An Efficient Approach for Load Balancing in Cloud Computing using Maximum, Minimum and Average Distribution

Manpreet Kaur[1], Dr. B.K. Verma[2]
[1]*Chandigarh Engineering College, Landran*
[2] *Chandigarh Engineering College, Landran*
(E-mail: manukaur697@gmail.com)

***Abstract*—** Load balancing and job scheduling always remains the prominent topic for research work. This study focuses on the issues that exists in traditional load balancing algorithms in cloud computing. The issues like les availability of VMs, issue of starvation etc. In order to resolve the issues of traditional load balancing algorithms, the novel approach i.e. EDLB (Equally Distributed load balancing) algorithm is developed. The EDLB performs the job assigning to the VMs on the basis of two different criteria. The first criteria are to measure the caliber of the VM to handle the jobs and second criteria is to measure the jobs whether they are lightweight or heavy weight. Then on the basis of the evaluated criteria the lightweight jobs are assigned to the low level VMs and the heavy weighted jobs are allocated to the highly caliber VMs. The simulation is done in JAVA for time shared and space shared approach. The obtained results prove that the EDLB outperforms the traditional weighted round robin technique and length based weighted round robin algorithm.

Keywords— *Cloud Computing, Load balancing, round robin algorithm, starvation, time shared, space shared.*

I. INTRODUCTION

Cloud Computing is an enhanced technique for utilizing the online provisions like storage or services in very advanced way. In cloud computing service the utility computing plays the vital role. Hence, the computing is described differently by different researchers [1]. As per the National Institute of Standards and Technology (NIST) the Cloud computing technique may be described as "An approach which is used for initiating the suitable, on-demand access of network in order to share the different computing resources in the network. Example of different configurable resources as follow: networks, memory unit, servers, and services [2]. Various resources that can either be virtualized, scalable or distributed hardware-based or software-based can be quickly shared with least efforts and minimum interaction with the service providers [3] from anywhere in the world. Numerous nodes that are involved in cloud computing are assigned with tasks in an organized manner to enhance its performance through quick request processing [4].

II. LOAD BALANCING IN CLOUD COMPUTING

Load balancing is a procedure that is performed to allocate or assign the coming jobs from the cloudlets to the available VMs. This is done to achieve the proper utilization of resources [5]. Along with the resource utilization, the load balancing also plays an important role to distribute the load of jobs equally to the system and to compensate the system failure or device failure. For example, if one of the virtual machines (VMs) aborts the job processing due to any kind of failure then in such case the working of whole system will not gets effected as the load of the failed VM is shifted to the other VMs to execute the aborted jobs. The load balancing is done by using load balancer; the role of the load balancer is to distribute the jobs in an equal manner so that none of VMs remain idle. The job distribution is done by following a strategy that is referred as load balancing algorithm [6].

The network load, CPU load and memory capacity of each server associated with some website traffic altogether contributes to the load over entire network system. The main objective of load balancing algorithms is to achieve proper load distribution situations in which each network component is assigned with equal amount of load at any point of time. It eliminates the situations which involve under- loaded and over-loaded resources in order to ensure efficient resource utilization. The load distribution process balances load of entire network by analyzing the schedule of each server. The servers that are less likely to be busy are assigned with new requests. On receiving requests from clients the load balancer circulates it through all connected servers. Once it reaches last server the request is again circulated forward and backward direction forming a loop pattern. In the absence of load balancer, the client is required to wait for processing of its request. The process of load balancing considers information regarding computational rate, job arrival rate, job waiting etc. which is required to be communicated among processors. The loss of significant can occur in case of load balancer failure which may result in disruptive network.

A. Types of Load balancing algorithms

Load balancing algorithms are classified based on the network component responsible for initiation of process into following classes [4]:

• **Sender Initiated:** Situation where the load balancing algorithm is started by the sender.
• **Receiver Initiated:** In this case receiver is responsible for algorithm initiation.
• **Symmetric:** Both sender and receiver initiates the load balancing algorithm.

On the basis of present system state load balancing algorithms fall under two distinct categories that are elaborated as following [4]:

1) **Static Load Balancing Algorithms**

Static load balancing algorithms does not involve the system information regardless of its present state due to which they are relatively less complex than dynamic algorithms. The system information pertaining to its global status is required as the primary information for load distribution process irrespective of present state and behavior of individual nodes. These algorithms are preferable when the communication and computational requirements are predefined which are further decomposed into tasks. The task-processors are allocated with these tasks prior to the initiation execution of parallel application**.**

2) **Dynamic Load Balancing Algorithms**

Dynamic load balancing algorithms are current state dependent algorithms due to which are considered to be relatively more flexible than static algorithms. Such algorithms do require any prior knowledge regarding system as it only takes current state into its account for functioning. Dynamic load balancing algorithms can be applied either as distributed or as non-distributed in distributed system. The execution of algorithm is performed by each node in the system whereas all servers are responsible for performing load balancing in a shared manner under distributed approach. The communication among nodes can be cooperative or non-cooperative. Under cooperative interaction all the nodes work in coordinated side by side manner which produces enhanced response time of entire system. Non-cooperative interaction involves nodes working in independent manner to achieve global local [8].

III. CHALLENGES & ISSUES OF LOAD BALANCING

Even through cloud computing is widely used now days but still the research is in its starting stage [7]. So before explaining the load balancing algorithm for cloud computing it is required to identify some key challenges and issues that affect the performance of load balancing algorithms [17].

- **Virtual machine migration**

The idea is to imagine a machine as a set of files or a file. It is possible to decrease the load on over loaded machine by moving the virtual machine among them in effective way. The main motto is to distribute the all type of load in a datacenter. The challenge is to remove and avoid drawbacks of cloud computing system when the load is dynamically distributed by virtual machine.

- **Stored data management**

Another key requirement is the storage of data. So how can data be distributed in the cloud system with most appropriate storage and fast access?

- **Emergence of small different datacenters for cloud computing**

A small size of data center can be more beneficial just because it will consume less electricity and cheaper than large one. And load balancing is showing as a global scale issue for certifying proper response time with optimal resource utilization and distribution.

- **Spatial distribution of the cloud nodes**

Some algorithms are proposed just for nearly located nodes in which communication delays are insignificant [17]. But still it is a issue to design an efficient load balancing algorithm which can be formulated properly for spatially distributed nodes [5].

- **Storage and Replication**

A full replication algorithm is not much beneficial for efficient storage utilization in a system. This is just because the same amount of data will be kept in all replicated nodes. Full replication algorithms cause unreasonable costs with requirements of large storage.

- **Algorithm complexity**

Load balancing algorithm is preferred to be less complicated in terms of operations and execution (implementation). A negative implementation complexity will lead to a extra complex process. Furthermore, for monitoring and controlling the implementation, algorithms require higher communication, more information and delays may cause more bottlenecks and then efficiency discards [17].

- **Point of failure controlling**

Some algorithms (centralized algorithms) provide effective mechanisms for processing load balancing in a particular pattern. But the issue is that there is only one controller for the entire system. In such condition, if the controller fails, then the entire system fails [17].

IV. PRESENT WORK

In traditional load balancing process, the load balancing was done by using the traits like First Come First Served (FCFS), Round Robin, Priority based job scheduling algorithms etc. All these algorithms work on various criteria such as FCFS follows the approach of first come first serve. In FCFS the job which comes first for the execution is allotted to the first VM and the job comes at last is allotted to the remaining VM. It did not evaluate other factors such as whether the allotted VM is capable enough to process the job or not, and whether the assigned job is highly prior than other coming jobs. To overcome the issue of FCFS, the round robin algorithm is introduced. In RR the job assignment is done on the basis of the time interval. An equal time interval is allotted to the jobs for execution. Once the interval is completed and the job is incomplete then the current job is resumed in next time interval. It also suffers from the issue that all the jobs get equal execution time irrespective to their priority. Then to resolve this issue, the concept of priority is added to the round robin in

which the job processing is done on the basis of the priority and execution time as well. After this a large number of scheduling and load balancing techniques have been developed till now, but fails to generate the qualitative output as some traditional load balancing mechanisms suffer from the problem of starvation, less availability of VMs, resource wastage etc. The propose EDLB mechanism is developed to overcome the issues of traditional load balancing algorithm. Thus a new solution has been proposed under this study. The proposed solution is based on two criteria:

1. The load distribution is done on the basis of the capability of the available virtual machines.
2. The load distribution is done equally among the virtual machines. This reduces the risk of starvation.

The methodology of proposed load balancing algorithm is as follows.

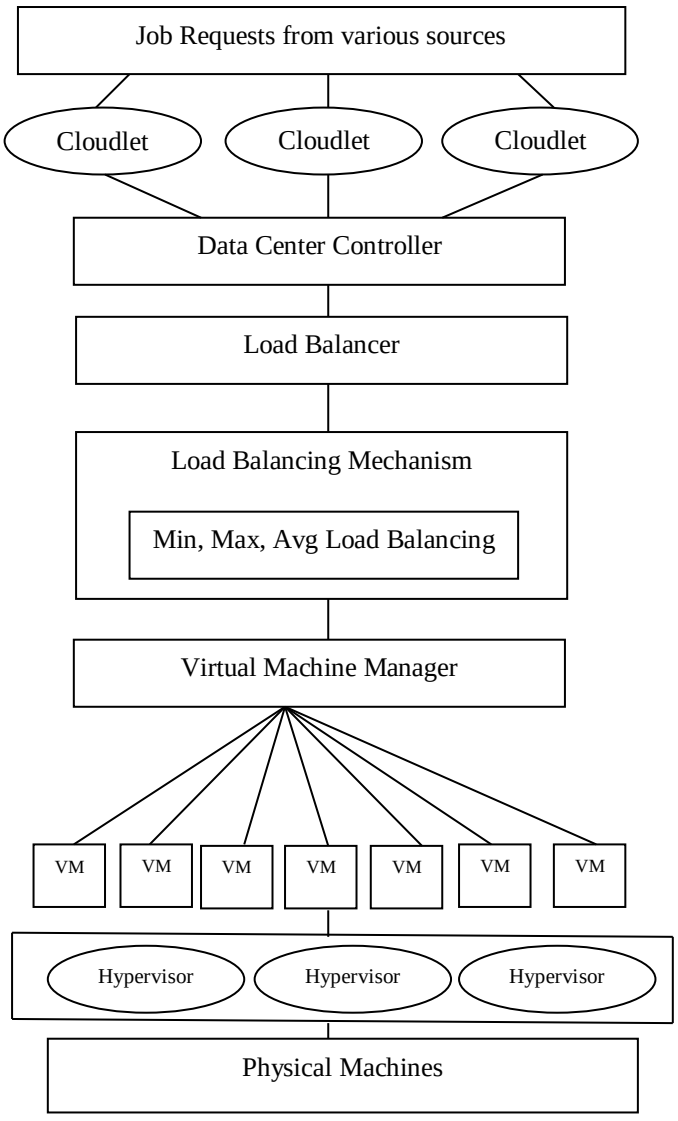

Figure 1 Framework of propose work

The figure 1 delineates the methodology of propose load balancing algorithm. The virtualization in propose framework comprised of a large number of VMs, a set of hypervisors and physical machines. Hypervisor is a setup that separates the OS and applications from basic computer hardware. This separation of hardware from OS facilitates the hardware to perform its task independently.

As per the figure the flow starts with the jobs that are coming from a source to various cloudlets. Cloudlets are used to define the jobs or tasks. Then it is the responsibility of the cloudlets to pass these jobs to the data center controller. Then the load balancer distributes these jobs equally to the available virtual machines. In propose work, the job distribution is done by using the proposed load balancing algorithm which firstly categorizes the available virtual machines into two categories on the basis of low level and high level VMs. The high level VMs comprised of those VMs which are capable to handle the heavy jobs and the low level VMs comprised of VMs that are capable to handle the light weighted jobs. After then, arrived jobs are divided into three parts i.e. $Min, Max\ and\ Avg\ jobs$. The jobs that are varying between $Min\ and\ Max\ jobs$ are allotted to the low level VMs and the Avg jobs are allotted to the high level VMs. After job allocation to the VMs, the VMs perform the jobs execution. Here the hypervisors are used to separate the installed VMs from computing hardware i.e. physical machines.

V. IMPLEMENTATION

This section of the study is organized to have a brief review to the performance of the propose ELDB load balancing algorithm. The ELDB is implemented in JAVA and the graphs in this section are derived to depict the performance of ELDB in the terms of task completion time with respect to space shared and time shared. In space shared, the jobs are executed by the CPU in a sequential manner whereas in time shared mechanism, the job execution is done on the basis of the time interval allotted to the jobs. For the purpose of analysis the comparison is done among EDLB, Length based weighted round robin and weighted round robin algorithm. The graph in figure 2 shows the contrast between completion time of the EDLB and traditional techniques on the basis of the space shared task migration mechanism.

The x axis in the graph shows the number of VMs that varies from 10 to 100 with the interval of 10. The y axis in the graph depicts the task completion time that ranges from 0 seconds to 160000 seconds. The bar in blue stands for EDLB, the bar in green depicts the performance of weighted round robin and the bar in red shows the performance of length based weighted round robin algorithm. On the basis of the observations it is concluded that the EDLB takes less completion time in comparison to the traditional techniques. The time taken by the EDLB, weighted round robin and length based weighted round robin to complete the assigned jobs are shown in table 1.

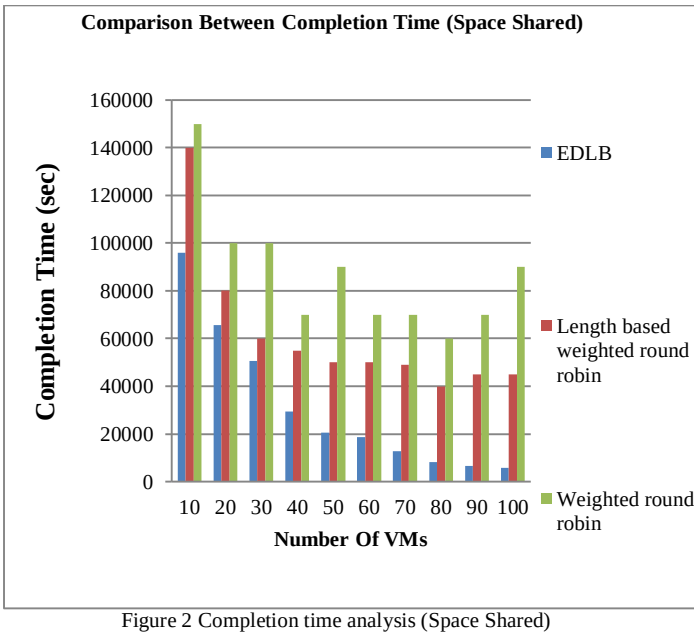

Figure 2 Completion time analysis (Space Shared)

Table 1. Analysis of Task Completion Time (Space Shared)

Number Of Virtual Machines	EDLB	Length based weighted round robin [1]	Weighted round robin [1]
10	96017.98095	140000	150000
20	65537.03589	80000	100000
30	50560.83268	60000	100000
40	29380.31413	55000	70000
50	20661.1664	50000	90000
60	18651.9156	50000	70000
70	12791.63399	49000	70000
80	8160.206938	40000	60000
90	6544.49	45000	70000
100	5866.893546	45000	90000

The graph in figure 3 defines the comparison analysis of completion time with respect to the time shared.

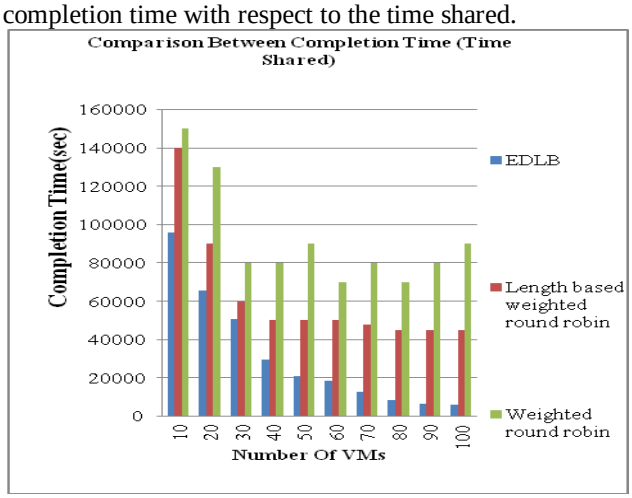

Figure 3 Completion time analysis (Time Shared)

The numbers of VMs are 100 and completion time is 160000 seconds. The completion time of weighted round robin technique is higher in comparison to EDLB and length based weighted round robin. Table 2 depicts the job completion time for time shared mechanism.

Table 2. Analysis of Task Completion Time (Time Shared)

Number Of Virtual Machines	EDLB	Length based weighted round robin [1]	Weighted round robin [1]
10	96017.981	140000	150000
20	65537.0359	90000	130000
30	50560.8327	60000	80000
40	29380.3141	50000	80000
50	20661.1664	50000	90000
60	18651.9156	50000	70000
70	12791.634	48000	80000
80	8160.20694	45000	70000
90	6544.49	45000	80000
100	5866.89355	45000	90000

VI. CONCLUSION

This study develops EDLB, an approach of load balancing in cloud computing. The EDLB distributes the load on the basis of the caliber of the VMs and the nature of the jobs. The analysis of the EDLB is done on the basis of time shared and space shared mechanism. On the basis of the simulated results, it is observed that the EDLB consumes less time to complete the jobs that are allotted to the VMs. The length based weighted round robin and weighted round robin algorithm is considered for comparison analysis. The minimum completion time takes by the ELDB is 6544.49 in both cases i.e. time shared and space shared. For the purpose of analysis a large number of VMs and cloudlets are considered.

The facts observed from the results shows that the EDLB has outstanding results in comparison to the weighted round robin and length based weighted round robin algorithm. But still more amendments are possible in EDLB to make in more efficient and qualitative. In future, more enhancements could be done by collaborating the load balancing algorithms.

REFERENCES

[1]. D. Chitra Devi et al, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Non preemptive Dependent Tasks", Hindawi, Vol. 2016, Pp. 1-14, 2016

[2]. Einollah Jafarnejad Ghomi et al, "Load-balancing algorithms in cloud computing: A survey", Journal of Network and Computer Applications, Vol. 88, Pp. 50-71, June 2017

[3]. Lipika Datta, "Efficient Round Robin Scheduling Algorithm with Dynamic Time Slice", I.J. Education and Management Engineering, Vol. 2, Pp. 10-19, 2015

[4]. Saraswathi Seemakuthi et al, "A Review on Various Scheduling Algorithms", International Journal of Scientific

& Engineering Research, Vol. 6, No. 12, Pp. 769-779, December 2015

[5]. Shalini Joshi et al, "Load balancing in cloud computing: Challenges & issues", Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on, May 2017

[6]. Ojasvee Kaneria et al, "Analysis and improvement of load balancing in Cloud Computing", ICT in Business Industry & Government (ICTBIG), International Conference on, April 2017

[7]. Monjur Ahmed et al, "Cloud Computing And Security Issues in The Cloud", IJNSA, Vol 6, Issue 1, Pp 25-36, 2014

[8]. Michael armbrust,"A view on Cloud computing", communication of the ACM, vol 53(4), 2009

[9]. Ling Qian,"Cloud computing-An overview", springer, Pp 626-631, 2009

[10]. Matthias Sommer et al, "Predictive Load Balancing in Cloud Computing Environments Based on Ensemble Forecasting", Autonomic Computing (ICAC), 2016 IEEE International Conference on, September 2016

[11]. Aarti Vig et al, "An Efficient Distributed Approach for Load Balancing in Cloud Computing", Computational Intelligence and Communication Networks (CICN), 2015 International Conference on, August 2016

[12]. Sidra Aslam et al, "Load balancing algorithms in cloud computing: A survey of modern techniques", Software Engineering Conference (NSEC), 2015 National, February 2016

[13]. Reena Panwar et al, "Load balancing in cloud computing using dynamic load management algorithm", Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on, January 2016

[14]. Agraj Sharma et al, "Response time based load balancing in cloud computing", Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on, December 2014

[15]. Velagapudi Sreenivas et al, "Load balancing techniques: Major challenge in Cloud Computing - a systematic review", Electronics and Communication Systems (ICECS), 2014 International Conference on, September 2014