ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

SDN-based Server Load Balancing Using Improved Health Monitoring and Admission Control

[1]Mohd A Saifullah and [2]M A Maluk Mohamed

[1]Anna University, M.A.M. College of Engineering, Information Technology, M.A.Saifullah, Tiruchirappalli. India.
[2]Anna University, M.A.M. College of Engineering, Information Technology, M.A.Maluk Mohamed, Tiruchirappalli. India.

ARTICLE INFO

Article history:
Received 23 June 2015
Accepted 25 August 2015
Available online 2 September 2015

Keywords:
Software Defined Networks, Cluster Computing, Server Load Balancing, Admission Control

ABSTRACT

The exponential growth of WWW users and services day by day resulting in high internet traffic and heavy load on the web servers. This is in turn increasing the service time of web requests and degrading the performance of the web server. Software-Defined Networking (SDN) has emerged as a promising solution to solve the challenge as it provides the flexibility for each user by giving the programmatic control of each flow. Server load balancing policy plays a critical role to achieve scalability and better quality of service offered by cluster of web servers. In this paper, we present a new policy called Server load balancing with improved server Health monitoring and Admission control (SHASDN) using SDN. SHASDN policy is capable of offering different client priorities, such as premium customers and default customers and honours customer SLAs. We show that using SHASDN strategy, web servers are able to maintain Service Level Agreements (SLA) without the need of a priori over-dimensioning of server resources. This is achieved by taking the real perspective of the service requests using the measurement of flow counters at that time and selectively drop some requests from the default clients if the default customers traffic is high. We analyzed and compared the experimental results of SHASDN strategy with the results of the very popular load balancing algorithm, Weighted Round Robin (WRR) and also Extended Health Monitoring for Load Balancing in OpenFlow based Networks (EHLBOF). Throughput and response time are the metrics measured in these experiments for the three different load balancing strategies. We show that even though the SHASDN strategy takes a little more server processing resources than WRR, it is capable to render assurances unlike WRR and EHLBOF.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: M A Saifullah and M A Maluk Mohamed, SDN-based Server Load Balancing Using Improved Health Monitoring and Admission Control, ***Aust. J. Basic & Appl. Sci.,*** *9(27): 132-138, 2015*

INTRODUCTION

The high demand on web based services is increasing the load that web servers need to support. But the customers expect high availability of service and better response times. Hence, service providers need to offer the services with very high performance to keep the existing clients happy and attract new clients (Cardellini *et al*., 2001; Andreolini. M., *et al*., 2004; Saeed Sharifian *et al*., 2010). Server load balancing solves these challenges as it makes several web servers take part in the same web service and share the load as the service capacity of a single web server is limited. Thus, Server load balancing gives critical benefits like availability, scalability, security and manageability of Web services. One of the most popular types of web server load balancing is cluster based web servers (Cardellini *et al*., 2002; Katja Gilly *et al*., 2011; Jayabal. R., *et al*., 2014; Saifullah M. A. and M. A. Maluk Mohamed, 2015b). In this option, the content is replicated on multiple servers of the cluster to achieve important benefits. But this requires robust load balancing strategy to achieve the benefits. A server load balancing strategy consists of distributing or assigning the tasks of a parallel application across the available servers in a cluster. The best load balancing strategy avoids the situation where some servers are idle while others are busy and have multiple jobs queued up (Andrews G. R. *et al*., 1982; Aron, M *et al*., 2000; Carter R. L *et al*., 1995; Saifullah M. A. and M. A. Maluk Mohamed, 2014a).

Present SDN-based web cluster solutions have to overcome the challenge of over load conditions and flash crowds or sudden high demand that are common in the context of current service requests (Long H. *et al*., 2013; Saifulla M. A. *et al*., 2015a; Wenbo Chen *et al*., 2014). In this proposal we are addressing the over load condition or flash crowds as this is bigger challenge compared normal condition

Corresponding Author: M A Saifullah, M A M College of Engineering, Computer Science & Information Technology, Siruganur, Tiruchirapalli, India
Ph: +91-40-23004339 Email: ssg_saif@mamce.org

of service requests (Casalicchio E *et al*., 2002; Boone B. *et al*., 2010; Saifullah M. A. and M. A. Maluk Mohamed, 2014b). Our previous work (Saifullah M. A. and M. A. Maluk Mohamed, 2015c) gives a better load balancing strategy called EHLBOF by using extended Health Monitoring of servers for OpenFlow based Networks but this solution addresses the challenge of cluster from servers side. The current proposal adds the solution for the over load conditions and sudden high demand from the point of clients by using classification of customers and admission control on top of EHLBOF strategy.

EHLBOF strategy solves the problem of sudden high demand by helping the cluster to serve maximum number of requests in overload conditions and maintain the response times at an acceptable level. So admission control is an important solution to solve the problem of flash crowds. In order to fulfill these challenges, Server load balancing using enhanced server Health monitoring and Admission control (SHASDN) proposed in this paper can do load balancing of requests and judiciously discard requests from default customers to reduce load on the web servers thereby to honour SLA to premium clients. Proposed SHASDN algorithm provides best effort to default customers.

SHASDN load balancing strategy can be used in broad variety of functional areas. To cite an instance, the delivery of multimedia content can be improved by using SHASDN load balancing strategy for the sake of meeting premium guarantees. Other example lies in e-commerce in the context of a call center negotiating with multiple credit checkers to get the payment validation. As SHASDN already recognizes premium customers, it gives higher priority with better execution time.

The organization of the paper is as follows: in Section 2, we introduce the proposed architecture of SDN-based web clusters and also the functional description of OpenFlow based controller. In Section 3, we present the proposed load balancing strategy we designed and implemented. In Section 4, we describe the experimental setup used to evaluate the performance of our proposal. Section 5 discusses the related work. Section 6 gives conclusions and future work.

2. The Proposed Architecture of SDN-based web cluster:

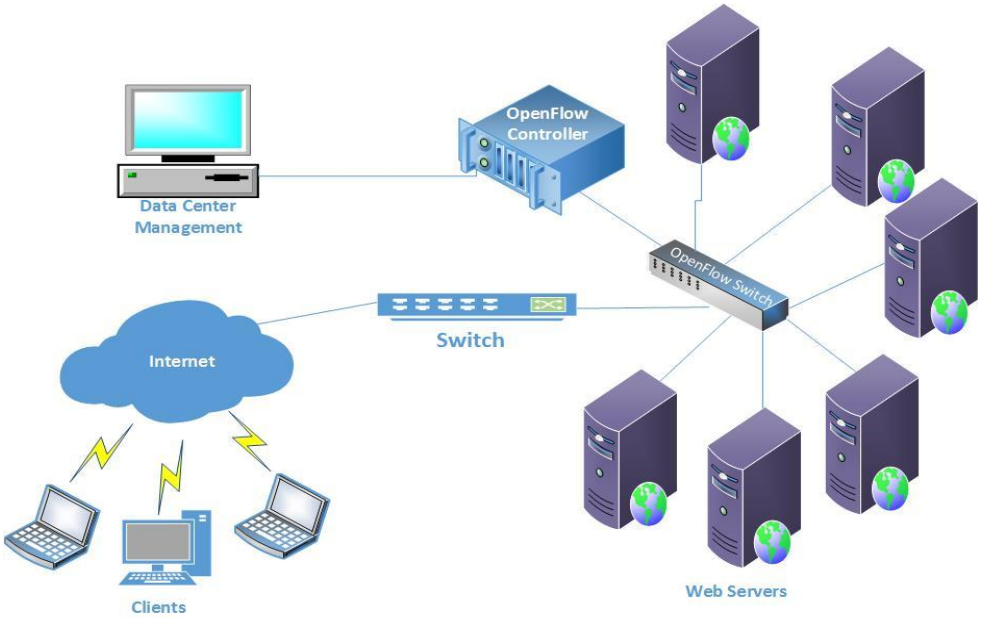

Fig. 1: Proposed Architecture of SDN-based web cluster

The proposed SDN-based web cluster architecture is shown in Fig. 1. The proposed cluster consists of mainly a OpenFlow based controller, a OpenFlow based switch and a set of web servers. An OpenFlow switch is a packet forwarding node which forwards network packets confirming to the rules specified in its flow table. Each row of flow table is called a flow entry or flow rule, each flow entry contains match fields, counters and instructions. These flow entries are added, modified and removed by the controller.

In this proposed cluster architecture the requests are routed by routers to the cluster through forwarding the requests to the centralized web-switch. The OpenFlow based switch is the front end node of the SDN based cluster of web servers which

receives requests and forwards them to the right web servers according the flow table populated by a OpenFlow based controller and if the incoming request does not match any flow entry in the flow table of the OpenFlow switch it sends the packet to the OpenFlow controller. On receiving of a new web service request by the OpenFlow controller from the OpenFlow switch, controller resolves its type of customer and eventually sends the flow entry to the OpenFlow switch to forward that request to a chosen web server according to its load balancing policy.

The functional description of OpenFlow controller:

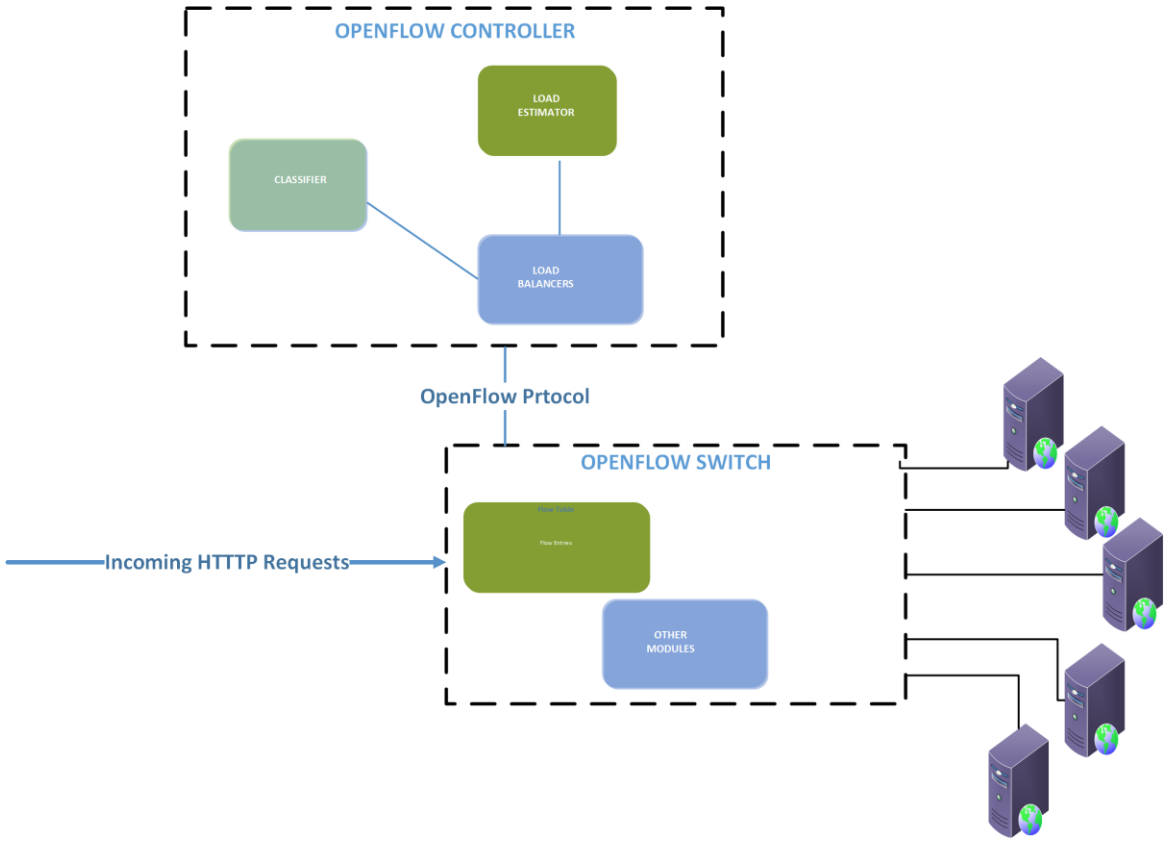

Fig. 2: High level Diagram of OpenFlow Controller and Switch

Fig 2. shows the proposed high level diagram of OpenFlow based controller with its main components from the context of load balancing and also the OpenFlow switch with its main components. The main components of OpenFlow controller in our context are classifier, load estimator and load balancer. Classifier classifies the service requests received in OpenFlow controller as premium customer's request or default customer's request using the information of packet-in message. Load estimator estimates the load of the cluster using the flow counters available in OpenFlow switch. Load balancer by consulting the load estimator finds out the right web server to be used for the given request.

All the client requests to reach the web servers of cluster have to reach the OpenFlow switch using virtual IP address. The OpenFlow switch finds out the right web server to service the request by matching the incoming request parameters with the available flow entries of flow table in OpenFlow switch which are populated by the OpenFlow controller and forwards the request to the server given by the matching flow entry. If no flow entry matches the incoming request, then the OpenFlow switch sends that request to the OpenFlow controller. OpenFlow controller upon receiving the new request as a packet-in message, it classifies the request (as premium customer or default customer) and consults the load estimator module and gives these two inputs to the load balancer. Eventually load balancer module inside the OpenFlow controller finds out the right web server to service the request and this information is sent to the OpenFlow switch to add into its flow table. And OpenFlow switch receives this message and according this flow entry incoming requests are forwarded to the selected web server which is given in the flow entry.

3 SHASDN: Server load balancing using Health monitoring and Admission control in SDN based Networks:

Theoretical background and implementation details of proposed SHASDN, Server load balancing strategy using improved server Health monitoring

and Admission control in SDN networks is described in this section. A single service request needs only a tiny percentage of the resources of the service provider and all the requests are considered as independent. Let n be the number of web servers, μi be the processing intensity for web server i, λi be the arrival intensity of web server i. We calculate this parameter as the average arrivals in a unit of time for each web server. The main objective of SHASDN strategy is to take care of the below SLA condition that all the web servers are not overloaded:

$$\lambda_i \prec \mu_i \qquad (1)$$

An OpenFlow software controller is currently used to install the rules in the forwarding network elements to act on the network traffic through the network and it enables the efficient traffic management. Packets received at this switch are matched with the match field of flow entries. If the packet matches any flow entry, counters of flow entry are updated and the action specified in the flow entry is executed. Controller can balance the traffic by sending the flow rules into the OpenFlow-enabled switches. Classification mechanisms are often used for the improved management of the highly changing workloads (Saeed Sharifian *et al*., 2011; Saifulla M. A. *et al*., 2002; Ying-Dar Lin *et al*., 2009; Saifulla M. A. *et al*., 2003). Frequently requests are categorized according to the importance of customers or type of IP addresses to render differentiated Quality of Service (QoS) requirements (Cardellini *et al*., 2002; Urgaonkar B *et al*., 2005). Load estimation is done by accumulating the flow counters available in the flow table of OpenFlow enabled switch.

The OpenFlow switch operates as an intelligent switch that forwards customer requests to a web server given in the matching flow rule. This intelligence or flow rules are provided by the OpenFlow Controller. The OpenFlow Controller receives the packet-in message from OpenFlow enabled switch whenever a new request does not match any flow entry of its flow table. As the new load balancing request is forwarded to OpenFlow controller, it classifies the request and consults the load estimator module and load balancer module decides the right web server to service the request. Eventually this decision is sent to OpenFlow enabled switch by adding a flow entry into its flow table. The OpenFlow controller judiciously discards some requests from the default customers to reduce the load on web servers in order to assure the SLA to the premium clients and render best effort service to the default clients.

4. Performance Evaluation:

Experimental test setup and experiments are described in this section. Experiments with varying workloads are run to find out the performance of the SHASDN strategy. Initially we ran the tests in mininet (B. Lantz, 2010) emulator as it provides environment to run the real code and also the development of the application is easier and faster. Extreme Networks summit x440 24t is used as OpenFlow switch in our experiments. The hardware configuration of web servers is core 3 2.0 GHz CPUs with 4GB of DDR RAM. We used enough 2.8 GHz core 2 quad machines as the client emulators to ensure that they would not become bottlenecked in any of our experiments. As the web switch uses a unique virtual IP address for the clients reachability, cluster's distributed architecture gets hidden from all the clients. Httperf (D. Mosberger and T. Jin, 1997) is a tool find the performance measurement of web servers. It is used to generate the client workloads.

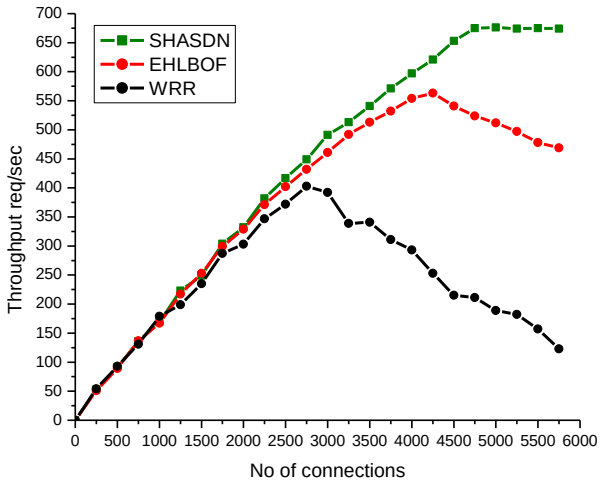

Fig. 3: Average throughput of the cluster vs Total number of clients

We changed the amount of load and calculated three parameters: throughput and response time of cluster for the three load balancing policies, WRR, EHLBOF and SHASDN. In all the experiments, we mixed traffic from both premium clients and also default clients with the ratio of 30:70 respectively. Fig 3. depicts the variation of throughput of the cluster for SHASDN, EHLBOF and WRR. In this experiment the number of clients are increased from 250 to 5750. We considered requests from both types of clients for the measurement of the throughput. Generally the curve of throughput resembles a intersection ∩ shape of set theory; it grows initially, gradually reaches a peak, and then slowly falls. Initially throughput grows, as the total number of clients are increased and then reaches a peak value when the CPU (a bottleneck resource for this case) reaches maximum utilization limit on the web server. After the maximum utilization is reached by a resource, its queue starts building up and causes throughput to fall down. If we remove the hindrance of the resource reaching high, the descending segment of the curve could be prevented.

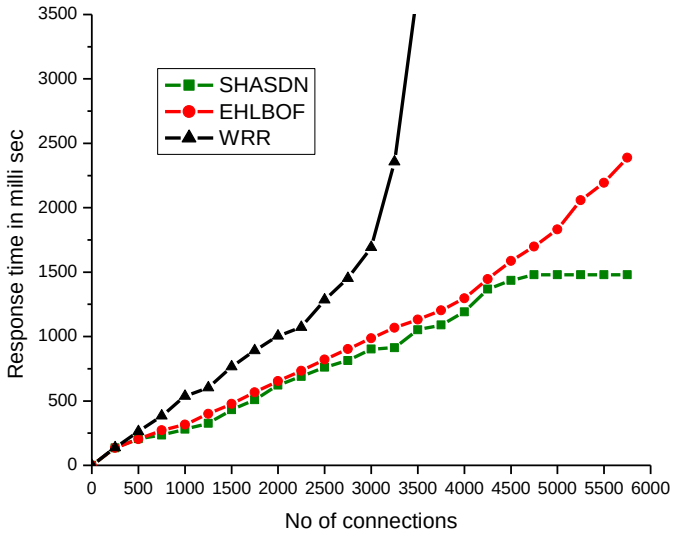

Fig. 4: Average Response time of the cluster vs Total Number of clients

We compare the three curves of throughput for these load balancing algorithms. The curve of throughput peaks at 4750 clients with 676 requests per second for SHASDN strategy. The peak of throughput curve for EHLBOF strategy is reached at 4250 clients with 403 requests per second. The peak of throughput curve for WRR algorithm is reached at 2750 clients with 563 requests per second. The throughput of SHASDN strategy outperforms that of EHLBOF and WRR algorithm. The reason for the lesser throughput for WRR algorithm and EHLBOF is web servers in the clusters are reaching 100% CPU (bottleneck resource for this case) utilization. The throughput of the cluster can also fall down if the load among the web servers of the cluster is not balanced well. SHASDN strategy depicts better throughput as it balanced the load well among the web servers in the cluster.

For the calculation of average response time we considered only requests from premium clients as it is more important compared to default clients. Fig. 4 shows the average response time of the three load balancing schemes, WRR, EHLBOF and SHASDN. This graph depicts how the average response time changes for SHASDN, EHLBOF and WRR. The total number of clients are increased from 250 to 5750 in this experiment. Initially the curve of response time is comparably flat then slowly increases with the total number of clients. By analyzing both the curves throughput curve and mean response time curve, we can observe that the point from where the mean response time curve's slope started to rise sharply is clearly coincides with the peak point of throughput curve (Fig. 3). We can see that the average response time increases as the total number of clients increases. The reason for the high rise of response time is CPU resource on web servers reaches the highest utilization and queue begins building up. As the SHASDN policy uses admission controller when the default customers traffic is high, the exhaustion of resources will not happen and the average response time does not degrade even under high load conditions.

5. Related Work:

M. Rasih Celenlioglu and H. Ali Mantar proposed an SDN-based routing and admission control model (Rasih Celenlioglu. M and H. Ali

Mantar, 2014). In their proposal, controller performs admission control based on PMP(pre-established multi-path)s. Network abstraction, load balancing and path resizing methods are developed to increase network resource utilization. Signaling scalability is achieved by resizing PMPs based on aggregated traffic in offline.

Suneth Namal et al proposed and evaluated a load balancing and admission control mechanism for mobile femtocell (MF), WLAN and Cellular networks using Software Defined Networking (Namal, Suneth et.al 2013). Their mechanism maximizes the resource allocation, and enhance the end-user experience in terms of reduced waiting time and drop-rate.

There are admission control mechanisms for network load balancing available like above in the literature. But our proposal is different and it is novel as we use admission control for server load balancing in data centers along with health monitoring of servers.

6. Conclusions and future work:

In this paper, a novel scalable load balancing strategy SHASDN that uses improved server Health monitoring and Admission control for the cluster of web servers in SDN is proposed. This proposed policy, SHASDN takes care of different client priorities, such as premium customers and default customers. We show that using SHASDN strategy web servers are able to maintain SLAs without the need of a priori over dimensioning of resources. This is achieved by taking the real time perspective of the user requests using measurement of flow counters at that moment and judiciously discard some requests from the default customers if the default customers traffic is high. We analyzed and compared the experimental results of SHASDN policy with the results of the very popular load balancing algorithm, Weighted Round-Robin (WRR) and also Extended Health Monitoring for Load Balancing in OpenFlow based Networks (EHLBOF). We show that even though the SHASDN strategy takes a little more processing resources than WRR and EHLBOF, it is capable to render better service to premium customers counter to WRR and EHLBOF. The core selection & control process of SHASDN policy can be enhanced with the help of real time status information of transactions and also using Business Activity Monitoring for the improvement of load balancing efficiency.

REFERENCES

Andreolini, M. and E. Casalicchio, 2004. “A Cluster-Based Web System Providing Differentiated and Guaranteed Services,” Cluster Computing, 7: 7-19.

Andrews, G.R., D.P. Dobkin and P.J. Downey, 1982. "Distributed allocation with pools of servers," in ACM SIGACT-SIGOPS Symp. Principles of Distributed Computing, Aug., pp: 73-83.

Aron, M., P. Druschel, W. Zwaenepoel, 2000. Cluster reserves: a mechanism for resource management in cluster-based network servers. In: Proc. of ACM SIGMETRICS.

Boone, B., S. Van Hoecke, G. Van Seghbroeck, N. Joncheere, V. Jonckers, F. De Turck, C. Develder and B. Dhoedt, 2010. ``Salsa: QoS-aware load balancing for autonomous service brokering," Systems Software., 83(3): 446-456.

Cardellini, V., E. Casalicchio, M. Colajanni, M. Mambelli, 2001. Web Switch Support for Differentiated Services. ACM SIGMETRICS Perform Eval Rev (29).

Cardellini, V., E. Casalicchio, M. Colajanni, P.S. Yu, 2002. The state of the art in locally distributed web-server systems. ACM Comput Surv (CSUR) 31: 263-311.

Carter, R.L and M. Crovella, 1995. “Dynamic Server selection in the Internet”. Tech. Rep. TR-95-014, Computer science Department, Boston University, Boston, MA.

Casalicchio, E., V. Cardellini, M. Colajanni, 2002. Client-aware dispatching algorithms for cluster-based web servers. Clust Comput., 5(1): 65-74.

Jayabal, R., R.Mohan Raj, 2014. Design and Implementation of Locally Distributed Web Server Systems using Load Balancer, International Journal of Engineering Sciences & Research Technology, 2014 2277-9655.

Katja Gilly, Carlos Juiz, Ramon Puigjaner, 2011. An up-to-date survey in web load balancing, World Wide Web 14(2): 105-131.

Lantz, B., B. Heller and N. McKeown, 2010. A Network in a Laptop: Rapid Prototyping for Software-Definded Networks. ACM SIGCOMM.

Long, H., Y. Shen, M. Guo, and F. Tang, 2013. “LABERIO: dynamic load-balanced routing in OpenFlow-enabled networks,” in Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA ’13), pp: 290-297, IEEE.

McKeown, N., T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, 2008. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2): 69-74.

Mosberger, D. and T. Jin, 1997. “Httperf: A Tool to Measure Web Server Performance,” Proc. USENIX Symp. Internet Technologies and Systems, pp: 59-76.

Namal, Suneth; Ahmad, Ijaz; Gurtov, Andrei; Ylianttila, Mika, 2013, " SDN Based Inter-Technology Load Balancing Leveraged by Flow Admission Control", IEEE Software Defined Networks for Future Networks and Services 2013 - SDN4FNS 2013, Trento, Italy, 11-13 November 2013, 2013.

Rasih Celenlioglu. M, and H. Ali Mantar, 2014, "A Scalable Routing and Admission Control Model in SDN-based Networks", 2014 ACM/IEEE Symposium on Architectures for Networking and Communications Systems

Saeed Sharifian, Seyed A. Motamedi, Mohammad K. Akbari, 2010. An approximation-based load-balancing algorithm with admission control for cluster web servers with dynamic workloads, The Journal of Supercomputing, 53(3): 440-463.

Saeed Sharifian , Seyed A. Motamedi , Mohammad K. Akbari, 2011. A predictive and probabilistic load-balancing algorithm for cluster-based web servers, Applied Soft Computing, 11(1): 970-981.

Saifulla, M.A., A. Hema Murthy, T.A. Gonsalves, 2002. Identifying patterns in internet traffic, Proceedings of the 15th international conference on Computer communication, pp: 859-865, August 12-14, 2002, Mumbai, Maharashtra, India

Saifulla, M.A. Hema A. Murthy, T.A. Gonsalves, 2003. Vector Quantization for Internet traffic modelling, Proceedings of the National Conference on Communications (pp 511 - 515), Chennai, India.

Saifullah, M.A. and M.A. Maluk Mohamed, 2014a "Server load balancing through enhanced server health report", International Journal of Applied Engineering Research (IJAER), ISSN 0973-4562 9(24): 28497-28516.

Saifullah, M.A. and M.A. Maluk Mohamed, 2014b. "Scalable load balancing using virtualization based on approximation", IEEE International Conference on Computer Communication Technology (ICCCT), 11-13, Hyderabad.

Saifullah, M.A. and M.A. Maluk Mohamed, 2015a. "Scalable load balancing using enhanced server health monitoring and admission control", IEEE International Conference on Engineering and Technology (ICETECH), March 20th, 2015 Coimbatore, TN, India.

Saifullah, M.A. and Maluk Mohamed, 2015b. "Efficient server load balancing through improved server health report", ARPN Journal of Engineering and Applied Sciences, ISSN 1819-6608 10(10): 4706-4716.

Saifullah, M.A. and M.A. Maluk Mohamed, 2015c. "OpenFlow based load balancing using enhanced server health reports", Submitted to IEEE International Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG), 2015 Kurnool, Andhra Pradesh, India.

Urgaonkar, B., G. Pacifici, P. Shenoy, M. Spreitzer, A. Tantawi, 2005. An analytical model for multi tier Internet services and its applications. Perform Eval Rev., 33(1).

Wang, R., D. Butnariu and J. Rexford, 2011. “OpenFlow-Based Server Load Balancing Gone Wild,” in Workshop on Hot-ICE.

Wenbo Chen, Zhihao Shang, Xinning Tian, and Hui Li, 2014. “Dynamic Server Cluster Load Balancing in Virtualization Environment with OpenFlow,” International Journal of Distributed Sensor Networks, Article ID 531538.

Ying-Dar Lin, Chun-Nan Lu, Yuan-Cheng Lai, Wei-Hao Peng, Po-Ching Lin, 2009. Review: Application classification using packet size distribution and port association, Journal of Network and Computer Applications, 32(5): 1023-1030.