are going to be discredited to derive AN algorithm to be enforced in user application such navigator.

In this paper [10], Authors gift design and elegance with dynamic scaling state of affairs to research the performance of Hadoop in high speed retrieval of data in an exceedingly cloud setting. This system is made by Master internet server with multi level compartmentalisation at NameNode and DataNode that contain the records and B+ tree as a header. The results shows that victimisation this design within the Hadoop and creating map section as a web server, quicker scan and write operation for MapReduce programs are going to be achieved. for giant databases of data warehouse applications in cloud, searching method takes a awfully durable because the knowledge is scattered. High sped retrieval of data are going to be significantly helpful for period applications supported Hadoop wherever fast attractive and storage of data is vital .

In this paper [11] Authors introduced a totally unique column generation approach for locating the VLAN assignment drawback in cloud information centers. They gift 2 decomposition approaches: a particular, likewise as a semi-heuristic model to understand higher runtime and measurability. They compare each models against the pure ILP model of the VLAN assignment drawback, and prove that there approach yields a substantial decrease within the dimensions of the explored search house with encouraging optimality gap. They conjointly compared their decomposition approach against state of the art protocols in traffic engineering; there comparative analysis has shown that there model outperforms its peers in most network topologies in terms of link load, gettable gap from edge disk resolution, likewise as in goodput. As they have antecedently mentioned, employing a simpler technique to travel from the relaxed disk resolution to the integral ILP resolution will probably improve their model’s optimality gap.

In this paper [12] Authors planned a fresh increased & economical Hybrid planning algorithm then enforced in cloud computing setting victimisation CloudSim toolkit, in java language. The analysis work concerned developing a cheap Hybrid VM load leveling algorithm for the cloud and conducting a comparative analysis of the planned algorithm with the prevailing algorithms. Intermediate deliverables enclosed checking out the prevailing VM load leveling algorithms, proposing a cheap algorithm for VM load leveling, implementing the algorithm on Cloud Analyst, and examination the planned algorithm with the prevailing algorithms on known parameters. By visualizing the cited parameters in graphs and tables they're going to simply establish that the interval and knowledge centre interval is improved likewise as value is reduced compared to the prevailing planning parameters.

For mobile cloud service, in several cases, clouds need to search service candidates for tasks, which suggests it's reserve to understand the computation of tasks. During this paper [13], authors propose a demand-driven task planning model, and declare how to estimate warrant complete time. AN algorithm, named 2DCGA is bestowed and tests show it's smart performance.

In this paper [14], authors bestowed MB, a VM migration policy appropriate for a distributed management in an exceedingly cloud datacenter. To do so, they relyed on a decentralized resolution for cloud virtual infrastructure management (DAM), during which the hosts of the datacenter area unit ready to self-organize and reach a worldwide VM reallocation found out, keep with a given policy.
They tested MB behavior by means of DAM-sim machine. The policy shows smart performance for varied information centers dimensions in terms of every range of migrations requested and most range of messages changed by one host. Therefore, they'll assert that the decentralized nature of their approach can intrinsically contribute to increase the measurability of the cloud management infrastructure. The machine will solely represent a snap-shot of the datacenter work, whereas they plan to extend DAM-Sim so on use the distributed policy to a dynamically dynamical state of affairs and to make a benchmark comparison of their approach that concentrate on time performance. At latter stage, they'll increase the frequency of variations in VM load requests to higher mirror real datacenter environments. Finally, they plan to check their implementation on a real cloud infrastructure and compare the time to induce a typical distributed call with the centralized implementation of an identical reallocation policy.

In this paper [15], authors propose a fresh offline load-balancing algorithm Prepartition to mirror the feature of capability sharing and glued interval constraint in Cloud information centers. On paper they prove that Prepartition could also be a (1+_)-approximation wherever _= one k and k could also be a positive integer . By increasing k it's potential to be terribly near best resolution, i.e., by setting k price, it's conjointly potential to understand predefined load balance goal as desired. There area unit still variety of study problems like creating appropriate decisions between total partition numbers and cargo balance objective, analyzing the performance in an exceedingly real information center, and considering precedence constraints among totally different VM requests.

In this paper [16], author gift a totally unique approach for rising resilience, the facility to hide failures, in cloud services employing a mixture of dark and load-balancing algorithms. The adoption of the dark paradigm permits the service to autonomously reduce computing capability necessities by degrading user expertise so on make sure that response Times Square measure finite. Thus, it provides a natural candidate for resilience improvement once failures end in capability shortages. However, progressive load-balancers area unit usually not designed for self-adaptive cloud services. The self-adaptivity embedded within the dark service interferes with the actions of load balancers that route requests supported measurements of the response times of the replicas. Throughout this investigation, they highlighted the excellence between load-balancers that act whenever a fresh request is received and algorithms that sporadically update the routing weights, checking out that the formers area unit far more effective than the latter ones. However, the dark paradigm sporadically updates the rheostat values to match specific necessities.

During this paper [17], author elaborates the construct of designing & dynamic provisioning participating in distinguished role during a n exceedingly signing the tasks in a cloud computing setting for evenhanded load distribution with aim to understand economical utilization of resources, improved interval of jobs and removing matters of node overloading and under-loading within the system. They mentioned the assorted load planning algorithms enforced in various heterogeneous networks a bit like the cloud, grid, etc. These algorithms area unit analyzed on numerous planning parameters and techniques. For e.g. higher utilization rate is achieved by victimisation min-min, segmental min-min, double min-min, & max-min algorithms; A* completes a task at earliest time, and weighted spherical robin reduces computation value. The analysis is performed for bigger resource utilization, reduced value & debt to understand most turnout and better performance.

In this paper [18], authors designed a cheap algorithm that manages the load at the server by considering this standing of the all accessible VMs for assignment the incoming requests intelligently. The VM-assign load balancer principally focuses on the economical utilization of the resources NMs. They tested that their planned algorithm optimally distributes the load and thus underneath / over utilization (VMs) things won't arise. When put next to existing Active-VM load balance algorithm, the load wasn't properly distributed on the VMs. keep with the authors the result proves that initial VMs area unit over utilised and later VMs area unit underutilized. There planned algorithm solves the matter of inefficient utilization of the VMs / resources compared to existing algorithm.

In this paper [19], author present a completely unique approach for improving resilience, the power to cover failures, in cloud services employing a combination of brownout and load-balancing algorithms. The adoption of the brownout paradigm allows the service to autonomously reduce computing capacity requirements by degrading user experience so as to ensure that response times are bounded. Thus, it provides a natural candidate for resilience improvement when failures cause capacity shortages. However, state-of-the-art load-balancers are generally not designed for self-adaptive cloud services. The self-adaptivity embedded within the brownout service interferes with the actions of load balancers that route requests supported measurements of the response times of the replicas. During this investigation, they highlighted the difference between load-balancers that act whenever a replacement request is received and algorithms that periodically update the routing weights, checking out that the formers are much more effective than the latter ones. However, the brownout paradigm periodically updates the dimmer values to match specific requirements.

In this paper [20], author elaborates the concept of scheduling & dynamic provisioning playing prominent role during a ssigning the tasks in a cloud computing environment for equitable load distribution with aim to realize efficient utilization of resources, improved reaction time of jobs and removing things of node overloading and under-loading within the system. They discussed the varied load scheduling algorithms implemented in various heterogeneous networks just like the cloud, grid, etc. These algorithms are analyzed on various scheduling parameters and methods. For e.g. higher utilization rate is achieved by using min-min, segmented min-min, double min-min, & max-min algorithms; A* completes a task at earliest time, and weighted round robin reduces computation cost. The analysis is performed for greater resource utilization, reduced cost & debt to realize maximum throughput and better performance.

In this paper [21], authors designed an efficient algorithm which manages the load at the server by considering the present status of the all available VMs for assigning the incoming requests intelligently. The VM-assign load balancer mainly focuses on the efficient utilization of the resources NMs. They proved that their proposed algorithm optimally distributes the load and hence under / over utilization (VMs) situations won't arise. In comparison to existing Active-VM load balance algorithm, the load wasn't properly distributed on the VMs. consistent with the authors the result proves that initial VMs are over utilized and later VMs are underutilized. There proposed algorithm solves the matter of inefficient utilization of the VMs / resources compared to existing algorithm.

In this Paper [22] authors proposed a virtual machine performance maximization load balancing algorithm that also balances load of physical machines. As per authors they firstly studies relationship between performance of VM (Virtual machine) and workload of PM (Physical machine) from perspective of both cloud providers and user’s. Authors used this relationship to predict the VM’s performance. Then they used greedy based algorithm to solve the problem. They evaluated their proposed algorithm on CloudSim platform and real OpenStack platform. As per the result evaluation done by authors, the proposed algorithm shows promising performance improvement.

## III. PROPOSED ARCHITECTURE

This algorithm focuses mainly on finding out if the incoming requests can be further divided into subparts & if the incoming requests can be further divided in subparts then assigning each subpart to the first available VM in the list or else assigning the first available VM to the full request. The functional flow of the algorithm is given in the figure 2.

### 3.1 Algorithm

The algorithm can be divided in two sub parts Registration & Login.

**Input**: No of incoming requests $Re_1$, $Re_2$ . . . . . . .. $Re_n$

Available VM $VM_1$, $VM_2$⋯ . . . . . . $VM_n$

**Output**: All incoming requests $Re_1$, $Re_2$ . . . . . . .. $Re_n$ (If possible subparts) are allocated to first available virtual machine among the available $VM_1$, $VM_2$⋯ . . . . . . $VM_n$

I. At the starting all the VM's will have 0 allocations.
II. VM-assign load balancer keeps the table/ index of VMs which has number of requests currently allocated to every VM.
III. When requests come at the data center it forwards it to the load balancer.
IV. Then Divisible Load Scheduling theory (DLT) is applied on incoming requests. If possible requests are further divided in to subparts & each individual subpart is considered as a request.
V. Parsed the Index table and first available VM is selected for execution.
VI. VM-assign load balancer returns the VM id to the data center.

VII. Request is assigned to the VM. Data center notifies the VM-assign load balancer about the allocation.

VIII. VM-assign load balancer updates the requests hold by each VM.

IX. When the VM finishes the processing the request, data center receives the response.

X. Data center notifies the VM-assign load balancer for the VM de-allocation and VM-assign load balancer updates the table.

XI. Repeat from step II for the next request.

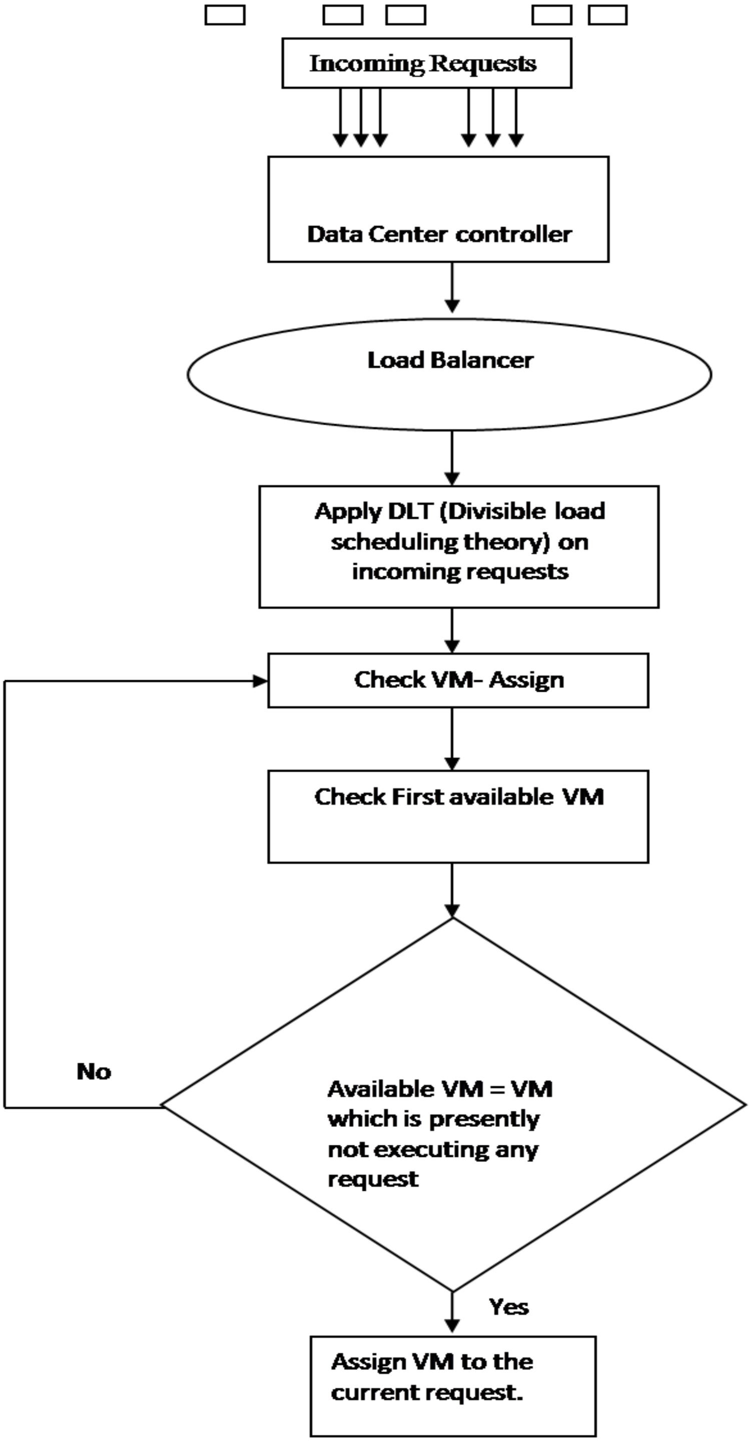
Incoming Requests
Data Center controller
Load Balancer
Apply DLT (Divisible load scheduling theory) on incoming requests
Check VM- Assign
Check First available VM
Available VM = VM which is presently not executing any request
No
Yes
Assign VM to the current request.

Figure 2. Proposed Load Balancing for Virtual Machine in Cloud computing paradigm.

## IV. SIMULATION AND RESULT

**Time Complexity-** For experiment we had taken three VMs & number of requests as mentioned in Table 1 each having three subparts of 10 unit of time. Figure 3 show time complexity comparison of proposed algorithm.

Table 1: Time Comparison

| No OF Requests | Total Execution Time | VM-Assign Load Balancer (Existing) | Proposed Approach |
|---|---|---|---|
| 1 (Total 3) | 30 | 30 | 10 |
| 2 (Total 6) | 60 | 30 | 20 |
| 3 (Total 9) | 90 | 30 | 30 |
| 4 (Total 12) | 120 | 60 | 40 |
| 5 (Total 15) | 150 | 60 | 50 |

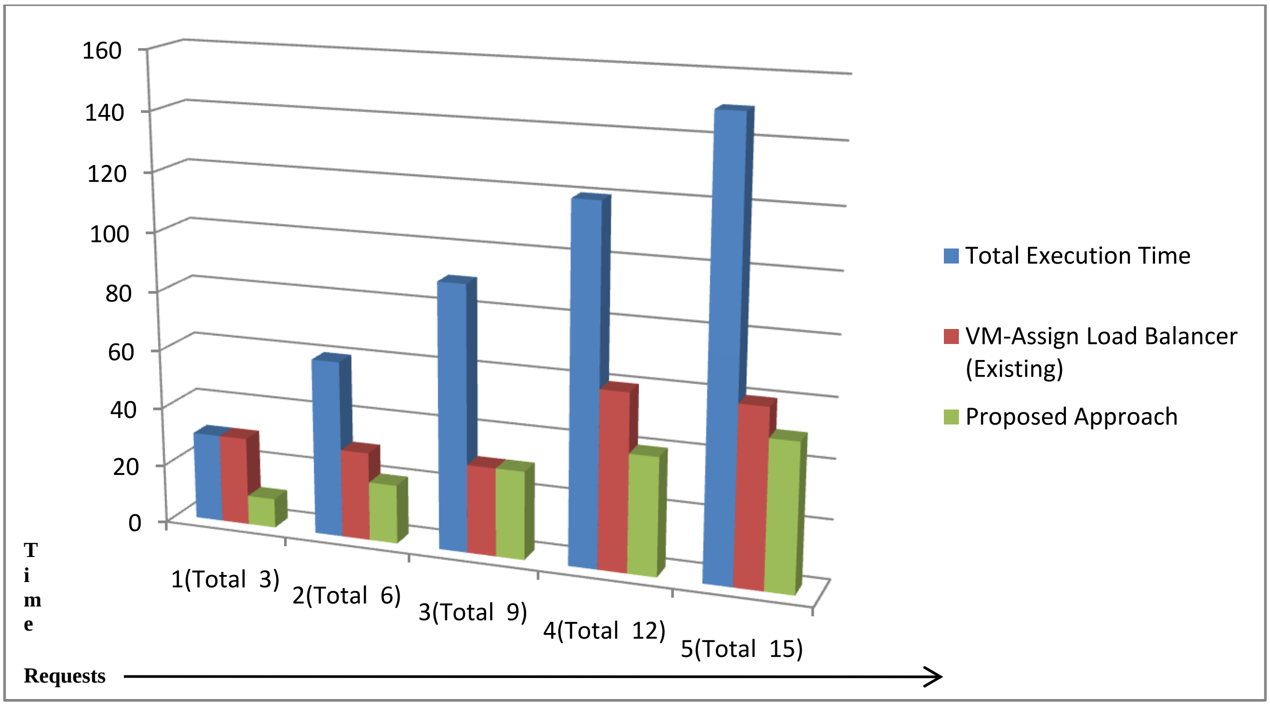

Figure 3. Graph to show time complexity comparison.

## V. CONCLUSION

The recent algorithms are intended which manages the load at the server by considering the present status of the all available VMs for assigning the incoming requests intelligently. The VM-assign load balancer mainly focuses on the efficient utilization of the resources/VMs. The algorithm distributes the load in such a fashion that under / over utilization (VMs) situations won't arise. In comparison to previous Active-VM load balance algorithm, the load wasn't properly distributed on the VMs. The recent algorithms are meant which manages the load at the server by considering the present status of the all available VMs for assigning the incoming requests intelligently.

We proposed a replacement algorithm during which we'll apply DLT (Divisible Load Scheduling Theory) on the recent algorithm to reinforce the use of resources/VMs. The proposed approach reduces the idle time of resources which can ultimately end in increased performance, better throughput & lesser turnaround.

## REFERENCES

1. Yannan Hu, Wendong Wang, Xiangyang Gong, Xirong Que & Shiduan Cheng presented paper entitled "Balanceflow: Controller Load Balancing for Openflow Networks" at Proceedings of IEEE CCIS 2012.
2. Yoshihiro Okamoto, Satoru Noguchi, Satoshi Matsuura, Atsuo Inomata & Kazutoshi Fujikawa presented paper entitled "Koshien-Cloud: Operations of Distributed Cloud as A Large Scale Web Contents Distribution Platform" at 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet.
3. Hsueh-Yi Chun, Che-Wei Chan, Hung-Chang Hsiao & Yu-Chang Chao presented paper entitled "The Load Rebalancing Problem in Distributed File Systems" at 2012 IEEE International Conference on Cluster Computing.
4. A. Khiyaita, M. Zbakh, H. El Bakkali & Dafir El Kettani presented paper entitled "Load Balancing Cloud Computing: State of Art" at 978-1-4673-1053-6/12/$31.00 ©2012 IEEE.
5. Qin Liu, Yuhong Guo, Jie Wu and Guojun Wang presented paper entitled "Dynamic Grouping Strategy in Cloud Computing" at 2012 IEEE Second International Conference on Cloud and Green Computing.
6. Ektemal Al-Rayis & Heba Kurdi presented paper entitled "Performance Analysis of Load Balancing Architectures in Cloud Computing" at IEEE 2013 European Modelling Symposium.
7. Soumya Ray & Ajanta De Sarkar presented paper entitled "Resource Allocation Scheme in Cloud Infrastructure" at 2013 IEEE International Conference on Cloud & Ubiquitous Computing & Emerging Technologies.
8. Hung-Chang Hsiao, Hsueh-Yi Chung, Haiying Shen & Yu-Chang Chao presented paper entitled "Load Rebalancing for Distributed File Systems in Clouds" at IEEE Transactions on Parallel and Distributed Systems, VOL. 24, NO. 5, MAY 2013.
9. A. Khiyafta & M. Zbakh presented paper entitled "Mean field game among cloud computing end users" at 978-1-4799-0324-5/13/$31 .00 ©2013 IEEE.
10. Gita Shah, Annappa & K. C. Shet presented paper entitled "Efficient Way of Searching Data in MapReduce Paradigm" at 978-93-80544-12-0/14/$31.00@ 2014 IEEE.
11. Chadi Assi, Sara Ayoubi, Samir Sebbah & Khaled Shaban presented paper entitled "Towards Scalable Traffic Management in Cloud Data Centers" at IEEE Transactions on Communications, Vol. 62, No. 3, March 2014.

12. Vishwas Bagwaiya & Sandeep k. Raghuwanshi presented paper entitled “Hybrid Approach Using Throttled and Esce Load Balancing Algorithms in Cloud Computing” at IEEE Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on 2014.
13. Zhiming Cai & Chongcheng Chen presented paper entitled “Demand-driven Task Scheduling Using 2D Chromosome Genetic Algorithm in Mobile Cloud” at 978-1-4799-2030-3 /14/$31.00 ©2014 IEEE.
14. Daniela Loreti, Anna Ciampolini presented paper entitled “A Distributed Self-Balancing Policy for Virtual Machine Management in Cloud Datacenters” at 978-1-4799-5313-4/14/$31.00 ©2014 IEEE.
15. Wenhong Tian*, Minxian Xu, Yu Chen & Yong Zhao presented paper entitled “Prepartition: A New Paradigm for the Load Balance of Virtual Machine Reservations in Data Centers” at IEEE ICC 2014 - Selected Areas in Communications Symposium.
16. Cristian Klein, Alessandro Vittorio Papadopoulos, Manfred Dellkrantz, Jonas D¨urango, Martina Maggio, Karl-Erik A˚ rze´n, Francisco Herna´ndez-Rodriguez & Erik Elmroth presented paper entitled “Improving Cloud Service Resilience using Brownout-Aware Load-Balancing” at 2014 IEEE 33rd International Symposium on Reliable Distributed Systems.
17. Divya Chaudhary, Bijendra Kumar presented paper entitled “An Analysis of the Load Scheduling Algorithms in the Cloud Computing Environment: A Survey” at IEEE 2014 9th International Conference on Industrial and Information Systems (ICIIS).
18. Shridhar G.Damanal and G. Ram Mahana Reddy presented paper entitled “Optimal Load Balancing in Cloud Computing by Efficient Utilization of Virtual Machines” at IEEE 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS).
19. T. Deepa and Dhanaraj Cheelu presented paper entitled “A comparative study of static and dynamic load balancing algorithms in cloud computing” at 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS).
20. Simranjit Kaur and Tejinder Sharma presented paper entitled “Efficient load balancing using improved central load balancing technique” at 2018 2nd International Conference on Inventive Systems and Control (ICISC).
21. J. Mercy Faustina, B. Pavithra, S. Suchitra and P. Subbulakshmi presented paper entitled “Load Balancing in Cloud Environment using Self-Governing Agent” at Proceedings of 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA).
22. Hui Zhao, Quan Wang, Member, IEEE, Jing Wang, Bo Wan, Shangshu Li presented paper entitled “VM Performance Maximization and PM Load Balancing Virtual Machine Placement in Cloud” at Proceedings of 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID).