International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

# Improved and Efficient Dynamic Load Balancing Algorithm in Cloud Based Distributed System

**Manjula K., S. Meenakshi Sundaram**

***Abstract:*** ***The basic idea of building Distributed Systems (DS) is to replenish systematic way to access disseminated resources in a decisive way. As It ensures remote access to the resources, the vital part in distributed systems is to allow proper allocation of the resources. Efficient allocation of resources amongst the various Computing Nodes (CN) can happen with the help of various scheduling policies. The various scheduling policies that are being used for efficient load balancing are Round Robin, Throttled, equally spread Load Balancing Algorithms. These are the traditional algorithms which help in performing load balance amongst CN.***

***In this research paper an Improved and Efficient Dynamic Load Balancing Algorithm is proposed which fairs comparatively well compared with the traditional existing approaches which come along with the Cloud Simulator. The simulation of the same is made using Cloud Simulator whose efficiency in terms of response time and cost is proved to be more than 5% when compared to the traditional approaches.***

***Keywords: Distibuted systems, Cloud Computing, Load Balancing, Cloud Sim.***

## I. INTRODUCTION

The distributed systems (DS) with high computing power comes with a collection of computing elements (CEs), or nodes, which allows CE's to collaboratively work to manage the loads that comes from the various sources in a effective manner. Thus a strategy has to be used to distribute the load amongst CE's which is called as load balancing policy[13]. It becomes necessary to design an effective load balancing policy which gives higher efficiency with optimal usage of resources, bandwidth, with less delay and high throughput[1]. Main advantages of any distributed systems are scalability, sharing, flexibility, transparency and many of such kind. There are many categories of DS of which cloud can be considered as collection of many distributed servers which provides services on demand to the users [2]. The technology Cloud computing is composed of the major components which are client, data centre & distributed servers. The client/end users are those who wish to avail the services from the distributed server and obtain the solutions with the optimal or less investment on the resources. Data centre can be a collection of servers hosting different applications which exist at a large distance from the clients [4], [5]. The third component of cloud is distributed servers- these are the parts of a cloud which are present throughout the web hosting different applications. The user who uses the application from cloud will be given a feel that he is the sole user of the application from the cloud on his machine with the virtue of virtualization. The three types of services provided by Cloud computing are Software as a Service, Platform as a Service and Infrastructure as a Service. SaaS provides software which makes client which need not to install on clients machine. PaaS provides platform to create an applications like database. IaaS provides computational power to user to execute task from another node [5].The fig shown below is a distributed system which will have numerous client computers, distributed servers and the data centers to process the request [14].

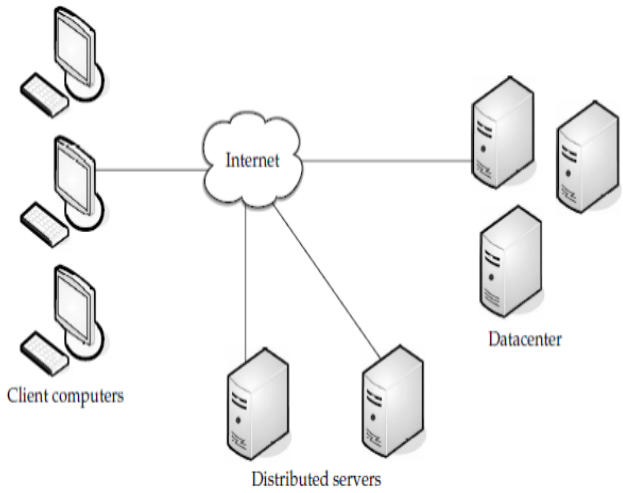


**Fig. 1: Distributed System**

## II. LOAD BALANCING

The load balancing strategy is the need of the hour in the growing networks as it manages and distributed the huge amount of data traffic to the servers in a decisive manner [10]. The various features of load balancing are: equal division of work across the entire computing elements, improved performance, user satisfaction with greater response time and lesser delays which helps to achieve greater resource utilization [6].

Manuscript published on January 30, 2020.
* Correspondence Author
**Manjula K**, Research Scholar, Department of Computer Science and Engineering, DBIT Bengaluru, affiliated to VTU Belagavi (Karnataka) India. E-mail:drsmsmk@gmail.com
**Dr. S. Meenakshi Sundaram,** Professor and HOD, Department of CSE GSSSIETW Mysuru, affiliated to VTU Belagavi (Karnataka) India. E-mail:drsmsmk@gmail.com

The typical stages for load balancing in cloud computing is show in Fig 2.

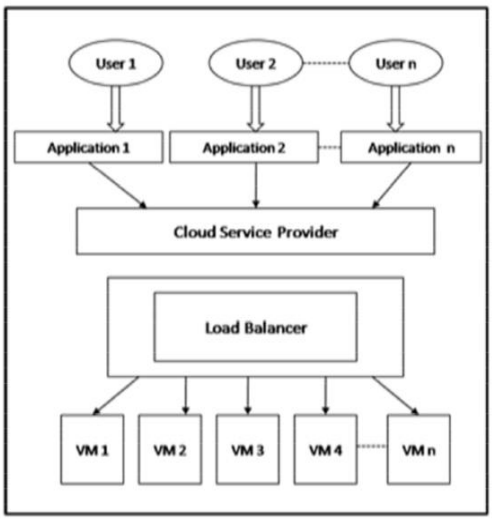


**Fig. 2: Typical Stages for Load Balancing in Cloud Computing**

In the above figure it can be seen that 'n' number of users are trying to access the VM's to run their application, such multiple users request for application on a cloud might slow up the servers, and the servers gets loaded heavily. Thus to fairly distribute the load and to take care of the server's or CN's status and to obtain higher response time proper load balancing policy can be used [7]. There has been a huge survey over the kind of algorithms used to distribute the load amongst the chosen node in the network. Basically the load balancing algorithms are broadly classified as: Static algorithms are the traditional algorithms where the decision of distribution of load will be done at compilation time and it will be fixed [3] . Dynamic algorithms are those which takes into consideration current load of the computing element and upon the obtained dynamic status of the nodes it distributes the load to the sever or the computing machine which is comparatively lightly loaded with all other nodes in the network [9]. In most of the survey it has been seen that dynamic algorithms for load balancing fairs well when compared to static algorithms. The various dynamic algorithms which exist are round robin, max min, throttled and many such algorithms have various limitations which can be in terms of delay, cost, and response time and lesser through put. The proposed dynamic approach in this paper takes care of the various drawbacks of the dynamic algorithms and thus aims at achieving higher throughput, response time, lower delay and cost.

## III. PROPOSED APPROACH

### A. Problem Statement

With the growth of network and technology user has access to the data which is in the remote place. Cloud distributed systems will allow the user to use all its services and get his/her applications run over the underlying architecture with lesser investment [8]. Multiple requests from the users to execute their applications might overload the computing elements. In the existing systems there exist various algorithms which do load balancing collecting either the initial status of the nodes or the dynamic load of the nodes [11]. But the algorithms of the existing system do take into consideration the dynamic status of the nodes along with the rearrangement of the machines in the order of their load percentage. Hence the dynamic algorithms of the existing approach fail to provide higher throughput, response time, lesser cost and delay. Thus Improved and Efficient Dynamic Load Balancing Algorithm is proposed which fairs comparatively well compared with the traditional existing approaches. The algorithm designed takes into account the Virtual machine's load and rearranges it which becomes helpful for the load balancer to identify the node which is lightly loaded.

### B. Methodology

The core entities of Load Balancing are a) Load balancer b) set of virtual machines which are used for the storage of the data. Each time the user makes a request for processing an application, the cloud service provider will monitor the nature of jobs that are already assigned to the computational entities with the help of load balancer and further take the decision on distributing the current request based on the algorithms defined for the cloud based systems. Dynamic algorithms help the load balancer in the process of decision making which node has to take up the new request based on their load status. Load balancer should use the proper dynamic algorithm for load balancing as all the nodes should be equally loaded. Before it assigns the new task to the computing element it does the scan over the VMList so that the task gets assigned to the lightly or the less burdened node in the network.

The initial steps the load balancer does are:

1. Check for the size of the tasks for which the request is sent. Size can be measured typically in terms of how much time it might take a to get completed.
2. Once the analysis of tasks for the time complexity is done a proper node has to be selected to assign the load and process it with help of algorithm.

## IV. AN IMPROVED EFFICIENT DYNAMIC ALGORITHM DESIGN AND ITS FLOWCHART

In modified efficient dynamic algorithm, the VM's are used in serial manner; instead it can be taken in parallel formation so that response time can be improved by which quick allocation of available VMs can be obtained. The sorting of the assigned load to VMs will be done load wise (the VM with less load will be on the top and greater load will be at bottom). After that the VMs are compared with the throttled capacity. So the next allocation of the request will be directly assigned to the VM which come at the top & accordingly the task allocation and task completion of the various VM's will be maintained in a tabular data structure. The proposed algorithm is as follows:

Input:

i. Tasks $T_1$, $T_2$, $T_3$… $T_n$.
ii. Initialize Virtual Machine (VM).
iii. Identify the completion time of the tasks $t_1$, $t_2$, $t_3$… $t_n$

Steps:

1: Monitor the initialization time.
2: Arrange the VM in parallel.
3: Sort the VM based on load percentage.
4: Pass the sorted VM list to the load balancer.

5: Load Balancer compares selects the VM from the sorted list and allocates the task to the chosen VM
6: Calculate the throughput of the computing element [12];

**Throughput = No of tasks completed /No of total tasks supplied**
**= fun (Tcomp) / Total no of tasks supplied ($T_n$)**

7: Record the task allocation and completion table.
8: Obtain the results in terms of response time, cost, through put
The flowchart for the proposed system is as:

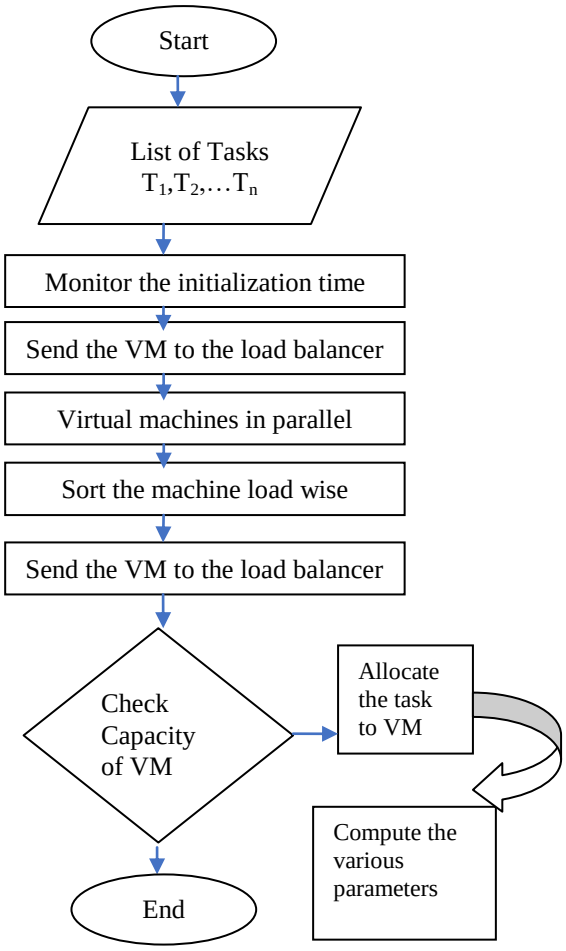


**Fig.3: Flowchart of Improved Efficient Dynamic Algorithm**

## V. IMPLEMENTATION

Simulation is carried out using Cloud Analyst [21] a package of Cloud Sim which allows to analyze the existing and newly designed algorithms performance considering various parameters like through put, response time, cost etc [14].
*Input to the algorithms:*

### A. Main configuration

The proposed algorithm is run choosing five user based regions to handle the client request to run the application. Load balancer has to choose a proper data center belonging to the regions with less amount of load. The details of the same is illustrated in the table below specifying requests/hr a user base is managing, data size of the request, off and on peak hours of the user base region are tabulated below:

**Table 1: Main configuration of User Base Regions**

| User Bases | Region | Requests/user | Data size /Request/Bytes | Pear hrs start(GMT) | Pear hrs end(GMT) | Avg. Peak users | Off peak users |
|---|---|---|---|---|---|---|---|
| UB1 | 2 | 50 | 100 | 20 | 29 | 750 | 50 |
| UB2 | 0 | 50 | 100 | 3 | 9 | 750 | 50 |
| UB3 | 3 | 60 | 100 | 3 | 9 | 1000 | 100 |
| UB4 | 0 | 50 | 100 | 3 | 9 | 750 | 50 |
| UB5 | 3 | 60 | 100 | 3 | 9 | 1000 | 100 |

In the above table requests/user indicates the total number of the task each region is currently handling, data size of the requests is in terms of bytes.

### B. Data center configuration

Every region will have sent of data centers and these data centers will handle the request and also store the data. Data centers are equipped with the virtual machines. Every access to the data centers will have a Virtual Machine cost, memory costs to carry out the computation, storage and data transfer cost. Details about the data center and the necessary configuration needed for simulation is tabulated below:

**Table 2: Data center configuration**

| Name | Region | Architecture | OS | VMM | Cost VM/hr | Memory Cost $/s | Storage Cost $/s | Data Transfer Cost $/s | Physical H/W units |
|---|---|---|---|---|---|---|---|---|---|
| DC 1 | 0 | X86 | LINUX | Xen | 5 | 0.005 | 10 | 0.1 | 2 |
| DC 2 | 3 | X86 | LINUX | Xen | 10 | 0.005 | 12 | 0.1 | 1 |
| DC 3 | 1 | X86 | LINUX | Xen | 8 | 0.005 | 11 | 0.1 | 1 |
| DC 4 | 4 | X86 | LINUX | Xen | 12 | 0.005 | 11 | 0.1 | 1 |
| DC 5 | 4 | X86 | LINUX | Xen | 10 | 0.005 | 12 | 0.1 | 1 |

can be seen from the above table that the commonly used architecture for datacenter is 80X86, Linux OS and XEN virtual machine. Cost related to VM is in terms of $.

## VI. RESULTS AND DISCUSSIONS

Once the initial configuration gets collected with the help of Table 1 & 2. Existing Algorithms like Round Robin, Equally Spread, throttled along with the proposed algorithm is run on the above configuration and the cost needed to carry out the task is noted down. The result of the same is tabulated below:

**Table 3: Virtual Machine Cost of the data centers**

| Algorithm | Total Virtual Machine Cost | Total Data Transfer Cost | Grand Total |
|---|---|---|---|
| Round Robin | 331.22 | 0.38 | 331.6 |
| Equally Spread | 225.83 | 0.32 | 226.15 |
| Throttled | 225.83 | 0.38 | 226.22 |
| Dynamic Algorithm | 220.08 | 0.32 | 220.4 |

After analyzing the results, it is noted that the proposed algorithm fairs comparatively well with respect to the existing algorithms. The proposed algorithm is efficient by more than 5% when compared to the existing approaches. The graph depicted below shows that the proposed algorithm fairs well in terms of VM cost when compared to existing approaches. In the below graph since the data transfer cost is negligible, it is represented below the scale of 60.

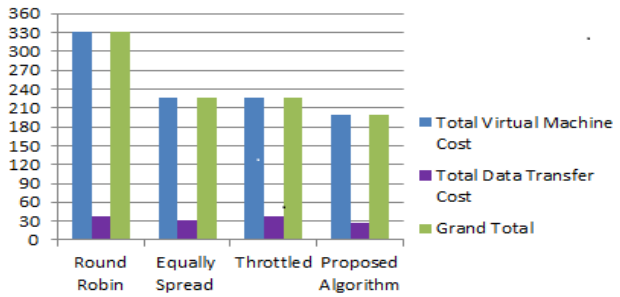


**Fig.4: Performance analysis of the algorithms**

**Table 3: Virtual Machine Cost of the data centers**

| Algorithms | Overall Avg Response Time in Hrs |
|---|---|
| Round Robin | 81.31 |
| Equally Spread | 77.49 |
| Throttled | 81.3 |
| Proposed | 70.25 |

The table 3 shown above is obtained once the algorithms are run with the datacenter and user base regions configurations. It can be seen that the proposed algorithm's response time is fair comparative with the existing algorithms. The average response time by the algorithms is depicted in fig 5

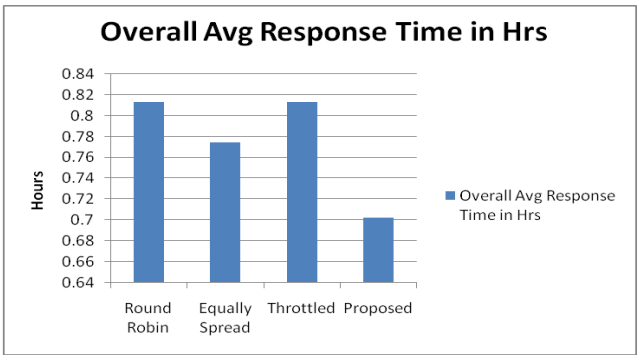


**Fig.5: Average Respose time of the algorithms**

The proposed algorithm has an improvement of around 7% in terms of response time when compared with the existing algorithms.

## VII. CONCLUSION & FUTURE WORK

Load balancing always has been an important issue which if not addressed will lead to unbalanced network. After going through the huge survey on the static and dynamic algorithms used for load balancing in distributed systems, it has been identified that the papers documented considers on the compilation load or the current or dynamic load of the nodes but in the proposed approach along with considering the dynamic load the VM capacity is also considered and are sorted in the ascending order. The proposed approach thus has given a fair result of more than 5% when compared with the existing algorithms. The proposed algorithm can be still improved in future by incorporating the cooperative game amongst the node, which might improve the results as the way the nodes communicate in the distributed systems play a vital role.

## ACKNOWLEDGEMENT

I would like to thank don bosco institute of technology for giving me research environment to carry out my work.

## REFERENCES

1. Dynamic Load Balancing in Distributed Systems in the Presence of Delays: A Regeneration-Theory Approach Sagar Dhakal, Majeed M. Hayat, Senior Member, IEEE, Jorge E. Pezoa, Cundong Yang, and David A. Bader, Senior Member, IEEE IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 18, NO. 4, APRIL 2007
2. G. Pallis, "Cloud Computing: The New Frontier of Internet Computing", IEEE Journal of Internet Computing, Vol. 14, No. 5, September/October 2010, pages 70-73
3. A. Khiyati, M. Zbakh, H. El Bakkali, D. El Kettani "Load Balancing Cloud Computing: State Of Art", IEEE, 2012.
4. Mladen A. Vouk, "Cloud Computing – Issues, Research and Implementations", Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, June 23-26, 2008, Cavtat, Croatia
5. J. Sahoo, S. Mohapatra and R. lath "Virtualization: A survey on concepts, taxonomy and associated security issues" computer and network technology (ICCNT), IEEE, pp. 222-226. April 2010
6. Giuseppe Aceto, Alessio Botta, Walter de Donato, Antonio Pescapè, "Cloud monitoring: A survey", SciVerse ScienceDirect's Computer Networks 57, PP- 2093–2115, ASOC 1894 1–12, © 2013 Elsevier B.V
7. Roughgarden T, Tardos E. How bad is selfish routing? Proceedings of the 41st IEEE Symposium on Foundations of Computer Science, Redondo Beach, CA, November 2000; 93-102
8. Bharadwaj v, Ghose D , Mani V, Robertazzi TG Scheduling Divisible Loads in Parallel and Distributed Systems. IEEE Computer Society Press: Loas Alamitos, CA, U.S.A., 1996
9. A modified round-robin load-balancing algorithm for cluster-based web servers IEEE, Zongyu Xu ; Xingxuan Wang
10. Sidra Aslam Munam, Ali Shah "Load Balancing Algorithms in Cloud Computing: A Survey of Modern Techniques" 2015 National Software Engineering Conference (NSEC 2015),IEEE, pp:30-35
11. Shinde, Vinayak D., Anas Sami Ahmed Dange and Muhib Lambay. "Load Balancing Algorithms in Cloud Computing." (2016).
12. Load Balancing Algorithm Based on Estimating Finish Time of Services in Cloud Computing Nguyen Khac Chien, Nguyen Hong Son, Ho Dac Loc , 2016 18th International Conference on Advance Communication Technology (ICACT)

13. Dynamic Multi-User Load Balancing in Distributed Systems Satish Penmatsa and Anthony T. ChronopoulosThe University of Texas at San Antonio Dept. of Computer Science, IEEE 2007
14. Cloud Computing Introduction ,Greg's Web Development Blog

## AUTHORS PROFILE

**Mrs. Manjula K,** currently working as Assistant Professor in Don Bosco Institute of Technology, Bangalore in the department of Information Science and Engineering. She has completed her master's from Visveswarya Technical University (VTU). She is pursuing her Research under Visveswaraya Technological University.

**Dr. S. MeenakshiSundaram,** is currently working as Professor and Head in the Department of Computer Science and Engineering at GSSS Institute of Engineering and Technology for Women, Mysuru. He obtained his Bachelor Degree in Computer Science & Engineering from Bharathidasan University, Tiruchirappalli in 1989, M.Tech from National Institute of Technology, Tiruchirappalli in 2006 and Ph.D. in Computer Science & Engineering from Anna University Chennai in 2014. He has published 53 papers in refereed International Journals, presented 3 papers in International Conferences and has delivered more than 40 seminars. He is a reviewer of Springer – Soft Computing Journal, International Journal of Ah Hoc Network Systems, Journal of Engineering Science and Technology, Taylor's University, Malaysia and International Journal of Computational Science & Engineering, Inderscience Publishers, UK. He has organized more than 40 seminars / Workshops / FDPs. He has attended more than 45 Workshops / Seminars. His area of interest includes Computer Networks, Wireless Communication, Software Engineering, Optical Networks and Data Mining. He is a Life Member of Indian Society for Technical Education (ISTE) and a member of Computer Society of India (CSI). He has 30 Years of teaching experience and 10 years of research experience. Currently 7 research scholars are pursuing Ph.D. in VTU Belagavi, India under his guidance.