Literature Survey on Traffic-based Server Load Balancing using SDN and Open Flow

S. Raghul
Department of Electronics
Madras Institute of Technology
Anna University, Chennai, India
Email id: raghulsekar77@gmail.com

Dr.T.Subashri
Assistant Professor,
Department of Electronics,
Madras Institute of Technology
Anna University, Chennai, India
Email id: tsubashri@annauniv.edu

K.R.Vimal
Senior Technical Assistant(SG),
Instructor,
Cisco Networking Academy,
Department of Electronics,
Madras Institute of Technology
Anna University, Chennai, India
Email id: vimalkrme@gmail.com

Abstract- At large data centers, server load balancing is used to balance the incoming requests among the servers to avoid excessive overloading of any individual server. However the present methods are static and suffer from many drawbacks. Static methods do not offer an even balance of data among servers and require dedicated hardware for load balancing.
Hence, dynamic methods are required to enhance the load balance of server. Dynamic method provides load balancing of server inclusion of the dynamic state of traffic on the network. In this paper, the need for dynamic load balancing methods of SDN is discussed and provided with the literature survey of the existing static load balancing schemes .

Keywords- server, load balancing, static, dynamic, SDN

I. INTRODUCTION

Due to the presence of high number of requests in large data centers, using a single server will not be enough to provide service to all the incoming requests. Hence 'n' number of servers are used in order to provide efficiency in attending to incoming requests . They are referred to as the mirror servers and enable load balancing mechanisms. Static balancing methods do not take the size of reply packets into consideration and hence do not provide ideal load balancing. Dynamic methods of server load balancing appear as a better alternative and hence its efficiency needs to be tested. The implementations of server load balancing based on dynamic methods are made simpler by using Software-Defined Networks which provide programmability and customization in networking devices.

Software-defined networking (SDN) is the technology which separates the Control plane from the Data forwarding plane making the network devices such as switches and routers fully programmable and makes the network behave as per the requirements of the individuals, as shown in Fig 1. Thus the network become more flexible, dynamic and cost-efficient and reduces the operational complexity. The benefits include service customizability; configurability improved operations, and increased performance.

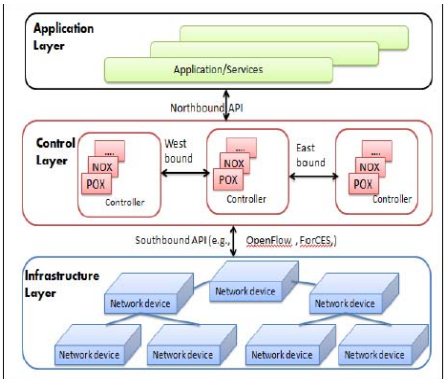

Fig. 1. Architecture of SDN

The SDN architecture consists of: application layer, control layer and the infrastructure layer. The infrastructure layer consists of the switches, routers and other physical components. The control layer consists of the controllers that provide the benefit of programmability by controlling the switches and routers.

978-1-5090-4740-6/17/$31.00 ©2017 IEEE

II. LITERATURE SURVEY

In this paper, a survey on server load balancing methods for Software-Defined Network which is an advanced network infrastructure is provided. The existing network infrastructure methods for server load balance is not suitable for the infrastructure of software defined network where the central controller of network elements are made by programming part. It is easy to control the load for a server which is physically connected to a controller part of the network infrastructure. But, the load balancing for servers on SDN network requires covering the various criteria for server management.

In [1]**, *Nikhil Handigol Et Al*** emphasize the importance of load balancing for servers in data – centers by spreading the incoming requests across several identical web servers. It also briefs about the current load balancing mechanism on SDN. The entry points of network needs a dedicated load balancing servers in order to balance the load on each and every point of network on the SDN infrastructure as depicted in Fig 2. It is also observed that the network load is always constant and static. It is observed that the load balancing is executed on server based on the congestion state of the server. But, it is does not include the congestion on the channel.

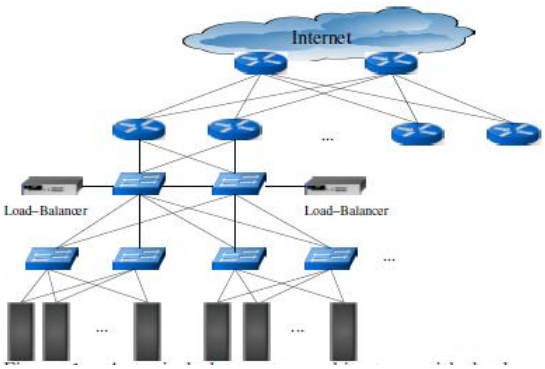

Fig. 2. Multiple entry points – Traditional load balancing

A solution is suggested to integrate software defined networking mechanisms to perform the load balancing in order to mitigate the issues encountered with the current system. This work provides the basic understanding of the need for server load balancing and existing flaws on the existing load balancing scheme. In this work, it is also suggested to move for load-aware server load balancing techniques for advanced network infrastructure like SDN.

The [2] author named ***Ada Gavrilovska*** provided the importance of mirror servers in large data centers is examined and inferred that mirror servers are beneficial to share the load among the replica of the real server.

The author [3]**, *Safae Zerrik*** discusses the architecture of SDN and the its advantages over the traditional network architecture.

In [4], the author discusses the problems pertaining in the current Internet architecture scenario and the changes expected that can eliminate the current drawbacks. In this work, it is mentioned that the Control, Management, and Data Plane separation as a future requirement for providing more security as it provides a separate interface for data and control information.

Wolfgang Braun and Michael Menth et al [5] provided the various innovations that are possible with the Software-Defined networks. Dynamic load balancing is also listed as one of the possible next-generation innovation in the advanced network infrastructure.

In [6] a white paper on SDN by Open Networking Foundation, the efficiency in networking possible with SDN is discussed at length and how SDN differs from the traditional networking devices is also mentioned. The possible use cases of SDN are also studied.

Marc Koerner and Odej Kao et al [7] provided the need for a dedicated hardware module for load balancers is eliminated by the use of Openflow methods. Thus the need for a dedicated load balancer at each entry point is also eliminated by the use of a centralized controller, as shown in Fig 3. Further they elaborate the use of multiple controllers, each for balancing a specific kind of application, say mails, browsers, storage, etc for having a unique balancing algorithm best-suited for each application.

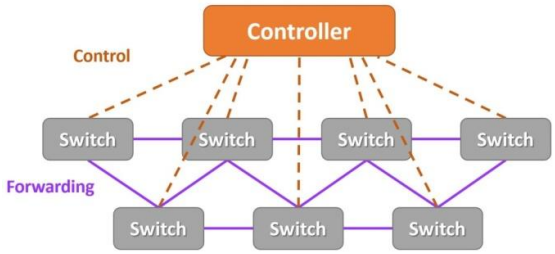

Figure 3: Centralized controller approach

In the paper [8] authored by ***Zhihao Shang Et Al***, the need for dedicated load balancing hardware module is eliminated by proposing the use of dynamic load balancing methods using OpenFlow.

Hailong Et Al *in [9]* compared the performance of static and dynamic load balancing techniques by three experiments namely round-robin, ratio and priority techniques. In round-robin technique, server handles the requests sequentially based on the number of requests or time duration. The server gets the load based on the predefined ratio of traffic on the software-defined network infrastructure. If the load at the main server becomes

978-1-5090-4740-6/17/$31.00 ©2017 IEEE

too heavy to handle, then the backup server is enabled to handle the traffic load on the priority based load balancing techniques.

The dynamic load balancing techniques namely least-connections, predictive and response time are compared with the static load balancing scheme. In least-connections method, new hosts are serviced with least-connections on the network is carried out based on the number of hosts that each server has to service. In predictive connections, mathematical calculations are used to predict the load on each server. In response-time method, a server with the least response-time is made to service the requests. The response time is determined by probing the servers.

From the experimental results *[9]*, it is predicted that Round - Robin method does not take into account the actual load status of each server and does not make full utilization of the sever resources and hence it may not provide real load balancing effect. Both the methods were performed using OpenFlow technique. The experiment uses Linux host, Floodlight controller and Mininet software tool to simulate the switches for software defined network infrastructure. In this paper, the three experiments conducted to prove that least-connections method is more efficient than the Round-Robin Method.

Fig 4 represents the experimental setup for load balancing on network

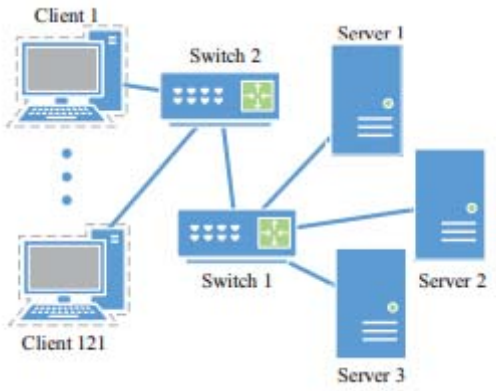

Fig. 4. Experimental Setup for load balancing technique based on least-connection method

Fig 5 displays the experiment results. The parameters for the experiments were set as shown in Table I.

TABLE I. Experimental parameters - least connection method (adapted from [9])

Experiment	Load Status(clients)		
	Server 1	Server 2	Server 3
Experiment 1	30	30	30
Experiment 2	90	30	0
Experiment 3	120	0	0

Fig .5(a). Results for Experiment 1

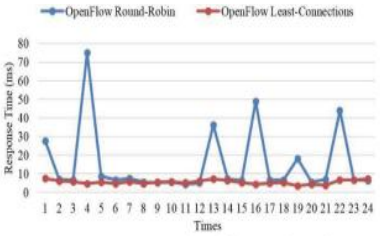

Fig. 5(b). Results for Experiment 2

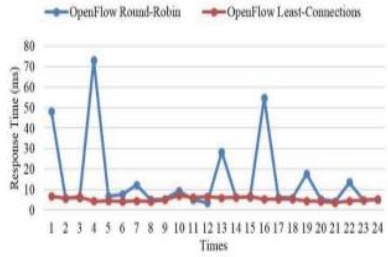

Fig. 5(c). Results for Experiment 3

The inference that was obtained was that when the numbers of connections are equal, both methods offer almost the same efficiency. But when different numbers of connections are present, least - connection method proves to be way ahead of Round robin in terms of efficiency.

In this paper, ***Mohammad Saifullah Et Al*** [10] discusses that load – balancing becomes efficient only if the balancing is done after taking the health status of the server into consideration. The parameters listed under health status of the server are the status of the physical resources of the server and the status of applications available on the server. A solution is suggested is EHLBOF (Extended Health monitoring for Load Balancing in OpenFlow based network) on POX controller. The system probes the servers periodically to get the health status parameters. Incase any server intimates illness, then the status of the server is updated to down and further requests are not sent to that server.

The method EHLBOF is compared with the existing Round-robin based static load balancing method which does not take the health status of the server's health into consideration. The test was executed using mininet software tool with 6 number of servers. It was run on computers with 4GB of DDR RAM with CPUs of 2.0 GHz capacity. The number of requests was gradually increased from 25 to 375 and it was found that EHLBOF method of load balancing provided throughput that

978-1-5090-4740-6/17/$31.00 ©2017 IEEE

was 47.39% more than that of round – robin method . Fig 6 denotes the results of the experiment.

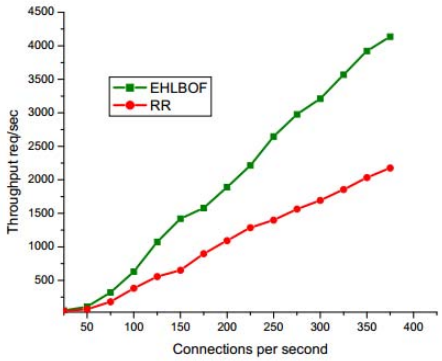

Fig. 6. Throughput comparison for round robin vs EHLBOF

Mei Lu Chin, Chong Eng Tan and Mohd Imran Bandan et al [11] suggested that instead of using the Round robin (DNS based) method for load balancing, balancing the servers after taking the current load and performance metrics of services into consideration will be more efficient and ideal.

In [12] authored by ***Deepika Et Al***, the static and dynamic load balancing methods are investigated to find the advantages and disadvantages of both the systems. A table of comparison as depicted in Table II is drawn between the two methods on factors like fault tolerance, stability, reliability, throughput, resource utilization. It is observed that the dynamic methods possess a better edge over the static ones.

TABLE II. Parameter comparison - static vs dynamic method

Parameter	Round Robin	Central Queue
Type	Static	Dynamic
Waiting Time	High	Low
Throughput	High	Low
Fault Tolerance	No	Yes
Overload Rejection	No	Yes
Adaptability	Low	High
Reliability	Low	High
Predictability	High	Low

Wang Yong Et Al [13] discusses a technique to achieve dynamic load balancing which is traffic-aware as shown in Fig 7. It uses four different modules: traffic detection module, load calculation module, dynamic load scheduling module and flow management module.

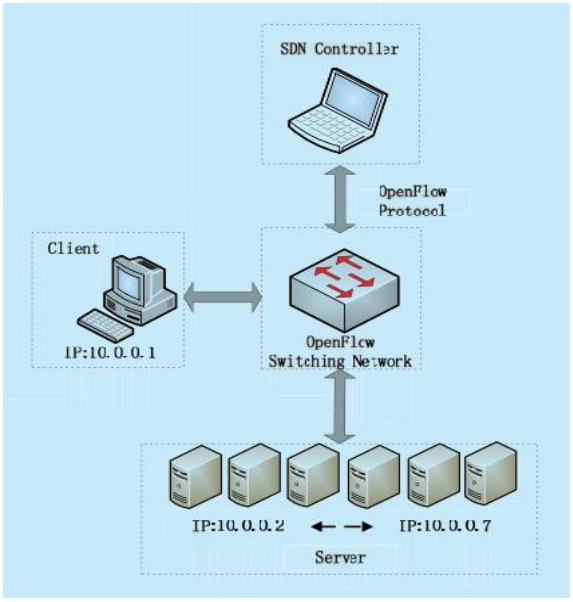

Fig. 7. Traffic-aware load balancing

Dynamic Load Scheduling module:

Dynamic Load Threshold Ω (Maximum imbalance degree) means system imbalance occurs and needs to call the dynamic load balance degree.

When,

$$\delta(t) > \Omega$$

The Maximum load scheduled to the node with the minimum load and the minimum load scheduled to the maximum load node, thus can balance the load among nodes and realize load balance of the system.

Main process:

1. While $\delta(t) > \Omega$
2. Find the least and maximum load.
3. set our node object to *N* and *M*
4. Find the largest flow f1 and smallest flow f2.
5. Dispatch the largest flow f1 to the node N.
6. Dispatch the smallest flow f2 to the node M.
7. End.

Fig 8 displays the threshold setting condition.

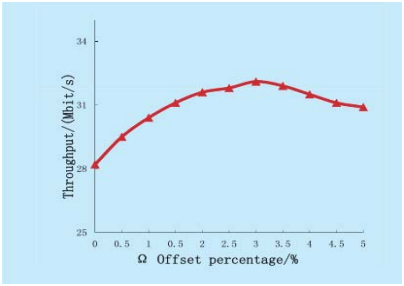

Fig. 8. Setting threshold for load imbalance

978-1-5090-4740-6/17/$31.00 ©2017 IEEE

It is observed that there is no degradation in throughput till the offset percentage reaches 3% and hence it is made the threshold for re-initiating the balancing algorithm.

Hui Long Et Al[14] discusses that dynamic OpenFlow enabled load balancing strategy performs better than the traditional techniques like Round-robin method. Their experiments prove that their dynamic method LABERIO outperforms Round Robin by reducing transmission time by 13%.

Derek l. Eager, Edward D.Lazowska, and John Zahorjan [15] presented a comparison of the simple and complex adaptive load balancing algorithms. Simple algorithms are those which generate lesser overheads, decisions and calculations and the complex algorithms involve a lot of them. It is inferred that the simple algorithms perform on par with the complex algorithms and provide an almost similar efficiency in balancing the loads.

Anna Hat and Xiaowei Jin et al [16] observed that depending on the workload on the distributed processors, the sender-initiated jobs are processed either at locally or remote processors. Depending on whether process is CPU intensive or I/O intensive, the jobs are shared to the relevant processor which has the least amount of the specific workload.

In [17] authored by ***Pushpendra Kumar Chandra Et Al,*** the requests are categorized into CPU intensive, memory intensive and IO intensive. Then the requests are assigned to the appropriate server which has the most unused resources corresponding to the intensiveness of the request processing.

Wenyu Zhou, Shoubao Yang, Jun Fang, Xianlong Niu and Hu Song **[18]** provided methods to balance multiple virtual machines and physical machines. The real time resource utilization of VMs and PMs such as CPU, memory and network is monitored and instant resource allocation is done based on the monitored parameters.

Niranjan G.Shivaratri and Phillip Krueger [19] et al presented a global list that is maintained and updated after each message is received. Each node corresponds to a unique state and is maintained by a unique list. Hence an array of size 'n' is used to store the node's state along with the 'next' and 'previous' pointers to point to the position it the list. Since each node is randomly accessed by the node ID, it is unnecessary to traverse the entire list to change the node's position in the list.

In [20] authored by ***Jie-Fang Liu and Fang-Min Dong***, they propose that for efficiency in the speed of intrusion detection system, parallel handling of data by multiple servers is an ideal option. It is also inferred that load balancing system can be used for the parallel handling of the data.

Natasha Gude [21] discusses the use of NOX controllers to realize Software Defined Networks .A programmable operating system in network devices and centralized programming model is made possible with the use of NOX controllers.

Hardeep Uppal [22] emphasizes the mechanism by which loadbalancing is done using Domain Naming System(DNS) in servers. In a local network, upon an incoming request, the server which needs to service the request is determined by the load balancing algorithms and when the client makes a DNS query for any of the available mirror servers, the address of the server determined to handle the request will be the DNS reply.

George Kornaros[23] mentions how load balancing is done between several cores within a processor. It is also mentioned that the same procedure can be used for load balancing in multiprocessor systems.

Yuanhao Zhou[24] implements traffic based dynamic load balancing using SDN . PyResonance ontroller is used and mininet tool is used as the test bench. The server is switched every time the packet delay time exceeds a threshold duration.

Richard Wang[25] describes that using Openflow load balancing technique, several wild card rules can be incorporated into the load balancing mechanisms to increase the efficiency of the present methods

III. CONCLUSION

In this paper, we have presented various experimental results on comparison between static and dynamic load balancing methods and observe that dynamic methods offer a better efficiency in terms of response time and throughput.

REFERENCES

[1] Nikhil Handigol, Mario Flajslik, Srini Seetharaman; “Aster*x: Load-Balancing as a Network Primitive”, 9th GENI Engineering Conference p1-2, 2010.

[2] Ada Gavrilovska; “Adaptable Mirroring in Cluster Servers”, The Tenth International Symposium On High Performance Distributed Computing 2001

[3] Safae Zerrik, Mohamed Bakhouya and Jaafar Gaber; “Towards a Decentralized and Adaptive Software Defined Networking Architecture”, Fifth International Conference on Next Generation Networks and Services, 2014

[4] Raj Jain; “Internet 3.0; Ten Problems with Current Internet Architecture and Solutions for the Next Generation”, Military Communications Conference, MILCOM 2006.

[5] Wolfgang Braun and Michael Menth; “Software-Defined Networking Using openflow: Protocols, Applications and Architectural Design Choices”, Future Internet 2014

978-1-5090-4740-6/17/$31.00 ©2017 IEEE

[6] Software-Defined Networking : The New Norm for Networks, Open Networking Foundation White Paper, April 2012

[7] Marc Koerner and Odej Kao; "Multiple Service Load-Balancing with openflow" , High Performance Switching and Routing 13th International Conference 2012

[8] Zhihao Shang, Wenbo Chen, Qiang Ma, Bin WU; "Design and implementation of server cluster dynamic load balancing based on openflow", Awareness Science and Technology and Ubi-Media Computing, International Joint Conference 2013

[9] Hailong Zhang and Xiao Guo; "SDN-based load balancing strategy for server cluster", *Proceedings of* CCIS 2014

[10] Mohammad Saifullah and M. A. Maluk Mohamed; "Open Flow-based Server Load Balancing using Improved Server Health Reports", International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics 2016

[11] Mei Lu Chin, Chong Eng Tan and Mohd Imran Bandan; "Efficient Load Balancing for Bursty Demand in Web based Application Services via Domain Name Services", International Journal of Science the and Internet, ISSN Volume 1, No.1, September October 2012

[12] Deepika, Divya Wadhwa and Nitin Kumar; "Performance Analysis of Load Balancing Algorithms in Distributed System", Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 4, Number 1 , pp. 59-66 2014

[13] Wang Yong, TAO Xiaoling, HE Qian, KUANG Yuwen; "A Dynamic Load Balancing Method of Cloud-Center Based on SDN", China Communications, February 2016

[14] Hui Long, Yao Shen, Minyi Guo, Feilong Tang; "LABERIO: Dynamic load-balanced routing in OpenFlow-enabled networks", 2013 IEEE 27th International Conference on Advanced Information Networking and Applications 1550-445X 2013

[15] Derek l. Eager, Edward D.Lazowska, and John Zahorjan; "Adaptive Load Sharing in Homogeneous Distributed Systems", IEEE Transactions on Software Engineering Volume: SE-1,Issue 5, May 1986

[16] Anna Hat and Xiaowei Jin; "Dynamic Load Balancing in a Distributed System Using a Sender-Initiated Algorithm", Local Computer Networks, Proceedings of the 13th Conference 1988

[17] Pushpendra Kumar Chandra; "Performance Analysis of Load Balancing Algorithms for cluster of Video on Demand Servers", 2009 WEJEJ International Advance Conputing Conference Patialae, India, 6-7 March 2009

[18] Wenyu Zhou, Shoubao Yang, Jun Fang, Xianlong Niu and Hu Song "vmctune: A Load Balancing Scheme for Virtual Machine Cluster Based on Dynamic Resource Allocation", Ninth International Conference on Grid and Cloud Computing 2010

[19] Niranjan G.Shivaratri and Phillip Krueger; "Adaptive Location Policies for Global Scheduling Algorithms", Distributed Computing Systems Proceedings 10th International Conference 1990.

[20] Jie-Fang Liu and Fang-Min Dong; "A Dynamic Adaptive Load Balance Algorithm in Parallel Intrusion Detection System", International Symposium on Computer Science and Computational Technology 2008

[21] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, Scott Shenker; "NOX: Towards an Operating System for Networks"; ACM SIGCOMM Computer Communication Review Volume 38, Number 3, July 2008

[22] Hardeep Uppal and Dane Brandon; " OpenFlow Based Load Balancing", University of Washington CSE 561

[23] George Kornaros, Theofanis Orphanoudakis, Nickolaos Zervos; "An Efficient Implementation of Fair Load Balancing over Multi-CPU SOC Architectures", Proceedings of the Euromicro Symposium on Digital System Design 2003

[24] Yuanhao Zhou, Li Ruan, Limin Xiao, Rui Liu; "A Method for Load Balancing based on Software Defined Network, Advanced Science and Technology Letters Vol.45 pp.43-48 CCA 2014

[25] Richard Wang, Dana Butnariu, and Jennifer Rexford; "OpenFlow-Based Server Load Balancing GoneWild, Proceedings of the 11th USENIX Conference on Hot Topics in management of internet, cloud, and enterprise networks and services, 2011.

978-1-5090-4740-6/17/$31.00 ©2017 IEEE