# Innovative Load Balancing Algorithm by applying DLT

Seema Kohare* Mohit Jain**
M.Tech Scholar* Head of Department Computer Science**
Department of Computer Science & Engineering*,**
BM Group of College of Engineering and Technology, Indore, Madhya Pradesh, India
s.kohare15@gmail.com*, hod.computers@bmcollege.ac.in**

***Abstract:*** **Cloud Computing became very popular in the last few years. As part of its services, it provides a flexible and easy way to keep and retrieve data and files. Load Balancing is essential for efficient operations in distributed environments. It helps in allocation and de-allocation of instances of applications without failure. Scheduling in cloud computing is a technique which is used to improve the overall execution time of the job. A good scheduling algorithm can help in load balancing as well.**

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

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

## 1. Introduction

Cloud computing is an on demand service in which shared resources, information, software and other devices are provided according to the clients requirement at specific time. It's a term which is generally used in case of Internet. The whole Internet can be viewed as a cloud. Capital and operational costs can be cut using cloud computing.

Load balancing in cloud computing systems is really a challenge now. Always a distributed solution is required. Because it is not always practically feasible or cost efficient to maintain one or more idle services just as to fulfill the required demands. Jobs can't be assigned to appropriate servers & clients individually for efficient load balancing as cloud is a very complex structure and components are present throughout a wide spread area. Here some uncertainty is attached while jobs are assigned.

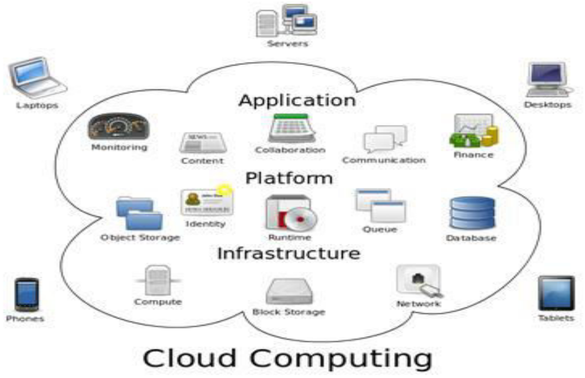

Figure 1: A cloud is used in the diagrams to depict the Internet

## 2. Previous Work

In this paper [1], Author has sought to engender robustness through the promotion of autonomic function by optimizing component level network structures and ensuring rapid propagation throughout the system of resource allocation techniques. The results gained show that increased natural connectivity, measured as the mean Eigen value of the Laplacian matrix of an e-service network, across the services gives a measurably increased improvement in throughput from autonomic operation and vice-versa.

Authors considered Load balancing, relevant to large scale services provision or cloud computing scenarios: To achieve load balancing the network is rewired to group like services together; the results showed increased resilience to node and edge failure as measured by the load balancing performance and robustness. It is envisaged that the decentralized methods of promoting robustness reported on in this paper can be adapted for application to these and many other scenarios involving large computer systems.

In this paper [2], authors presented NBVirMan, a monitoring system framework for virtualized NaradaBrokering systems that automates the task of monitoring and determining VM allocation/deallocation based on pre-defined threshold values. Incorporating NaradaBrokering in the cloud platforms gives a lot of benefits towards Green IT by reducing power consumptions.

In this paper [3], authors presented the *ElasticStream* system that dynamically allocates computational resources on the cloud in an elastic manner for a data stream processing application. To minimize the charges for using the Cloud environment while satisfying the SLA, they formulated a linear programming problem to optimize the costs as a trade-off between the application's latency and charges. Authors also implemented a system to assign or remove computational resources dynamically on top of the data stream computing middleware, System S. Through experiments using Amazon EC2, a commercial Cloud environment, we confirmed that our proposed approach could save 80% of the costs while maintaining the application's latency in comparison to a naïve approach.

To solve the problem of load imbalance in cloud environment, in this paper [4], authors proposes resource allocation method named SLB. SLB consists of two parts: (1) Online VM's performance data statistical analysis and resource demand forecast; (2) An algorithm for the purpose of load balancing, which chooses proper host in resource pool based on the resource demand forecast of VM and the historical load information of hosts. Experimental results show that SLB can make load balance in time, and also make more balanced use of different resources.

In this paper [5], Authors proposes BalanceFlow, which can handle controller load balancing entirely at the controller level. We introduce an extension for OpenFlow switches: CONTROLLER *X* action. Upon controller load imbalance, the super controller runs partition algorithm and reallocates the load of different controller by distributing allocation rules to switches. Based on the evaluation of their BalanceFlow architecture, they show that BalanceFlow can flexibly adjust the workload of each controller, and succeed in finding a balance between controllers' load and average propagation latencies of the whole network.

In this paper [6], Authors constructed a hybrid image delivery system using the distributed cloud and legacy servers, and operated as a public website during August 2010 and August 2011. The user-side server selection mechanism make image server switching faster, consequently, the distributed cloud and legacy servers are well integrated. Distributed datacenters and multiple Internet gateways (map646 servers) enable the network load balancing and wide-area live migration.

There system worked almost stably, because user-side server selection mechanism and geo-distributed load balancing worked well, and also this system have geo-distributed redundancy. According to authors they got insight that the wide-area live migration of high loaded VM can be fail sometimes.

A novel load balancing algorithm to deal with the load rebalancing problem in large-scale, dynamic, and distributed file systems has been presented in this paper [7]. There proposal strives to balance the loads of nodes and reduce the demanded movement cost as much as possible. In the absence of representative real workloads (i.e., the distributions of file chunks in a large-scale storage system) in the public domain, they have investigated the performance of their proposal and compared it against competing algorithms through synthesized probabilistic distributions of file chunks. The synthesis workloads stress test the load balancing algorithms by creating a few storage nodes that are heavily loaded. The performance results with theoretical analysis, computer simulations and a real implementation are encouraging, indicating that their proposed algorithm performs very well. Their proposal is comparable to the centralized algorithm in the Hadoop HDFS production system and dramatically outperforms the competing distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead.

In this paper [8] authors surveyed the state-of-the-art of load balancing in cloud computing system. They establish the state of the art load balancing in the cloud computing system, giving a definition of this term, its classification and examples of its implementation in classical distributed systems and in the cloud computing system key technologies as well as research directions and cases study of search.

In this paper [9] authors study the problem of dynamic grouping in cloud computing. To simultaneously achieve cost efficiency, load balancing, and robustness, they propose two kinds of grouping strategies: mathematic grouping and heuristic grouping. Extensive experiments have been performed to verify the effectiveness of their strategies.

Cloud computing is a relatively new IT paradigm that offers huge amount of resources at reasonable cost. The special characteristics of cloud environments and the dynamic nature of its virtual infrastructure call for efficient load balancing solutions that are capable of maintaining low values for response time and server loads. Among the critical factors that affect the performance of a load balancer is its architecture which can be decentralized, centralized or hierarchical. In this paper [10] authors carried out a comparative study between the three architectures and how they affect the cloud performance.

A simulated model for a public cloud has been built for this purpose at different scales and the system performance was measured under the three possible load balancing architectures. The experimental results illustrated the dominant performance of the hierarchical architecture for load balancers due to its ability to split the load balancing overhead among many load balancers running various algorithms that can supplement each other while maintaining some nature of centralized management over the cloud.

In this paper [11] authors proposes a novel load balancing algorithm along with the problem environment. In the proposed algorithm client submits the requirement or characteristics of the job to cloud provider. Provider stores the requirement in the repository in xml format. The final selection of the resource is based on the resource occupancy matrix, duration of the job and service charge. The entire algorithm has been developed by Jdk 7.0. It also explains the algorithm with different possible statements and assumption, with the flow of the working process of the proposed algorithm through interfaces.

In this paper [12], authors presented a novel load-balancing algorithm to deal with the load rebalancing problem in large-scale, dynamic, and distributed file systems in clouds. There proposal strives to balance the loads of nodes and reduce the demanded movement cost as much as possible, while taking advantage of physical network locality and node heterogeneity. In the absence of representative real workloads (i.e., the distributions of file chunks in a largescale storage system) in the public domain, they have investigated the performance of their proposal and compared it against competing algorithms through synthesized probabilistic distributions of file chunks. The synthesis workloads stress test the load-balancing algorithms by creating a few storage nodes that are heavily loaded. The computer simulation results are encouraging, indicating that there proposed algorithm performs very well. There proposal is comparable to the centralized algorithm in the Hadoop HDFS production system and dramatically outperforms the competing distributed algorithm in terms of load imbalance factor, movement cost, and algorithmic overhead. Particularly, their load-balancing algorithm exhibits a fast convergence rate. The efficiency and effectiveness of their design are further validated by analytical models and a real implementation with a small-scale cluster environment.

In this paper [13] authors proposed a mean field game theoretic framework for cloud computing systern. Where the player interacts through the response time provided by the system and decides willingly to reduce their load to act in steady state. The system of coupled SDE can be discredited to derive an algorithm to be implemented in end user application such navigator.

In this paper [14], Authors present architecture and design with dynamic scaling scenario to investigate the performance of Hadoop in high speed retrieval of data in a cloud environment. This system is constructed by Master web server with multi level indexing at NameNode and DataNode which contain the records and B+ tree as a header. The results shows that using this architecture in the Hadoop and making map phase as a web server, faster read and write operation for MapReduce programs can be achieved. For large databases of data warehouse applications in cloud, searching process takes a very long time as the data is scattered. High sped retrieval of data can be particularly useful for real-time applications based on Hadoop where quick fetching and storage of data is necessary.

In this paper [15], author elaborates the concept of scheduling & dynamic provisioning playing prominent role in assigning the tasks in a cloud computing environment for equitable load distribution with aim to achieve efficient utilization of resources, improved response time of jobs and removing the situation of node overloading and under-loading in the system. They discussed the various load scheduling algorithms implemented in various heterogeneous networks like the cloud, grid, etc. These algorithms are analyzed on various scheduling parameters and strategies. 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 achieve maximum throughput and higher performance.

In this paper [16], authors designed an efficient algorithm which manages the load at the server by considering the current 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 will not arise. When compared to existing Active-VM load balance algorithm, the load was not properly distributed on the VMs. According to the authors the result proves that initial VMs are over utilized and later VMs are underutilized. There proposed algorithm solves the problem of inefficient utilization of the VMs / resources compared to existing algorithm.

## 3. Proposed Architecture

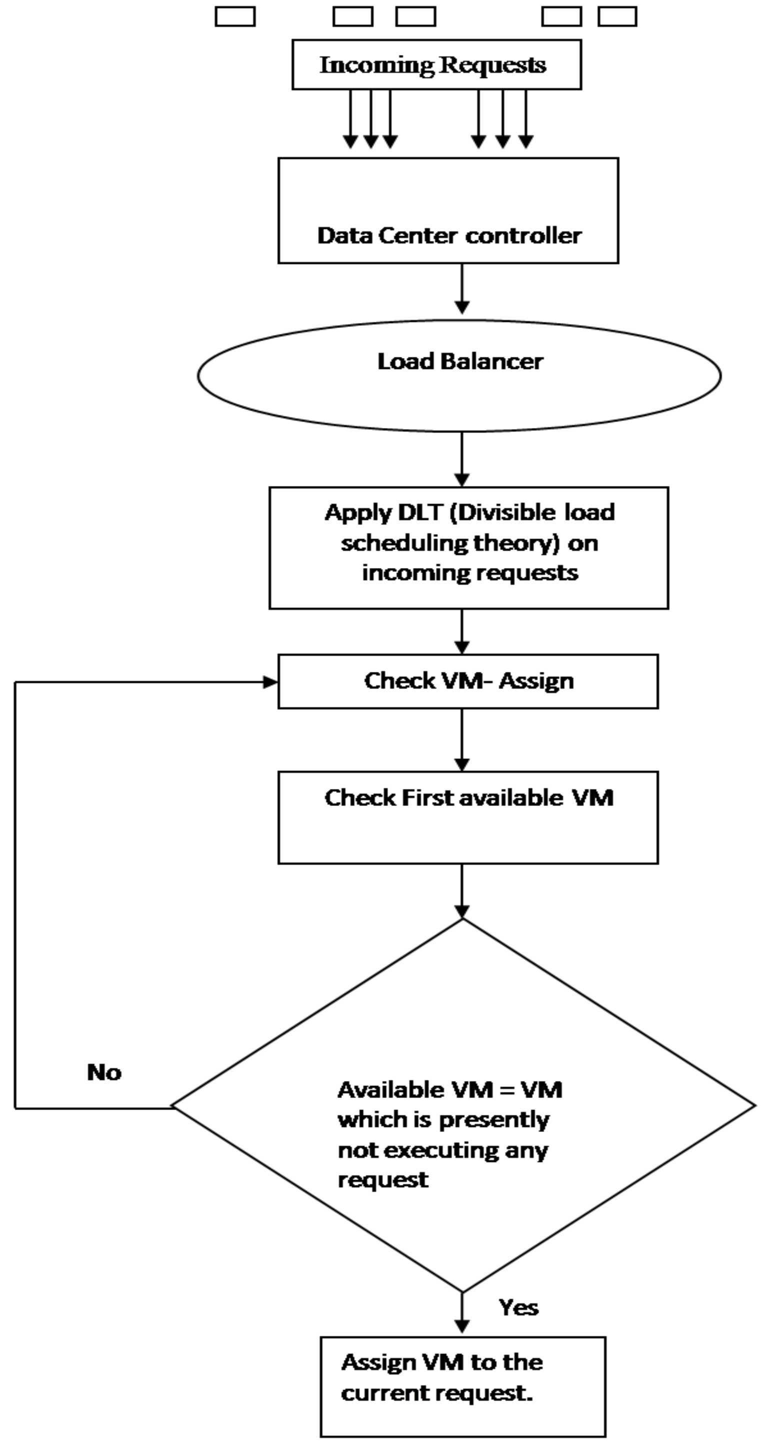
Incoming Requests
Data Center controller
Load Balancer
Apply DLT (Divisible load scheduling theory) on incoming requests
Check VM- Assign
Check First available VM
Available VM = VM which is presently not executing any request
No
Yes
Assign VM to the current request.

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

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

**Input**: No of incoming requests R1, R2 . . . . . . .. Rn

Available VM VM1, VM2⋯ . . . . . . VMn

**Output**: All incoming requests R1, R2 . . . . . . .. Rn (If possible subparts)are allocated to first available virtual machine among the available VM1, VM2⋯ . . . . . . VMn

I. Initially all the VM's have 0 allocations.
II. VM-assign load balancer maintains the index / assign table of VMs which has no. of requests currently allocated to each VM.
III. When requests arrive at the data center it passes to the load balancer.
IV. Then DLT (Divisible Load Scheduling theory is applied) on incoming requests. If possible requests are divided in subparts & each individual subpart is considered as a request.
V. Index table is parsed 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.

## 4. 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.

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 |

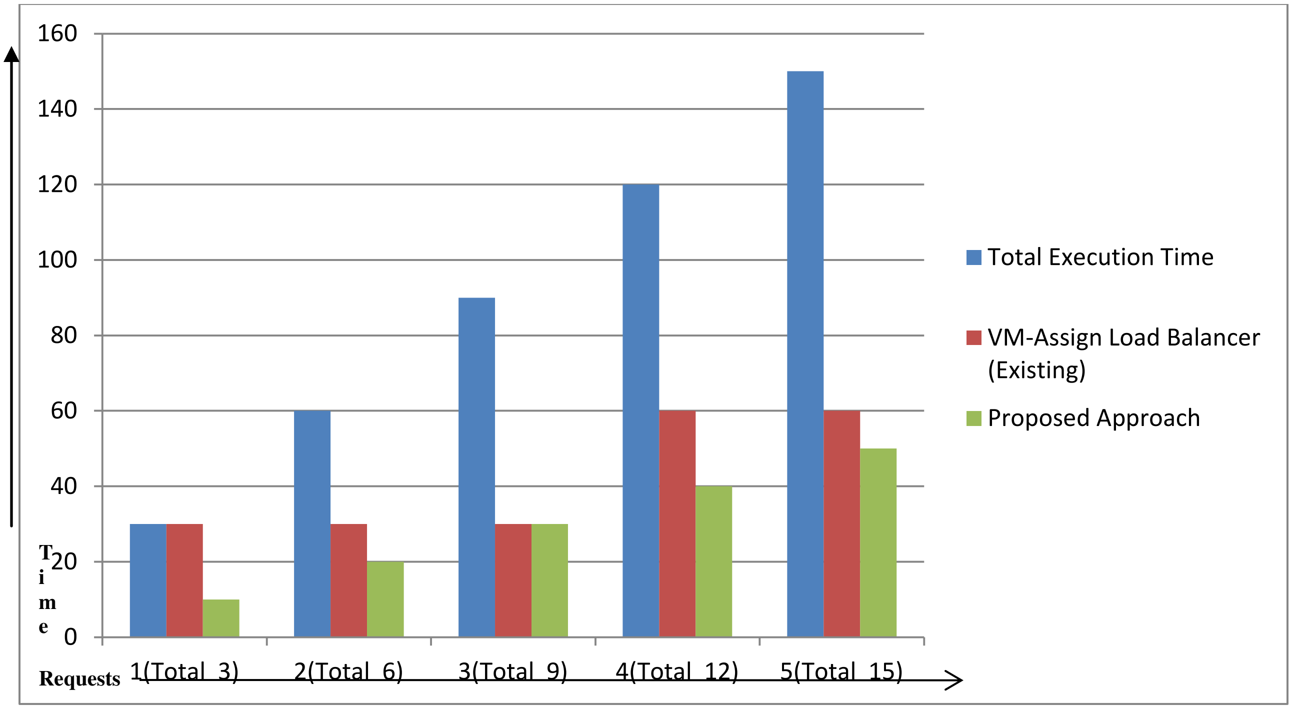

Figure 3: Graph to show time complexity comparison

## 5. Conclusion

The recent algorithms is designed which manages the load at the server by considering the current 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 manner that under / over utilization (VMs) situations will not arise. When compared to previous Active-VM load balance algorithm, the load was not properly distributed on the VMs. The recent algorithms is designed which manages the load at the server by considering the current status of the all available VMs for assigning the incoming requests intelligently.

We proposed a new algorithm in which we will apply DLT (Divisible Load Scheduling Theory) on the recent algorithm to enhance the utilization of resources/VMs. The proposed approach reduces the idle time of resources which will ultimately result in increased performance, better throughput & lesser turnaround time.

## References

[1] Martin Randles, A. Taleb-Bendiab & D. Lamb presented paper entitled “Robustness in Autonomic E-Service Systems” IEEE 2010 Developments in E-systems Engineering.

[2] Frank Yong-Kyung OH, Shin-gyu KIM, Hyeonsang EOM, Heon Y. YEOM, Jongwon PARK & Yongwoo LEE presented paper entitled “A Scalable and Adaptive Cloud-based Message Brokering Service” at Feb. 13~16, ICACT2011.

[3] Atsushi ISHII & Toyotaro SUZUMURA presented paper entitled “Elastic Stream Computing with Clouds” at 2011 IEEE 4th International Conference on Cloud Computing.

[4] Zhenzhong Zhang, Haiyan Wang, Limin Xiao & Li Ruan presented paper entitled “A Statistical based Resource Allocation Scheme in Cloud” at 2011 International Conference on Cloud and Service Computing.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] Ektemal Al-Rayis & Heba Kurdi presented paper entitled "Performance Analysis of Load Balancing Architectures in Cloud Computing" at IEEE 2013 European Modelling Symposium.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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).

[16] Shikha Garg, Dr. D.V. Gupta and Dr. Rakesh Kumar Dwivedi presented paper entitled "Enhanced Active Monitoring Load Balancing Algorithm for Virtual Machines in Cloud Computing" at Proceedings of the SMART -2016, IEEE Conference ID: 39669, 5th International Conference on System Modeling & Advancement in Research Trends, 25th_27'h November, 2016.