Performance Comparison of Load Balancing Architectures in Cloud Computing Environment

Rajat[1], Dr. Sanjeev Kumar[2]
[1] M.Tech. Scholar, Department of CSE, GJU S&T, Hisar, India.
[2]Assistant Professor, Department of CSE, GJU S&T, Hisar, India

Abstract- Cloud Computing is the new style of computing which provide different type of services like-servers, storage, application, database, networking, software and more over the internet on the user's computer or devices. Cloud computing is in demand and it is also getting progressed constantly which led to increase the traffic on the cloud servers. So load balancing become an essential thought in the cloud computing environment. There are three different architectures centralized, decentralized and hierarchical are available for load balancing in the cloud environment. In centralized there is a load balancer in middle that make all the decisions and in decentralized many number of load balancers make decision collectively whereas in hierarchical load balancers are placed in a tree like structure. In this paper performance of all the three architectures are compared in the public cloud environment. The simulation result of these architectures shows that the hierarchical architecture of load balancer is the best for the public cloud condition and for the further research scope it might be tried that result are same for the other sort of cloud or not.

Keywords –Web 2.0; Virtualization; Scheduling; Load Balancer; Public Cloud; Homogeneous and Heterogeneous Network;

1. Introduction

Now a days, computing turns out to be consistently more vital and more utilized. The measure of data traded over the networks or put away on a PC is expanding consistently. Accordingly, to handle or process this huge amount of data and to meet the needs of organisations more PC hardware is needed. To take full advantage of their venture, the over-equipped companies and organisations open the infrastructure they have to others by exploiting the Internet and related advances like web 2.0 and other rising technologies like virtualization by figuring a brand new model: the cloud computing[1].

Cloud computing is an internet based model in which few self contained abstraction such as infrastructural elements, application development and deployment environments, self contained software applications and all these abstraction are delivered as a service similar to utilities such as water and electricity. And all these services are available anywhere at any time only thing user need is an internet connection [2]. The definition of cloud computing provided by National Institute of Standards and Technology (NIST) says that: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [3].

Cloud computing provides various benefits to the organizations and the individuals by reducing cost and increasing flexibility. There are a lot of companies and an organization that uses the cloud and each one want the best performance, so there is a need of load balancing. Load balancing is the technique of distributing the load among different devices in any network[4]. Accordingly load should be distributed over the devices in the cloud environment so that each resource does the equal amount of work at any point of time. In the event that likewise of a node failure the system ought to ordinarily adjust the undertakings influenced to the inadequate asset so accessibility is protected and the client still could benefit from cloud capacities immediately in execution [5], [6].

Main aim of load balancing are scalability, adaptability, cost effectiveness and to maintain steadiness. In cloud computing there is lot of techniques exists for load balancing. In a cloud environment Load balancers can be placed or deployed in three ways that are centralized, decentralized and hierarchical architecture [7].

The main aim this paper is to compare performance of all the cloud computing architecture in public cloud environment. This paper begin by quickly portraying the cloud computing that what it is and which services cloud provide to a consumer. In the section 2 load balancing architecture are discussed in detail. Third chapter focuses on the related work that is already done and reviewed to do following work. The research methodology used and experimental sittings that are done in order to compare the performance are stated in section 4. Chapter 5 shows the results after simulation. Finally section 6 proposes the conclusion and the scope for future work.

2. Load Balancing Architectures

There are various ways in which nodes are established in a network. On the bases of spatial distribution of nodes that which node is responsible for the load balancing can be divided into three types of architectures that are:

- Centralized Load Balancing
- Decentralized Load Balancing
- Hierarchical Load Balancing

In **Centralized Load balancing** all the allocations of load balancers and decisions are made by a single node that which cloud asset does which assignment and via using which technique. So a single central node is the main factor to handle the load balancing of the whole system those results into the less overall time. As compare to distributed method it has less overhead and it work on the bases of global view of the system [8].
Scalability of this type of architecture is poor and it also constitutes a single point of failure. In this there is lot of chances that central node gets failed and it is very difficult to get recovered from a node failure. In this central node decide that which algorithm (Static or Dynamic) should be applied for load balancing, based on the overall knowledge of cloud network that is stored in the central node. In this approach an incorporated load balancer is enforced within the client space. This frequently gathers workload information from the all alternative workstations and at a point when a new job arrive it directs the job to appropriate processor, based on the collected workload information [9].

In **Decentralized Load Balancing** technique to make the precise load balancing decisions there are a number of load balancer monitor the system workload instead of single node in centralized technique. This technique offers awesome adaptability and versatility. Each Load balancer in the framework keeps up neighborhood information base to guarantee productive distribution of jobs in static condition and re-distribution in dynamic condition.
In decentralized load balancing each load balancer in a framework may use different algorithm for the distribution of the jobs. In this type of scenario, there is less chances of node failure. Hence the framework is more reliable and fault tolerant and none of the node get overloaded [10].

These types of architectures can easily tolerate the failure and have less administrative burden. If any single node stops working then the whole network still works well. Moreover, these networks can also tolerate the multiple node failure over time. These networks only require a little intervention from human operator because these are self-healing and self-organizing network [11].

In **Hierarchical load balancing** load balancers are placed in a tree like structure in which main load balancer (parent node) receives all request and spread these request to different load balancer (children) at different level. The load balancer at different level can use different algorithm. In this every node is managed and balanced by the parent node. The parent node is responsible for scheduling, allocation and for distributing the load to its child node [12]. It is combination of centralized and decentralized so take the advantage of both the technique.
Hierarchical architecture easily can be used in large homogeneous or heterogeneous networks. This architecture also has some disadvantage that it is complex and hard to implement. It also includes extra overhead of sort out between the load balancers themselves and these are less fault tolerant[13].

3. Related Work

In traditional computing system it is assumed that communication overhead between processors is almost same or it is ignored because at that time task grain size and number of hopes that have to be traversed are small[14-16]. But in the case of cloud computing environment data size and the task grain size is big which cause big communication delay and reduce the accuracy of load balancing and at last that lead to aging problem of gathered information.
In centralized load balancing there is a central computer that collects all the information about the system and makes it global. It works well when the number of processor is up to few thousand [17]. However in the case of large distributed systems it is nonscalable and have other limitations like [18], [20]: (1) central node need a large amount of storage capacity to store global information: (2) With large number of processors central node communication can become bottleneck: (3) the process of gathering global information may become an expensive process. As compare to centralized systems, decentralized load balancing system is more scalable in large distributed system [21], [22]. In this each processor share load information with its neighbor processors only [23]. However it also has limitations in practice. For example in case of non preemptive scheduler it suffers from aging of workload information. This delay leads the load balancing to make poor decision and to have large response time. In hierarchical load balancing strategy processors are divided into groups and the groups are divided into a

hierarchy. This system deal with scalability problem and also reduce the response time and the memory required for the storing information. In this a tree like structure is formed and each processor at lowest level of the tree send the work load information with parent processors respectively. But in this as the size of load balancing meta data grows, the cost for load balancing also get increased [24-27].

4. Simulation Setup

As in cloud computing still there is uncertainty that which load balancing architecture works better in which environment. So to contribute in that main aim of this paper is to compare all the three architecture in public cloud environment and to find answer to the question that: out of centralized, decentralized and hierarchical load balancing architecture which one have the best performance in the identical cloud environment when analyzed on the bases of two measure that are server load and response time?

To get the answer of the above mentioned question a simulation is done in a simulation tool that includes following steps:

- Firstly all the critical elements like number of nodes in framework, load balancing technique to be used, and how load balancers are placed in the framework are identified.
- Number of nodes are varied a number of time to stimulate it better like an original cloud. Because in reality number of user may increase or decrease at any time.
- Two simple algorithm are selected first one is Round Robin in which a slice of time is given to each process and second one is number of connection that try to make equal number of request on each server.
- Two measures are selected to analyze the performance that are: Server Load and Response Time.
- Different scenarios are implemented for different architectures.
- Run the simulation for a number of times to get improved result and accept the mean outcome

The simulation framework that was network simulator Opnet modeler is installed and run on a Compaq laptop with Intel Dual Core CPU, 2.13 GHz, 2 GB RAM, Window 8 of 64 bit as an operating system. For simulation a database type service with application type of heavy load database query is run on the public type of cloud Model. There are 5 total numbers of servers for fulfilling the request. In centralized one load balancer is placed in middle to balance the whole load and in decentralized the entire requests are divided between two load balancers. Three load balancers are placed in a tree like structure to form a hierarchical architecture. In this one is work as root and other two works as child load balancers. Each architecture is simulated for 1 hour with many repetitions on different number of users. The Following table shows the simulation setting for different scenario:

Table 1. Simulation Setup

Parameter	**Value**
Service Type	Database
Application Type	Heavy load database query
Cloud Deployment Model	Public
Simulation Time	1 Hour
Number of Server	5 Servers
Number of Client	20, 40 and 90 clients
Number of load balancer (Centralized)	1 Load Balancer
Number of load balancer (Decentralized)	2 Load Balancer
Number of load balancer (Hierarchical)	3 Load Balancer

The numbers of clients are 20, 40 and 80. Centralized, decentralized and hierarchical include 1, 2 and 3 load balancers respectively. Which load balancer use which algorithm is shown in the following table:

Table 2. Simulation Balancers Algorithm

Scenario	**Load Balancer**	**Algorithm**
Centralized	Load Balancer	Round Robin
Decentralized	Load Balancer 1	No of Connection
	Load Balancer 2	Round Robin
Hierarchical	Load Balancer prnt	Round Robin
	Load Balancer CH1	No. of Connection
	Load Balancer Ch2	No. of Connection

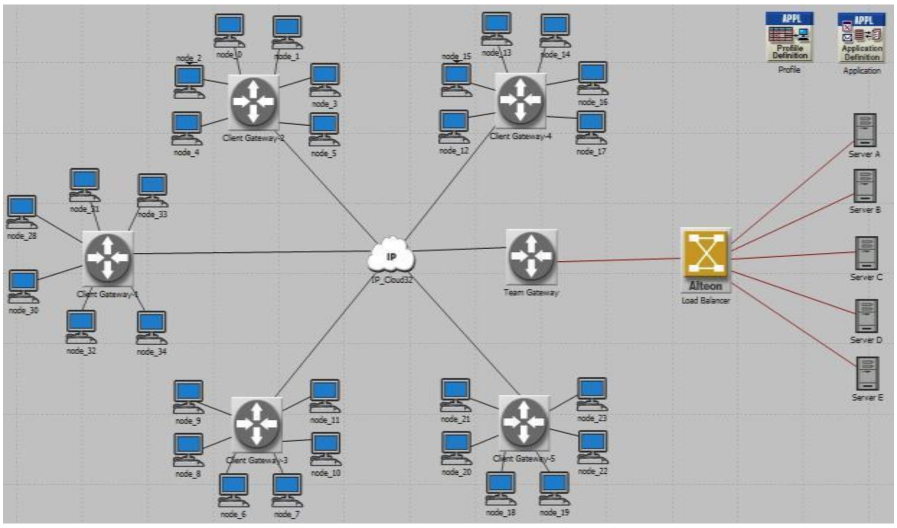

Fig.1: Centralized Load Balancing Architecture

As shown in Fig.1, The centralized has one load balancer that is using the round robin algorithm to make the decisions. When any host make a request, the request goes to his client gateway and then to the team gateway through ip_cloud. The team gateway sends the request to load balancer, then it is decided that which server serves the request.

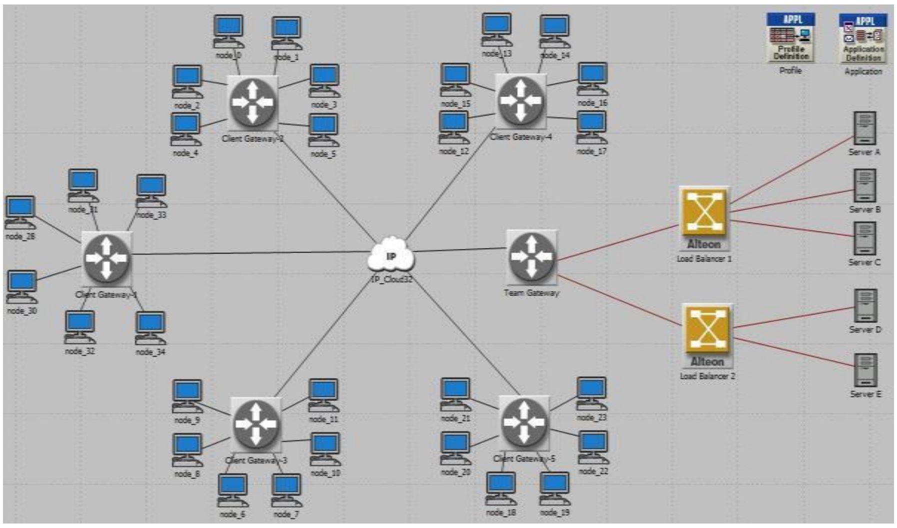

Fig.2: Decentralized Load Balancing Architecture

As shown in Fig.2, The decentralized has two load balancers, one of them is making decision on the basis of number of connection where as other is using the round robin algorithm to make the decision. In this same as in centralized all the requests get together at team gateway and after that the requests are handled to one of the load balancers. Then each load balancers keep track of the load of some of the servers and on the basis of that request is submitted to one of the servers.

As shown in Fig.3, The hierarchical, root node make use of round robin algorithm and the child nodes are making decisions on the basis of number of connections. In this firstly all the requests from gateway is given to the load balancer at the root node, this load balancers have information about the status of its child load balancers. On the basis of that information request is handled by one of the child load balancer then same as in decentralized these child node have information about some of the servers and then these requests are submitted to one of the server appropriately.

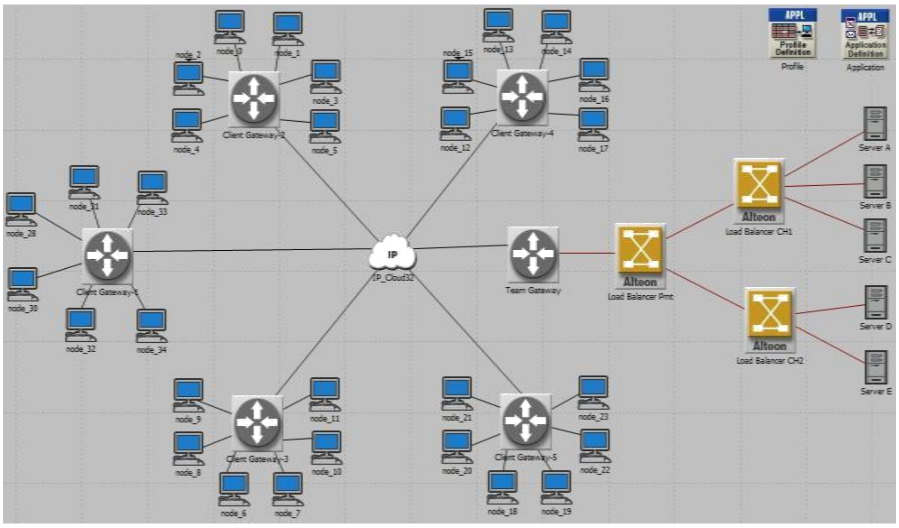

Fig 3: Hierarchical Load Balancing Architecture\

5. Result and Discussion

The simulation is done in OPNET tool on the each type of architecture.. The results are taken on the bases of Server Load and Response Time.

- Server Load

Server Load is used to show that how many processes are waiting to get the server. If the number of processes in waiting get increased performance is not good. Fig 4- Fig.6 showing the server load in each architecture by varying the number of clients from 20 to 80. For different number of clients average server load is computed. Average server load is calculated as the number of running processes by each server/ second (request/ second).

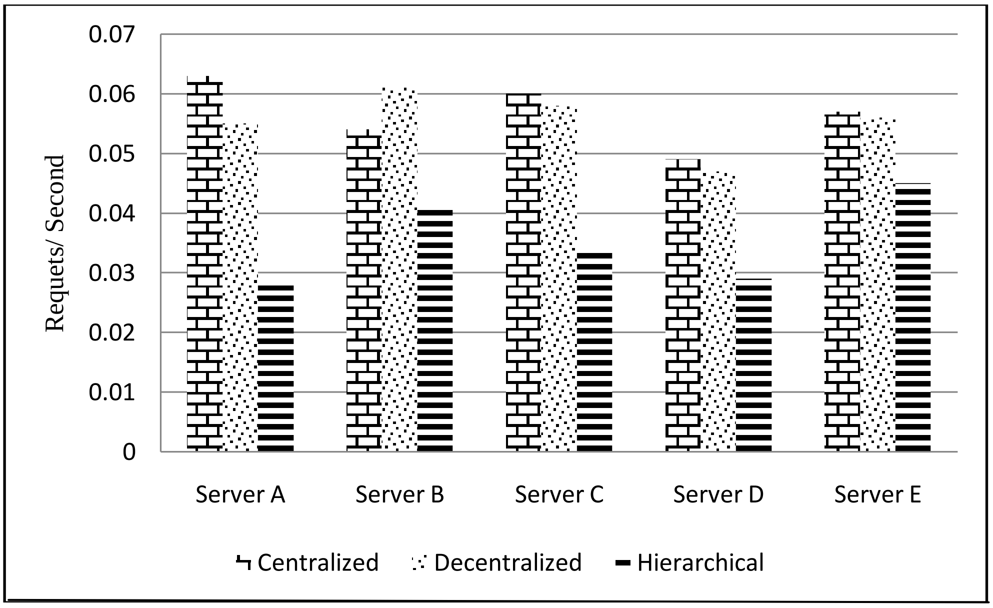

Fig. 4. Load on each server in each architecture when the number of client was 20

Figure 4 is showing the average server load on each server for each type of architecture when the number of users are set to 20. It is found that centralized architecture have more load than decentralized in the case of server A, C, D and E. Only in case of server B decentralized is showing more load. But for all the servers when numbers of users are 20, the hierarchical architecture outperformed the centralized and decentralized architecture.

Fig. 5 shows the status of each server in terms of server load when the number of clients are 40. It shows that centralized and decentralized almost giving same results, on some servers centralized is good while on some others decentralized is better than centralized. As it is showing that centralized has less load than decentralized in the case of server A, D and E. But in this also the hierarchical is showing the best result.

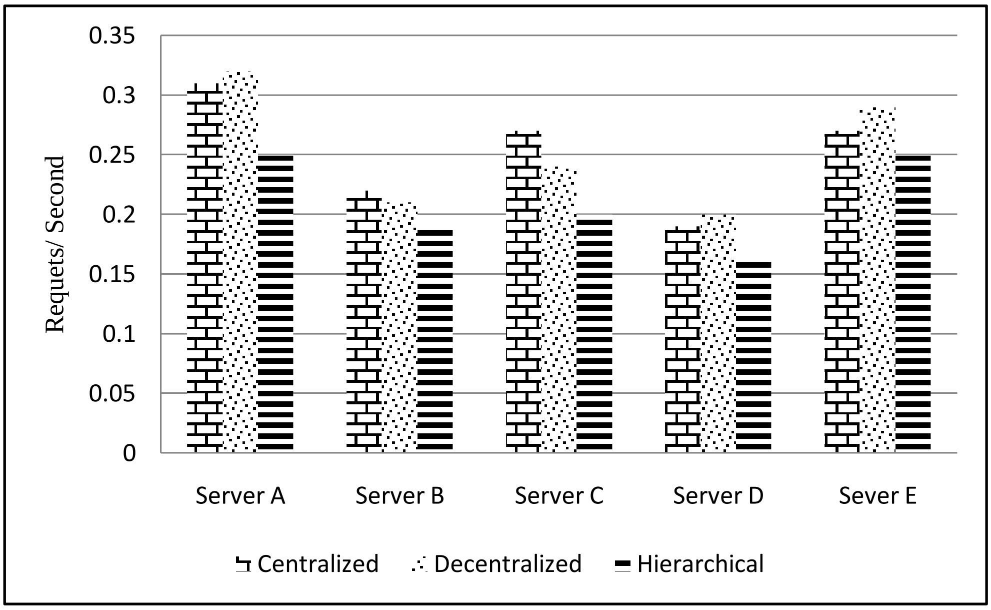

Fig. 5. Load on each server in each architecture when the number of client was 40

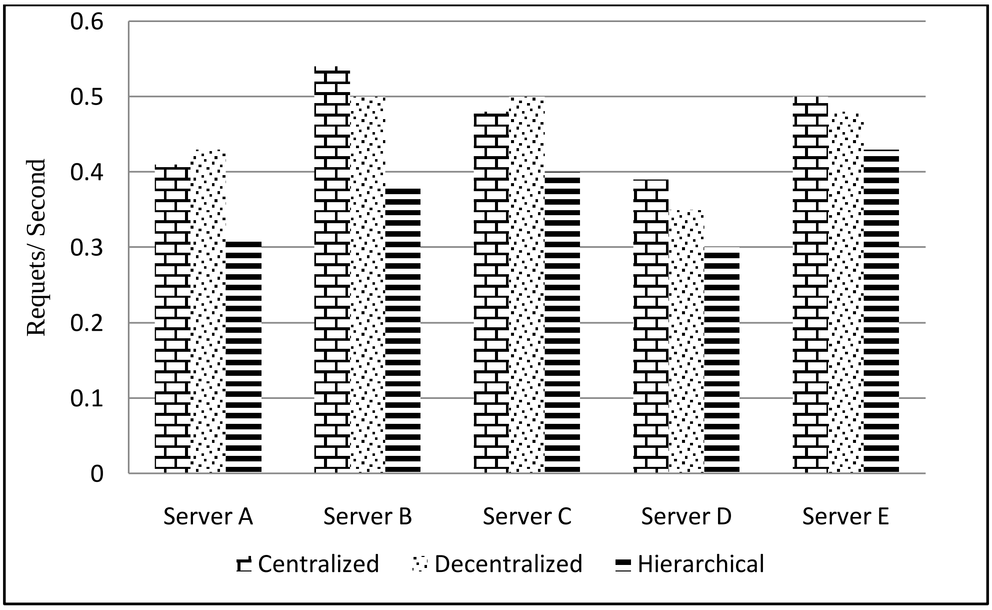

Fig. 6. Load on each server in each architecture when the number of client was 80

The average load on each server when the number of users is 80 as shown in the figure 6. In this when the results are taken, it is found that centralized and decentralized have almost same performance. But on each server the hierarchical architecture shows the better result.

After viewing all these three graphs it is analyzed that hierarchical architecture is better than the other two architectures in the case of public cloud. The result shows that the hierarchical load balancer outperformed the centralized and decentralized load balancer. This is due to the fact that hierarchical load balancer distribute the load among different servers efficiently as compared to the centralized load balancer and decentralized load balancer.

- Response Time

Response time is the time take by a system to react to a request. Low value of response time shows that load balancing is implemented very well. Fig. 7 displays the results on the bases of response time. In the graph vertical and horizontal axis represents the response time and the number of clients respectively. Fig.7 shows the response time of all the servers with different number of clients in all the three architectures. It is clear from the graph that response time increases with the increases the number of clients.

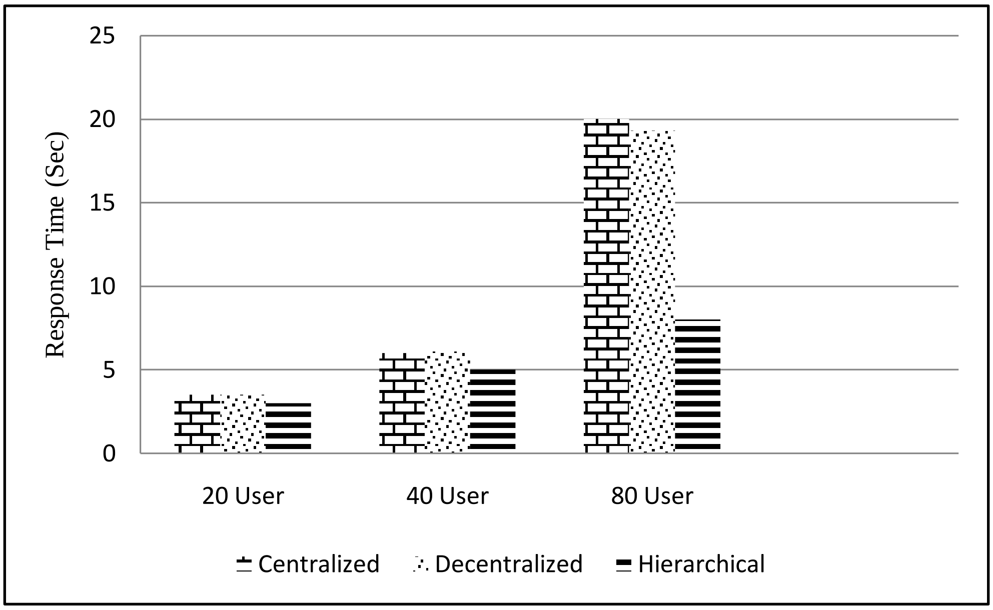

Fig. 7. Response time of each architecture with different number of user

In each case centralized and decentralized architecture almost have the same response time whereas hierarchical shows less response time. It means hierarchical load balancer handles the request well when it arrives.

The result shows that hierarchical architecture has better performance than the other two architectures. Hierarchical propose good result because load balancer balance the load across its sub tree. Groups are much smaller than the entire set of processors so reduce time and memory required for the load balancing. At each level workload information is converted in such a way that root node represent their entire sub trees. So it looks like that all the nodes belong to the root node. By using this root node can apply centralized load balancing to each sub domain and can take advantage of centralized load balancing

6. Conclusion and Future Work

Cloud Computing is a rising and most trending field of IT industry that uses the internet to provide all the things like data, infrastructure, platform as a service via internet. But still it is in developing mode so having many issues and challenges. One of the main issues out of all is Load Balancing that is required to distribute the load to get better performance.

Main aim of the paper is to compare different type of load balancing architectures in public cloud computing environment on the bases of some measures. So these are simulated via using a network simulator that is Opnet modeler at different scales. Finally it is concluded that hierarchical architecture has better performance than the other two architectures.

In fact, in the case of load balancing there is no “one size fits all”. This means, as we know the performance of load balancing depends on various factors like type of application, type of load, type of architecture and many other variables. So as the variables get changed there may be a change in result also. In future this simulation can be done under another circumstance to check that it proposes same result or not.

References

[1] Songjie, J. Yao, and C. Wu, “Cloud computing and its key techniques,” 2011, pp. 320–324.

[2] L. Wang *et al.*, “Cloud computing: a perspective study,” *New Gener. Comput.*, vol. 28, no. 2, pp. 137–146, 2010.

[3] P. Mell, T. Grance, and others, “The NIST definition of cloud computing,” 2011.

[4] R. Kaur and P. Luthra, “Load balancing in cloud computing,” in *Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing, ITC*, 2012.

[5] M. Belkhouraf, A. Kartit, H. Ouahmane, H. K. Idrissi, Z. Kartit, and M. El Marraki, “A secured load balancing architecture for cloud computing based on multiple clusters,” in *Cloud Technologies and Applications (CloudTech), 2015 International Conference on*, 2015, pp. 1–6.

[6] N. J. Kansal and I. Chana, “Cloud load balancing techniques: A step towards green computing,” *IJCSI Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 238–246, 2012.

[7] M. Rahman, S. Iqbal, and J. Gao, “Load Balancer as a Service in Cloud Computing,” presented at the 2014 IEEE 8th International Symposium on Service Oriented System Engineering, 2014, pp. 204–211.

[8] A. Vig, R. S. Kushwah, and S. S. Kushwah, “An Efficient Distributed Approach for Load Balancing in Cloud Computing,” presented at the 2015 International Conference on Computational Intelligence and Communication Networks, jabalpur, india, 2015, pp. 751–755.

[9] W. Zhu, C. Sun, and C. Shieh, “Comparing the performance differences between centralized load balancing methods,” in *Systems, Man, and Cybernetics, 1996., IEEE International Conference on*, 1996, vol. 3, pp. 1830–1835.

[10] H. mehta, P. kanungo, and M. Chandwani, “Decentralized Content Aware Load Balancing Algorithm for Distributed Computing Environments,” presented at the presented at the International Conference and Workshop on Emerging Trends in Technology (ICWET) – TCET, mumbai, india, 2011, pp. 370–375.

[11] G. Jackson, P. Keleher, and A. Sussman, “Decentralized scheduling and load balancing for parallel programs,” in *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*, 2014, pp. 324–333.

[12] M. Katyal and A. Mishra, “A comparative study of load balancing algorithms in cloud computing environment,” *Int. J. Distrib. Cloud Comput.*, vol. 1, no. 2, pp. 5–14, Dec. 2013.

[13] A. Khiyaita, H. El Bakkali, M. Zbakh, and D. El Kettani, “Load balancing cloud computing: State of art,” in *Network Security and Systems (JNS2), 2012 National Days of*, 2012, pp. 106–109.

[14] g. cybenko, “Dynamic load balancing for distributed memory multiprocessors,” *J. Parallel AndDistributedComput.*, vol. 7, no. 2, pp. 279–301, 1989.

[15] T. L. Casavant and J. G. Kuhl, “A taxonomy of scheduling in general-purpose distributed computing systems,” *IEEE Trans. Softw. Eng.*, vol. 14, no. 2, pp. 141–154, 1988.

[16] ZhilingLan, V. E. Taylor, and G. Bryan, “Dynamic load balancing for structured adaptive mesh refinement applications,” presented at the in Proceedings of the International Conference on Parallel Processing (ICPP ’01), Spain, 2001, pp. 571–579.

[17] J. C. Phillips, G. Zheng, S. Kumar, and L. V. Kalé, “NAMD: Biomolecular simulation on thousands of processors,” in *Supercomputing, ACM/IEEE 2002 Conference*, 2002, pp. 36–36.

[18] A. Bhadani and S. Chaudhary, “Performance evaluation of web servers using central load balancing policy over virtual machines on cloud,” in *Proceedings of the Third Annual ACM Bangalore Conference*, 2010, pp. 1–4.

[20] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, “Dynamically scaling applications in the cloud,” *ACM SIGCOMM Comput.Commun.Rev.*, vol. 41, no. 1, pp. 45–52, 2011.

[21] J. O. Gutierrez-Garcia and A. Ramirez-Nafarrate, “Agent-based load balancing in Cloud data centers,” *Clust. Comput.*, vol. 18, no. 3, pp. 1041–1062, Sep. 2015.

[22] M. Randles, D. Lamb, and A. Taleb-Bendiab, “A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing,” in *in Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA ’10)*, Perth, Australia, 2010, pp. 551–556.

[23] I. Ahmad and A. Ghafoor, “A semi distributed task allocation strategy for large hypercube supercomputers,” in *Periodic Hierarchical Load Balancing for Large Supercomputers*, 1990, pp. 898–907.

[24] G. Zheng, A. Bhatelé, E. Meneses, and L. V. Kalé, “Periodic Hierarchical Load Balancing for Large Supercomputers,” *Int. J. High Perform. Comput.Appl.*, vol. 25, no. 4, pp. 1–36, 2011.

[25] N. Malarvizhi and V. R. Uthariaraj, “Hierarchical load balancing scheme for computational intensive jobs in Grid computing environment,” in *Advanced Computing, 2009.ICAC 2009. First International Conference on*, 2009, pp. 97–104.

[26] R. G. Dobale and R. P. Sonar, “Review of Load Balancing for Distributed Systems in Cloud,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 5, no. 2, 2015.

[27] G. Zheng, E. Meneses, A. Bhatele, and L. V. Kale, “Hierarchical Load Balancing for Charm++ Applications on Large Supercomputers,” presented at the presented at the 39th International Conference on Parallel Processing Workshops, 2010, pp. 436–444.

[28] J. Yang, L. Ling, and H. Liu, “A Hierarchical Load Balancing Strategy Considering Communication Delay Overhead for Large Distributed Computing Systems,” *Math.Probl.Eng.*, vol. 2016, pp. 1–9, 2016.