Variants of Software Defined Network (SDN) Based Load Balancing in Cloud Computing: A Quick Review

Jitendra Bhatia, Ruchi Mehta(✉), and Madhuri Bhavsar

Nirma University, Ahmedabad 382481, Gujarat, India
mehtaruchi1312@yahoo.com

Abstract. Nowadays users of cloud are increasing rapidly hence handling of and allocation of that resources are the main challenge. Load balancing strategy refers to scatter the dynamic workload over the various node to guarantee that no single node is over-burden. There are few limitation of conventional load balancers in terms of flexibility and adaptability. To overcome this pitfalls, the usage of Software Defined Network based approach in load balancing proves to be advantages. Software Defined Networking is a developing innovation which helps to quickly strategies in familiarizing the administrations with the business segment without relying upon the seller based setup of the gadgets. SDN helps to set control decision for algorithm that apply for system which increases the performance of that algorithm, reduces response time, increase scalability, flexibility and results in reduction of the energy consumption of system. In this paper, we examined the feasibility of SDN-based load balancing and discussed variants of the SDN-based load balancing using various controllers.

Keywords: Cloud computing · Software defined network
Load balancing · OpenFlow · Controller

1 Introduction

1.1 Cloud Computing

The quick improvement of the Internet that has encouraged a large number of new innovations including cloud computing. The cloud computing rapidly emerged as a virtualization technology that aims to provide scalable, transparent network to end users [2]. The cloud can give facility to use on-demand computing and storage application to end users. The end user cannot have knowledge of where that service is from and how they are delivered to them. Cloud Computing has mainly three components that are the client computer, data center, and distributed servers. Client computers means devices that users can communicate with other cloud components such as data center and distributed server via the Internet. There are three types of client that are thin, thick and mobile client.

Z. Patel and S. Gupta (Eds.): ICFITT 2017, LNICST 220, pp. 164–173, 2018.
https://doi.org/10.1007/978-3-319-73712-6_17

A thin client is a most popular client in the cloud whereas distributed servers are placed at different location. In that central server are there which can monitor traffic, client demands and ensures that all the process runs smoothly or not and the data center is a collection of servers where various application are deployed which can be accessed via the Internet.

As the usage of cloud is increasing, it has a huge effect on the cloud data center because of the large number of request that arrives at the same time, but sometimes the data center cannot allocate the resources at this peak time which results in to the situation for adding the resources to fulfill the end user request. On the off chance that the load is not adjusted at specific path, then including new resources leads to wastage of assets [9]. To accomplish the negligible reaction time and to decrease the utilization of registering assets required that adjusting the heap among all the accessible assets decently.

1.2 Software Defined Network

SDN is a new methodology in networking that allows the administrator of the network to manage the network abstraction through the lower level functionality [22]. The concept of SDN is open and reprogrammable. This is an architecture that is not only controls network devices but also controls an entire network. The main goal of this architecture is to allow network engineers and administrator to respond rapidly to evolving business needs [10]. It control the entire network centrally so that there is no need for touch individual devices for any change in the network. Due to changes of the functions and performance, the traditional networks have lots of drawbacks. For achieving better services we can include so many solutions in the network device. This will make our system large, fat and complex which results in complexity to achieve the better network performance [10]. To overcome this limitation, Software Define network was proposed in the year of 2003 [18]. There are two parts of SDN infrastructure that are control plane and data plane [8]. The control plane is responsible for controlling the data transmitted over the network. The logic of the controlling can be implemented in the server as a software component. And data plane is responsible for forwarding data. It is available on the network devices like switches, router. This SDN is an emerging architecture that is dynamic, cost-effective and manageable. Control plane consists various controller and applications and data plane/forwarding plane consists of various networking devices. OpenFlow protocol is suitable standard protocol for SDN-based system implementation [18].

Software defined network architecture is shown in Fig. 1. A layered architecture that has application layer, the control layer, and infrastructure layer. The application layer consists various applications like firewall, load balancer etc. and network services that interact with the control layer with the help of northbound APIs. The control layer consists of centralizing control plain used for communication with below layer of sdn known as infrastructure layer using OpenFlow protocol which uses southbound API while the infrastructure layer consists both network and virtual devices that implement OpenFlow protocol for implementing the traffic forwarding rule [8].

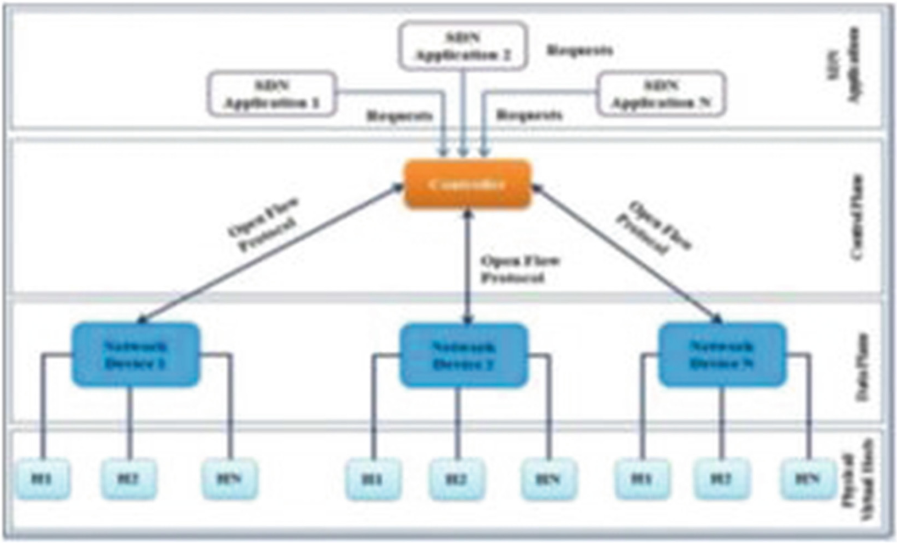

Fig. 1. SDN architecture [8]

1.3 OpenFlow

The OpenFlow standard is utilized as a part of SDN for correspondence between applications. There is likewise an OpenFlow empowered controller characterized in the OpenFlow switch particular. It is supervised by Open Networking Foundation. The OpenFlow convention permits consistency, direct control of the foundation, consequently evacuating the requirement for complex system administration [10]. It includes adaptability and substantial flexibility from the exclusive conventions of a solitary equipment merchant. The design of OpenFlow comprises of three parts: 1. controller, 2. secure channel and 3. OpenFlow empowered switch [10]. Thus the Switches utilize a stream table to forward the bundle to the goal. Stream table contains the rundown of stream sections. Switches utilize the conventions that are characterized in the stream table for sending the parcels. A secure channel is utilized for secure correspondence amongst switch and the controller. The Controller is a product program that is utilized to include, adjust/change and erase the stream table passage of the switch utilizing this convention [10] (Fig. 2).

1.4 Difference Between Traditional Network and SDN Network

The SDN has so many advantages with compared to traditional networking are as followed:

- Migration VM become easier.
- SDN only requires one centralized control plane which offsets the cost of the forwarding plane.

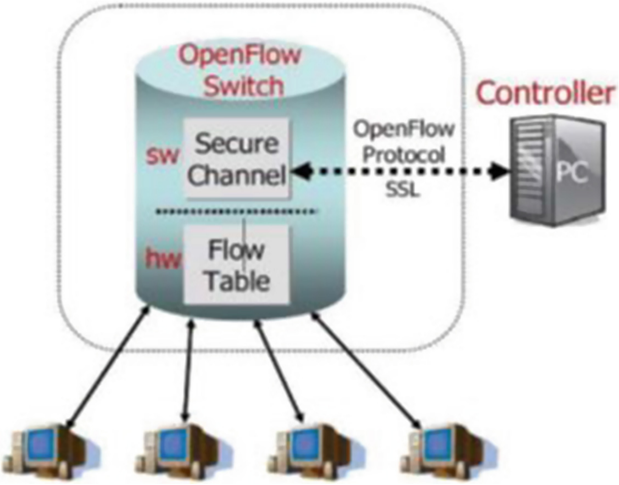

Fig. 2. OpenFlow architecture [8,10]

- In SDN QOS is provided in more efficient way.
- SDN is directly programmable because Whole network is directly programmable because forwarding functions and control functions are logically decoupled from each other which enables the network pragmatically configured by automation tool including OpenStack, Chef, and Puppet.
- SDN does not require a huge amount of resources so that investment of resources can be reduced and also some of the SDN products are open source.

Table 1. Difference between traditional network and SDN network

Traditional networking	SDN
Static and inflexible networks and are not useful for new business ventures	Programmable networks during deployment time as well as at later stage based on the change in the requirements and helps new business ventures through flexibility, agility, and virtualization
Hardware appliances	Configured using open software
Distributed control plane	Logically centralized control plane
Work using protocols	APIs to configure as per need
Policy to treat an incoming data packet is written into its firmware	SDN facilitates admins with granular control over the way switches handle data, giving them the ability to automatically prioritize or block certain types of packets Which in turn, allows for better efficiency without the need to invest in expensive, application-specific network switches

But they also have some challenging issues like Quality of service, security, Load balancing, scalability [8].

1.5 Load Balancing

Load balancing is used to distribute load and improve the performance, availability of resources and scalability of the system, hence we can achieve the minimum response time of the application and increase the throughput [8].

The load balancing algorithm in cloud typically divided into two sections that are static load balancing algorithm and dynamic load balancing algorithm. The static algorithm can most suitable in a homogeneous environment [2]. But this type of algorithm is not suitable where attributes are dynamically changing during the execution time. Whereas dynamic algorithm can be used in the environment when the attributes are changed continuously and also give good result in the heterogeneous and dynamic environment [2].

In the static mechanism, An algorithm like Round robin and central load balancing decision model (CLBDM) [2] are considered. In the Round robin load balancing algorithm, it provides resources to the task as FCFS (First come First serve) basis. The central load balancing decision model works same as round robin load balancing algorithm, but in addition to this, it calculates the duration of the connection of the client request and server response by measuring the overall execution time of the task on the given resources.

In the dynamic load balancing algorithm includes algorithms like Least-connection algorithm, Response time algorithm, and Predictive algorithm [22]. Least-connection algorithm can detect the number of connections that associated between the client and the server, in a particular time interval. And if a new access request arrives then they forward the request to the least connection server [22] and that server processes the request and reply back to the requester. In the Predictive algorithm, the server load can be predicted for the next period of particular requests. In the response time algorithm, the balancer can estimate the load of the each and every server using sending a ping request to the server [22].

These are the traditional network that has some limitations like complex software and hardware used in the system that can increase the operation cost of the organization and their re-usability of code and their applicability to the particular application is different hence architecture is poor.

Table 2. Variants of SDN controllers

Controller	OpenSource	Language	Multi-threaded	GUI	Invented By
Nox [6]	Yes	C++/Python	No	No	Nicira Networks
POX [13]	Yes	Python	–	No	Nicira Networks
Beacon [5]	Yes	Java	Yes	Yes	Standford University
Floodlight [12]	Yes	Java	Yes	Yes	Big Switch Networks
OpenDayLight [7]	Yes	Java	Yes	Yes	Multiple Contributors
RYU [14]	Yes	Python	–	No	NTT OSRG and VA Linux

Mainly in this paper, we surveyed the variants of the current SDN-based load balancer and also discussed the various parameters of that variants and how they can be used to achieve effective load balancing in cloud computing using software define networks.

2 Variants of SDN Based Load Balancer

2.1 Heuristic Based Load Balancer

The main goal of this load balancer is to minimize both server and network load. This method defines its own objective function that discovers best path and best server in the fastest manner. The data plane, controller, application layer is 3 layers of this framework. The data here global view of the perspective network. Hence dynamic path selection can be accomplished by minimal response time than the hop based updates in routing [9].

This method can be tested in the Java environment [9]. Using the objective function complete the request in the smallest span of time. The dynamic path can be reduced, by evaluating hop by hop method of the network. Using objective function, congestion and delay via existing algorithm because can be reduced as it that selects the least loaded path [9].

2.2 SDN Based Traffic Engineering Based Load Balancer

Traffic Engineering (TE) means optimization of performance of the network by dissecting, anticipating and managing the conduct of the information that is transmitted over the network [21]. Software defined network based TE that comprises of ideal design organization and traffic load balancing. Here author describes main component of Traffic Engineering manager for this method [21] that focuses on ideal topology arrangement and traffic load adjusting. The optimal topology consumption is power utilization of DCN is always higher than what amount required because the movement interest in DCN progressively changes from time to time [21]. Traffic load balancing is to minimize the congestion of all the possible connections by separating the traffic of DCN design or subset topology discovery by ideal topology composition algorithm using dynamic traffic load balancing [21]. Here both the algorithm can be describe using heuristic and linear way. This method testing prospective uses the mininet simulator and the virtual instance is OpenVSwitch. For notification Floodlight [12] which is Java API is used. This system reduces the 41% power consumption and 60% lower maximum link utilization compared to existing static routing scheme [21].

2.3 OpenFlow Protocol Based Load Balancer

The main objective of this load balancer changes the manual and costly hardware of clusters to the OpenFlow based controller using the local network infrastructure. The main key idea of this proposed load balancer is to replace the expensive and statically defined network component and cluster through openflow controller. Here two algorithms such as OpenFlow based Round-Robin and OpenFlow based Least connections algorithm has been proposed [22]. Above two techniques are tested on the mininet [3] and Floodlight [12] Java API. In first approach, response time is very unstable because server cluster varies greatly. Whereas the second approach takes lesser time and give the better performance so that this method always help to forward load to least connection server [22].

2.4 SDN Based Design of Load Balancer Middlebox for Data Center

In this load balancer, there are number of SDN controllers and OpenFlow switches inside the middle box network and that are basically based on clos network [4] framework architecture. Here, Users can be arranged in nonblocking Clos network to achieve better efficiency and resource utilization. There are two methods describes here and that are switches inside the middle box and server inside the middle box [16]. Delay and packet loss can occur because traffic passes inside the middle box through a switch. Port rate, traffic load, queue length etc. are information of the switch that can be collected by a controller. With the help of these attributes the path within the middle box can be changed [16] and load will be equally distributes. This type of SDN load balancer has been tested in the Matlab and it improves the utilization and reduces the latency.

2.5 Data Flow Network Load Balancer in Eucalyptus

Here authors used the Eucalyptus cloud system architecture for proposed LBVMD [19] i.e. load balancer VM deployment mechanism. This system basically consists three components that are Eucalyptus, agent, and the openflow-enabled switch. In that agent is important part of this system because it monitors the network and send information to Cloud controller of Eucalyptus for selecting appropriate Node controller to create new VM [19]. OpenDaylight [7] is used as a controller for the system [19]. In this system, the VM made by the system obliged planning component gives preferred execution over existing instruments. Openflow and the Eucalyptus distributed computing setup are utilized as a part of the testbed [19].

2.6 SDN Based Dynamic Load Balancer

Dynamic load balancing method of cloud-center is based on SDN (SDN-LB). SDN-LB includes four main modules: traffic detection module which is responsible for dynamic traffic monitoring and statistics; load calculation module which aims to estimate the load distribution of cloud environment; dynamic load scheduling module which proposes a hybrid load balance algorithm to realize high performance load balance for Cloud center; flow management module which is responsible for deployment load balance strategy based on a hybrid load balance algorithm [20]. This SDN-based dynamic load balancing algorithm inside

POX [13] controller is used which is written in Python language [20]. It yields a higher throughput and better efficiency compared to existing dynamic load balancing methods [20].

2.7 Extended Health Monitoring for Openflow Network (EHLBOF)

In this paper [17], mainly focuses for checking the health of the server i.e. the status of physical resources of server and applications that are running on the server. The EHLBOF [17] probes periodically and gets the status of servers that are present in the network by using Simple Network Management Protocol (SNMP) request and process the response message from the server and update its status in the server. And suppose any problem arises then discard any future request for that particular server [17]. This method has been tested on the mininet [3] and pox [13] and the result shows that the throughput increases double than the round robin method because this method can dynamically find the status of each server and update the same dynamically [17].

2.8 Path Load Balancing

This paper [11] basically describes path load balancing. This method is divided into three parts that are data collection, evaluation model [11] and flow table installation. This method selects the path dynamically based upon the traffic information of the particular node. This traffic information help to detect fault in link or node based on which they select another path [11]. This method has been tested in mininet [3] and POX [13] and the result when compared with shortest path algorithm in terms of reliable, efficient and effective [11] also it give the assurance about the quality of packet.

2.9 Single Flow Table and Group Flow Table Combination

This paper basically focuses on flow table rules and the algorithms related to single flow table and group flow table and combination of both [15]. Group flow table has the traffic and number of packets. For single flow table is search based on the information of the health of the backend server and in case of rule is utilized directly else it decides whether that is need for modification the load balancing that is matched it offer to monitor in case of server down, maintenance of Group flow table [15]. This method was implemented on the mininet [3] and OpenDayLight [7] and analyzing the results show that the life cycle of single flow table is short and group flow table is very long. Single flow table analyzes traffic of each and every client given its information for changing in the Group Flow table [15].

2.10 End Host Load Balancing

This paper describes [1] mainly two load balancing scheme based on the controller and switch [1]. Controller based load balancer is based upon the Round

Robin in that decision is based on individual TCP session. Here controller can select an interface of the available N interface. In switch-based load balancing, a hash function is used. The Hash function selects the random switch and perform equal load balancing and thereby selecting one interface for whole TCP session [1]. This method has been tested on the GNS3 and RYU [14]. The result was analyzed that controller based load balancing can run in the time that nearest to an optimal value of running time [1].

3 Conclusion

Cloud computing is new emerging era of computing, it uses the resources like processing, storage and network based application. For this type of application, performing the task like customizing systems administration and virtualization are the major issues which can be solved by using SDN. SDN utilizes the system assets flexibly and fulfills the client application without any limitations. We surveyed the SDN based load balancing mechanism in the cloud environment and also discussed various types of variants that helps the load balancing mechanism efficiently with compare to the traditional mechanisms. We concluded from variants that we can achieve minimum response time, higher throughput than conventional method of load balancing and also results in reduction of the power consumption using traffic engineering mechanism. Using OpenFlow, it forwards the load to least connection sever which reduces latency and improves the utilization of data center using close network based middle box design. It improves server health problem and also finds an optimal value of running time using SDN controller based load balancer.

References

1. Al-Najjar, A., Layeghy, S., Portmann, M.: Pushing SDN to the end-host, network load balancing using openflow, pp. 1–6 (2016)
2. Al Nuaimi, K., Mohamed, N., Al Nuaimi, M., Al-Jaroodi, J.: A survey of load balancing in cloud computing: challenges and algorithms, pp. 137–142 (2012)
3. Antonenko, V., Smelyanskiy, R.: Global network modelling based on mininet approach, pp. 145–146 (2013)
4. Clos, C.: A study of non-blocking switching networks. Bell Syst. Tech. J. **32**(2), 406–424 (1953)
5. Erickson, D.: The beacon openflow controller, pp. 13–18 (2013)
6. Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N., Shenker, S.: Nox: towards an operating system for networks. ACM SIGCOMM Comput. Commun. Rev. **38**(3), 105–110 (2008)
7. https://www.opendaylight.org/. The opendaylight platform
8. Meng, K.C., Govindarajan, K., Ong, H.: A literature review on software-defined networking (SDN) research topics, challenges and solutions. In: 2013 Fifth International Conference on Advanced Computing (ICoAC) (2013)
9. Koushika, A., Selvi, S.: Load valancing using software defined networking in cloud environment. In: 2014 International Conference on Recent Trends in Information Technology (2014)

10. Lara, A., Kolasani, A., Ramamurthy, B.: Network innovation using openflow: a survey. IEEE Commun. Surv. Tutor. **16**(1), 493–512 (2014)
11. Li, J., Chang, X., Ren, Y., Zhang, Z., Wang, G.: An effective path load balancing mechanism based on SDN, pp. 527–533 (2014)
12. Big Switch Network. http://www.projectfloodlight.org
13. Nicra. http://www.noxrepo.org/pox/about-pox/
14. NTT Laboratories OSRG Group NTT. http://osrg.github.com/ryu/
15. Qilin, M., Shen, W.: A load balancing method based on SDN, pp. 18–21 (2015)
16. Zhao, J., Yang, Y., Shi, L., Tu, R., Wang, X., Wolf, T.: Design of a load-balancing middlebox based on SDN for data centers. In:2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (2015)
17. Saifullah, M.A., Maluk Mohamed, M.A.: Open flow-based server load balancing using improved server health reports, pp. 649–651 (2016)
18. Network Virtualization. https://www.sdxcentral.com/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/
19. Chen, J., Chou, F., Hsieh, W., Hsieh, W., Lee, Y.: Load balancing virtual machines deployment mechanism in sdn open cloud platform. In: 2015 17th International Conference on Advanced Communication Technology (ICACT) (2015)
20. Qian, H., Yong, W., Xiaoling, T., Yuwen, K.: A dynamic load balancing method of cloud-center based on SDN. China Commun. **13**, 130–137 (2016)
21. Li, J., Hyun, J., Yoo, J., Han, Y., Seo, S., Hong, J.: Software defined networking-based traffic engineering for data center networks. In: The 16th Asia-Pacific Network Operations and Management Symposium (2014)
22. Zhang, H., Guo, X.: SDN-based load balancing strategy for server cluster. In: 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems (2014)