e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:03/March-2022 Impact Factor- 6.752 www.irjmets.com

DIRECT SERVER RETURN LOAD BALANCING USING BERKELEY PACKET FILTERING ALGORITHM

Ashish Bhardwaj[*1], Avika Tyagi[*2], Darpan Pal[*3], Nagresh Kumar[*4]

[*1,2,3,4]Department Of Computer Science And Engineering, Meerut Institute Of Engineering & Technology, Meerut, India.

ABSTRACT

Cloud computing is the on-demand delivery of computer system resources from applications to storage and processing power, over the internet to offer expandable, scalable and perfectly elastic software services. It allows pay per use service capabilities with scalable features. Load balancing is a solution in cloud computing that distributes incoming network traffic to different servers connected in a network. Because without load balancing the overall throughput and performance of computing resources gets reduced. As it is often seen that the servers are either overloaded or under loaded, hence a solution was required to strengthen the system by managing both the cases. Considering the importance of load balancing techniques, this paper presents a detailed review about the efficient approach i.e. Direct Server Return Load Balancing using Berkeley Packet Filtering algorithm so that load balancing can be done using, an efficient approach in future.

Keywords: Load Balancing, Direct Server Return, Cloud Computing, Berkeley Packet Filter.

I. INTRODUCTION

Generally, Cloud Computing can be considered as internet based computing that provide storing and retrieving data from different remote servers present over an internet [6]. The expedite growth in the advancement of cloud technology, provides service to various needs of customers by providing those resources through online. It helps in reducing demand of hardware and software at client's side. The only thing you need to do is just run an application interface, that is similar to using any web browser, and the rest of the work will be carried out by Cloud network itself. Some of the popular cloud services we are using are gmail, hotmail or yahoo etc. In all these applications, data can be made available to the consumer located in any geographical location. While implementing this distributed environment, there arises a need to distribute the dynamic workload equally among the servers connected in a network that can also monitors the overall performance of system at the same time. This need came into existence as Load Balancing. This ensure proper utilization of different resources which in turn increases user satisfaction by solving one of the biggest problem of fair allocation of computing resources. Load balancing split up the upcoming traffic on the available nodes and results in reduced latency. By evenly distributing the workload networks and resources exhibit maximum throughput with minimum response time to the client. Thus, load balancing became an important aspect at Cloud Service Provider side that can help in improving the performance and efficiency of the computing and network resources along with guaranteed Quality of Service at Service Level Agreement agreed between consumer and provider.

II. LITERATURE REVIEW

Over the time, different researchers proposed different algorithms related to load balancing in cloud computing. Alam et al. [1] made a heuristic solution using Cloudsim simulator, the algorithm includes task grouping with prioritization of nodes. Toosi and Buyya [2] proposed a fuzzy logic-based solution using Cloudsim and Google Cluster, they emphasize to reduce energy and cost of system. Mahalingam et al. [3] in 2015 concluded that balancing of load is the main aspect to be considered in cloud computing he proposed the weight based optimized algorithm that uniformly distributes of incoming jobs over various virtual machines. Chien et al. [4] in 2016 proposed a solution that concerns with improvement of system performance, processing time and response time by evaluating the task completion time of different servers. Mondal and Choudhury [5] in 2015, proposed a strategy based load balancing algorithm using Cloud-Analyst simulator that solves the problem of dynamic workload across various resources as per requirement of client. In this paper we have discussed Proposed technique in section 3, Process Description in section 4, Result in section 5 and conclusion in section 6 of the Direct Server Return Approach.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:03/March-2022 Impact Factor- 6.752 www.irjmets.com

III. PROPOSED TECHNIQUE

Load balancing is done in order to improve application responsiveness and ensures that no single server bears too much demand thus, it is able to avoid the condition where nodes are either heavily loaded or under loaded in a network, hereby it improves overall performance of the system as well as increase availability of applications and websites for users. The objective of proposed technique is to allow the target server to respond to clients directly instead of going back to load balancer using Berkeley Packet Filter which in result reduces the unnecessary traffic in the network. In addition, proposed model also performs health checks on each server. The technique used in proposed model would be Direct Server Return Load Balancing using Berkeley Packet Filtering Algorithm. Direct Server Return (DSR) approach is designed to distribute traffic load across clustered processors, storage and network infrastructure in order to increase network reliability, efficiency and performance. It is a solution of lopsided network load distribution in system. The solution provides a way that incoming requests will be redirected to the source instead of load balancer meaning different network path will be used by request and response traffic. The advantage of using different network paths helps in avoiding extra hops and reducing the latency by which response time between the client and the service improves and also some extra load from load balancer is removed.

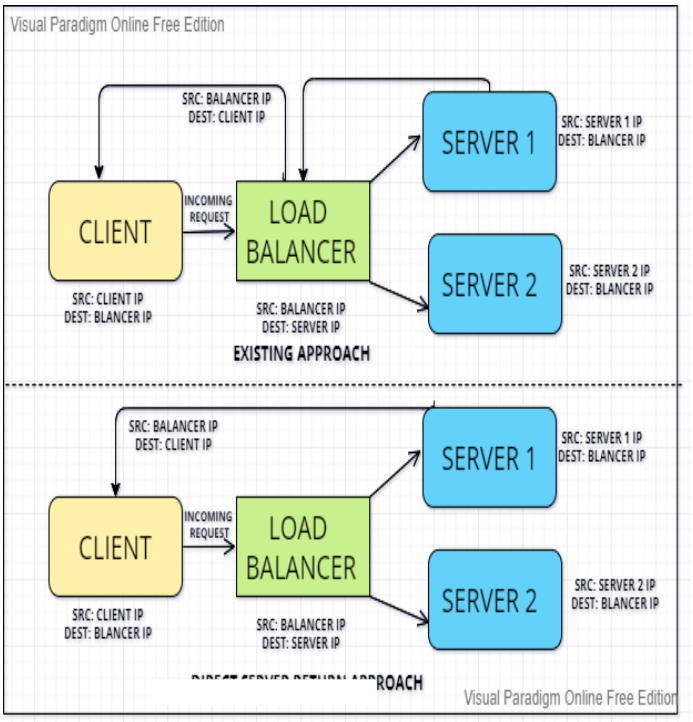

Fig 1: Existing Approach V/S DSR Approach

IV. PROCESS DESCRIPTION

In existing method, [7] at load balancer an IP address is assigned to all incoming requests then, these requests will be distributed to different servers connected with load balancer in a network. Different algorithms work at load balancer which will select appropriate server among the network. [8]Once request is reached to target server, it will process the request and send required data onto the client by the load balancer. Now, because all servers are sending the data to the load balancer, there is a lot of traffic of responses due to which rises network performance consequently suffers. Hence a solution is required which can modify the network traffic flow by allowing the server to respond directly to the client, relieving the load balancer to handle the heavy traffic load, this approach is known as Direct Server Return DSR. At load balancer, the request for changing the destination IP address will be done using Berkeley Packet Filtering method, which will give an access to all binary streams so that the IP address can be changed and send it back to the network interface. In addition to this, there will be individual threads for each node that will be doing health checks for each node as well as the requests which is redirecting to client. Health checks is done by load balancer to monitor activeness of target

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:03/March-2022 Impact Factor- 6.752 www.irjmets.com

servers. It enables the load balancer to detect an unhealthy nodes and distribute request to active ones. Step by step description of process is as follows:

A. Incoming Request: Whenever a load balancer receives incoming requests from clients it simply routes requests to one of its registered servers. Here we are moving request to balancer using one thread. The function **packetSenderListener()** is defined which will simply send packet to the destination.

B. Generic Routing Encapsulation: Once request is received by server one can initiate request for changing the IP address. Using generic Routing Encapsulation one packet is being sent to the destination. Here a function is defined namely **handleBackendIngressTraffic()** that encapsulates the IP packet delivered to backend.

C. Distribution of load: Many algorithms exist for distribution of load requests among the servers like Round Robin, weighted Round Robin, IP Hash, Least connection, etc. Here, **initBackend () function** initiates backend node to handle the load.

D. Berkley Packet Filtering Algorithm: The function **bpf()** is making filters for load balancer here BPF will return packets that pass the filter only. This will help in eliminating moving unwanted packets from the OS kernel to the process, hence improving overall performance.

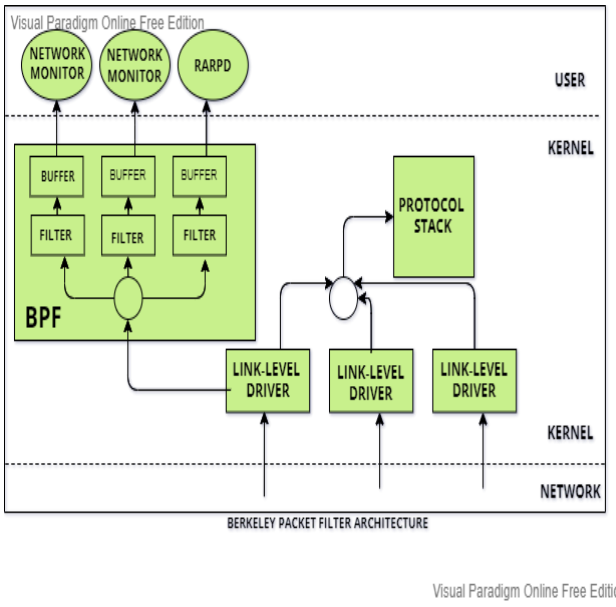

Fig 2: Architecture of Berkeley Packet Filtering Algorithm

E. Health Checks: To check status of registered severs whether they are active or overloaded, load balancer node performs health checks in the system, individual threads for each node will be running for performing health checks for each node. The function **reportCardGenerator()** is monitoring the health here.

V. RESULT

We have taken two metrics namely CPU utilisation and Throughput in order to measure performance of proposed system. Then we compared our DSR approach with existing algorithms namely Round Robin and Weighted algorithms. Figure 3 depicts the CPU utilisation and Figure 5 depicts Throughput table when different loads are applied to a system. Figure 4 and Figure 6 (Red colour shows DSR, Blue colour shows Riund Robin and green colour shows weighted algorithm) shows graphical representation of overall result.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:03/March-2022 Impact Factor- 6.752 www.irjmets.com

CPU (%)			Load (in thousands)
DSR	Round Robin	Weighted Algorithm	
3	1	0	0
5	4	1	5
13	8	6	10
18	10	7	15
23	14	7	20
28	19	9	25
33	25	15	30
38	27	16	35
43	29	18	40
48	35	22	45
53	37	29	50
58	41	30	55
63	48	32	60
66	51	33	65
67	50	37	70
76	56	43	75
81	58	45	80
80	61	51	85
91	63	55	90
98	68	57	95
99	69	59	100

Fig 3: Resource Utilisation Table

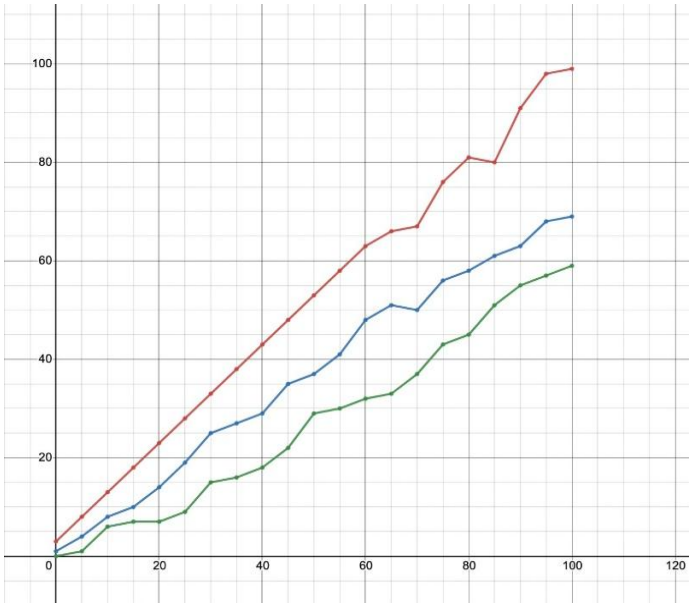

Fig 4: Resource Utilisation Graph

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:03/March-2022 Impact Factor- 6.752 www.irjmets.com

Through put (%)			Load (in thousands)
DSR	Round Robin	Weighted Algorithm	
20	25	18	0
22	26	20	5
26	28	25	10
31	30	27	15
35	33	27	20
39	34	31	25
43	35	31	30
46	37	32	35
48	39	31	40
48	40	30	45
50	41	30	50
52	42	31	55
57	40	33	60
56	41	34	65
57	43	35	70
61	45	39	75
62	43	40	80
65	42	41	85
68	41	42	90
68	45	43	95
70	44	45	100

Fig 5: Throughput Table

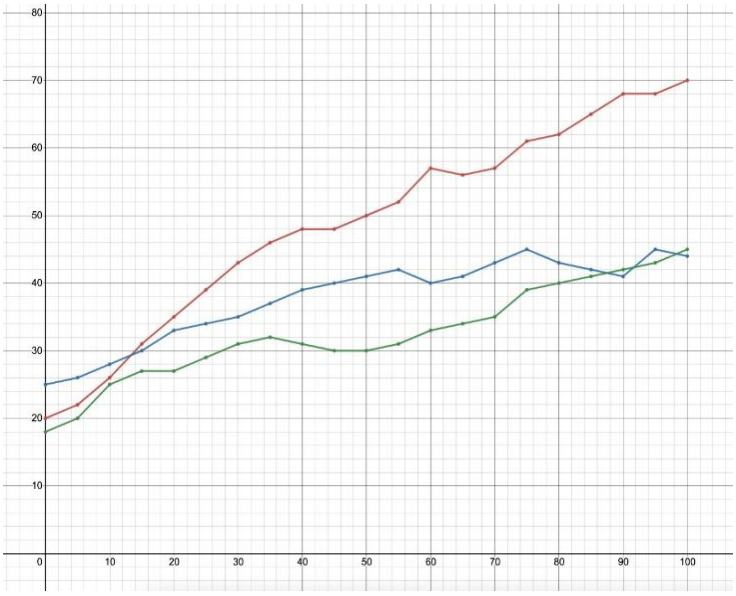

Fig 6: Throughput Graph

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:03/March-2022 Impact Factor- 6.752 www.irjmets.com

VI. CONCLUSION

In this paper, it is very evident that Direct Server Return approach using Berkeley Packet Filtering algorithm is far better than existing algorithms. As response time reduces the overall performance of system improves.

VII. REFRENCES

[1] M.I. Alam, M. Pandey, S.S. Rautar ay A proposal of resource allocation management for cloud computing Int. J. Cloud Comput. Serv. Sci., 3 (2) (2014), pp. 79-86.

[2] Toosi, A.N., Buyya, R., 2015, December. 'A Fuzzy Logic-based Controller for Cost and Energy Efficient Load Balancing in GeoDistributed Data Centers', In: 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), pp. 186–194.

[3] Mahalingam, Nandhalakshmi Nithya, "Efficient Load Balancing in Cloud Innovative Research in Computer and Communication Engineering, 2015, vol.3, 5409 – 5415.

[4] Chien, N.K., Son, N.H., Loc, H.D., 2016, January, 'Load balancing algorithm based on estimating finish time of services in cloud computing', In: 2016 18th IEEE International Conference on Advanced Communication Technology (ICACT), pp. 228–233.

[5] B. Mondal, A. Choudhury simulated Annealing (SA) based load balancing strategy for cloud computing (IJCSIT) Int. J.Comput. Sci. Inf. Technol., 6 (4) (2015), pp. 3307-3312

[6] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis,and A. Vakali, Cloud computing:Distributed internet computing for IT and scientific research, Internet Computing, vol.13, no.5, pp.10- 13, Sept.-Oct. 2009.

[7] P. Mell and T. Grance, The NIST definition of cloud computing, http://csrc.nist.gov/publications/ nistpubs/800-145/SP800- 145.pdf, 2012.techtarget.com/definition/public-cloud, 2012.

[8] Di, S., Kondo, D., Walfredo, C.: Host load prediction in a Google compute cloud with a Bayesian model. Proceedings in the international conference for high performance computing, networking, storage and analysis, SC, pp. 1–11 (2012).

[9] Computing Using Weighted Throttled Algorithm", International.