Research Journal of Engineering Technology and Management (ISSN: 2582-0028), Volume 02, Issue 01, March-2019
Available at www.rjetm.in/

Video streaming over software defined networking with server load balancing

Yogita Gawade[1],Amruta Harale[2],Dnyaneshwari Ekhe[3],Shahana Anjum[4] and Pranali Lokhande[5]

[1,2,3,4,5] Computer Science Engineering, SPPU, MIT AOE Pune India.

[1]yogitagawade16@gmail.com, [2]amrutaharale83@gmail.com, [3]sha.anjum57@gmail.com,

ekhednyaneshwari@gmail.com[4] ,pplokhande@comp.maepune.ac.in.[5]

* Corresponding Author:Yogita Gawade Manuscript Received: Manuscript Accepted:

Abstract

Abstract—Software defined networking internet architecture is used efficiently to deliver multimedia services with efficient Quality of Service (QoS).We are proposing the model for server load balancing over a only one operator openflow network to improve quality of service(QoS) levels of video streaming services over servers.For this purpose,we will design a controller.This model will perform dynamic server load balancing by continuous looking for load of each server and dynamic rerouting of clients to change servers with lower load.It will improve the server performance provide good quality of service to end user by load balancing technique.

Keywords: *COLBAS,RC,SC,SDN,FSM.*

I. INTRODUCTION

Dealing with the design and administration of traditional network seems quite difficult as they are equipped with various hardware devices such as routers, switches, and middle boxes like load balancer, network address translator, intrusion detection system, firewalls etc. which are non-programmable and vendor specific i.e. any alteration as per the requirement can be done only by the vendor which in case of complex network can be challenging and fallible. If there is a need to add or remove any functionality from the network, it cannot be done without tampering the network infrastructure and will directly affect its logic. Also settings up any new protocol server have to pass through various testing and standardization to ensure interoperability provided by particular vendor. Software defined network (SDN) [1] has evolved as the new network frame- work which is programmable and vendor neutral. It conquer the shortcomings of long-established network architecture by abstracting the main intelligence of network i.e. control plane from forwarding plane (data plane). The centralized controller is the brain/core of SDN network which is responsible for broadcasting information to the switches/routers residing below it via southbound interface and application/business logic residing above it via northbound interface. Any communication between controller and network elements are governed by Open flow protocol [2].

Load balancing approach will be studied deeply in literature survey. Regular load and static load of applications can be performed by mapping data onto different processors. Numerous algorithms have been developed for statically partitioning a computational traffic. This model represented in the form of graph where graph is partitioned among different processors. Graph and hyper- graph partitioning techniques are used to map tasks onto different processors for load balancing.These techniques are used for preprocessing purpose which are tend to be expensive. Partitioning is done at the start & then algorithm is applied and it applicable to balance the load imbalance that causes as the application proceeds further.Next task is to consider the present mapping and shifts tasks only if a processor is burdened with data.For asymmetrical requests, work stealing is used for task development and is part of runtime systems such as Cilk Work stealing is normally used for task parallelism of the one seen in the divide-and-conquer applications, where tasks are being produced constantly. A recent work by Dinan et al. balances work stealing to 8192 processors using the PGAS programming model and RDMA. In effort that trailed, a classified technique designated as retentive work stealing was used to scale work-stealing to over 150K cores by taking advantage of the principle of perseverance to improve the load balance of task-based applications.CHAOS provides an inspector-executor approach to load balancing for irregular applications. At runtime before the start of the first iteration to rebalance, the data and the associated calculation balance are resolute. The planned strategy is more attentive towards iterative computational science applications, where computational tasks tend to be insistent.Centralized, distributed and hierarchical are the main categories of Dynamic load balancing algorithms for iterative applications. Integrated strategies tend to be good load balancing but it has poor scalability. Processors independently make load balancing decisions based on workload of specific servers to apply several distributed algorithms. Common nearest neighbor algorithms are dimension-exchange and the diffusion methods. In repetitive fashion, dimension-exchange method is done and is defined as the hypercube architecture. In each aspect of the hypercube, a processor completes load balancing with its neighbor. First Cybenko and later on Boillat specified diffusion based load balancing algorithms. Slow merging to the balanced state is occurring in this algorithm. A non-local method to regulate the flow has been suggested by Hu and Blake, which is slight in the l2-norm but needs inclusive communication. The load is reflected as a mark in the token distribution problem studied by Peleg and Upfal.

II. OBJECTIVE

1.To increase the efficiency and reduce downtime by applying SDN load balancing technique.

2.To Handle Peak Performance.

3.To detect and Manage Failures.

III. LITERATURE REVIEW

Although SDN is widely used network architecture there are various issues such as load balancing that needs to be taken care of[1]. Load balancing is an intelligent congestion cognizant routing in Software Defined Network (SDN). In any SDN based network architecture, load balancing is an indispensable entity to promote the availability and scalability that further leads to attain the minimal response time of application.In [3],the load balances aims to disseminate the client re- quest among available servers to prevent the single server from being overburdened. In case of server failure, load balancer provides fault tolerant capability by redirecting the request to the remaining server. Traditional load balancer uses dedicated hardware which was vendor locked and non- programmable, so the network administrator cannot change or write new load balancing algorithm which makes them inflexible and non-scalable. The outlay of purchasing any dedicated load balancer is very large. So large number of load balancer will be required if the network is large.In [4],it is seen that overall cost of the network increase without use of SDN. With the advent of Software Defined Network which is programmable in nature, allows to code and implement own load balancing algorithm converting the dumb physical switches into powerful load balancer using SDN Controller. It provides agility to quickly adapt to changes and flexibility to add new trait to existing network architecture as and when required The load balancer architecture consists of server pool connected to the load balancing device. The entire incoming request will be directed to virtual IP address of load balancer. Load balancer will overwrite the destination IP with one of the server IP. Different load balancing algorithm will be used to select one of the server. Once the server process the request, reply will be sent back to load balancer where it will change destination IP to Address of host. Different strategies to select the server are discussed below.The most popular stateful algorithm for load balancing is Hash-based [5]. The algorithm works by first calculating the hash value of traffic flow using IP address of source and destination, port number of source and destination and URL.The request is then forwarded to the server with highest hash value. If any other request comes with same hash value, it will be forwarded to same server. For same server it creates unique hash value. If the request is coming from same host again and again, this algorithm cannot scatter the load across different links. The limitation of this algorithm is inability to generate different hash value for source and destination host.

In [6] it is discussion on multiple service load balancing strategy having multiple controllers to service load of different type of servers. For example, to handle web server and email server, there will be different load balancer. FlowVisior [7] is used to slice the network resource and assign it to different controllers. The major limitation with this paper was large overhead due to multiple controllers. The link delay parameter was not considered which can greatly affect the performance.Other load balancing model was based on PyResonance controller [8]. Resonance is SDN control platform that pro- mote event-driven network control. To define network policy it preserves Finite State Machine (FSM) model. Different states shows action depending on network condition. Transition between states shows reaction to dynamic network event. The controller control and balance the network flow by mapping the host (containing FSM) to state it is in. This module however, was just a prototype. In actual SDN environment it is more complex to balance the traffic.

In [9],a load balancing technique for handling client request at line rate enhancing the network performance by proper utilization of available resources is discussed. The main aim of this strategy is to increase throughput and minimize latency and response time by scattering the incoming packets in round robin manner.[10].The round robin algorithm uses circular queue to determine the server to whom the request will be forwarded. The load balancer comprises of IP address of service and server. The entire client request will go to service IP (address of load balancer) and load balancer will redirect the request to server in round robin manner. So the method is actually balancing service IP and to check whether the server is live or not, ARP probes are sent to the servers. The experiment shows that round robin strategy outperforms random load balancing strategy.For dynamic load balancing of server, it is very important to make the flow table dynamically to reflect the current load.In [11],an algorithm to dynamically design the flow table and classify different client processing is presented. It was based on single flow table (for single client) and group flow table (for multiple client request). Single flow table is used for traffic inspection of each client while multiple flow tables are used to perform traffic forwarding function. It reduces the overhead of maintaining large number of flow tables for each client.

In [12],there is another load balancing policy named as weighted round robin. The architecture and implementation was same as in case of round robin but the only difference here was that a static weight was attached to each server which was directly proportional to servers actual capacity i.e. if the capacity of server A is 5 times the capacity of server B the weight assigned to server A will be 5 and that to server B will be 1.This load balancing method is very beneficial in case when there are 2 servers with equal capacity. We want one to get less connection because it is accomplishing some critical task and should not be easily overloaded. Other perk of using weighted round robin algorithm is in case of heterogeneous server i.e. one server running i5 processor and other running i3 processor then we assign static weight to the servers so that server with lower capacity will get less number of request as compared to the one with higher capacity. Although weighted round robin give more prominent result as compared to round robin but it only solves the problem of server being homogeneous. Other limitations of round robin still exits.In [13] a load balancing strategy is proposed in which unnecessary delay is eliminated by not involving load balancer in return traffic i.e. the response is directly send to the client. This scheduling algorithm works by selecting the server with lowest flow connection. A flow statistics message is send to switch by load balancer after every 5 second. Flow Statistics Received Event occurs when the switch send its flow statistics information to the controller. Number of flows sent to the server by load balancer is counted by load balancing application and packet is send to server with least active connections.

However using single controller may cause scalability and availability problem. Using multiple controllers can alleviate this problem in wide area networks. There is still absence of any flexible load balancing mechanism for distributed controller. COLBAS [14], a load balancing scheme for hierarchical controller configuration was proposed by Selvi et al. This method relies on controller collaboration via cross- controller communication. It assigns one of the controllers as Super Controller. The hosts connected to switches generate request according to poison process with interarrival time of packet exponentially distributed. The super controller collects the load metrics from the controller and check if there is any imbalance. If the request handled by any controller exceeds the upper threshold, Super Controller continues to redistribute the request until the load of a controller reaches lower threshold value. In case if multiple overloaded controller and link/node failure, how the mechanism will work was not explained by the authors of this paper. The major limitation of this paper was that it considers packet size to be identical, which is not always possible.

Sufiev et al. [15] proposed Dynamic Cluster, a multi- controller load balancing method for SDN. The architecture consists of a Super Controller (SC) and various clusters of Regular Controller (RC). Each cluster must contain equal number of RCs. Load balancing is performed at two levels:- low level load balancing occurs when any RC reaches its threshold value and high level load balancing occurs when on periodical checking, initiated by SC ,it is realized that some cluster have reached its maximum load threshold. To break the interdependency between SC and RC, Cluster Vector (CV) is defined that contain the address of all the RCs in a cluster. Whenever there is cluster imbalance, SC will run partition algorithm to rearrange the RCs in a cluster and return the new CV to the controllers. So two clusters cannot directly communicate with each other without the involvement of Super controller. There can be availability problem in case the Super Controller fails. Also due to periodic collection of load information by Super Controller can cause communication overhead.

DALB algorithm [16] is put forward for load balancing. This method was completely based on distributed decision i.e. there will be no centralized controller. Every controller will first measure its load to check if it surpasses the threshold value. If yes, then it will collect the load information of other controllers and calculate the value of load balancing rate and maximum load. If load is large then it will identify the switch with maximum load and migrate it to low load controller. To avoid the frequent collection of load information from controllers, dynamic and adaptive threshold is adopted. But the limitation with this mechanism is that whenever a controller becomes overloaded then only before making load decision it is collecting the load information of other controller, which reduces the time efficiency of load balancing.

Another approach for load balancing in SDN architecture with multiple controllers [17] which was based on load information. The controller makes the load balancing decision at its own level. Every controller will repeatedly broadcast the information about its load to all the other controllers and save the load information of other controllers to make load migration attainable. So overloaded controller need not collect the load information of other controllers before making decision. Periodically informing the controller about load reduces the time to make decision but on other hand it can cause communication overhead. If present load on controller has not changed much as compared to the last value then notifying it to other controller will only lead to superfluous operation. So the author even proposed an algorithm to lessen the frequency of informing. Each load balancing module in controller consists of: (1) measurement of load to check if controller has reached its threshold value (2) informing about load to other controller (3) make load balancing decision and (4) shifting the switch to other controller to balance the load on local controller.

In [18],an algorithm is implemented that assign load to server according to delay. It assigns delay to each link between server and switch according to the speed and dynamic weight is assigned according to delay. The server with minimum delay handles more traffic in contrast to server with more delay. Although this strategy consider parameters like link delay, link speed which is very important from performance point of view but due to single centralized controller there can be single point of failure.In [19],a load balancing algorithm is proposed that is based on artificial neural network. This algorithm works by measuring the four parameters of every transmission path i.e. packet loss rate, transmission hops, bandwidth utilization ratio and transmission latency.

IV. ACKNOWLEDGMENT

We might want to express our profound feeling of appreciation and regard towards our guide Prof.Pranali Lokhande, Department of Computer Engineering, MIT Academy of Engineering.we have gotten from her while gathering information on this paper and all through our investigations.. we whole- heartly thank to our Dean Prof.Ranjana Badre for their guidance. we also indebted to all Sr. Engineers and others who gave us their valuable time and guidance. The various information and sources we used during my report completion find place in our report.

V. CONCLUSION

We have proposed load balancing server and efficient use of open flow controller ,a detailed survey of load balancing has been done. Software Defined Network has been developed to manage large networks like cloud computing technology, wide area networks, data centre big data.With this technology a new OpenFlow controller application is designed for load balancing to improve quality of service levels. Proposed system will work in college e-learning applications that handle multiple requests.

VI. REFERENCES

[1] Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2015): 14-76.

[2] Vaughan-Nichols, Steven J. "OpenFlow: The next generation of the network?" Computer 44.8 (2011): 13-15.

[3] A. Voellmy and J. Wang, “Scalable software defined network controllers,” SIGCOMM Comput.Commun. Rev., vol. 42, no. 4, pp. 289–290, Aug. 2012.

[4] Serverwatch.com, “5 Load Balancers You Need to Know,”2015.

[5] “Load-Balancing: Hash Methods,” Calix, 2010.

[6] Koerner, Marc, and Odej Kao. "Multiple service load-balancing with OpenFlow."High Performance Switching and Routing (HPSR), 2012 IEEE 13th International Conference on.IEEE, 2012.

[7] Sherwood, Rob, et al. "Flowvisor: A network virtualization layer." OpenFlow Switch Consortium, Tech. Rep 1 (2009): 132.

[8] Zhou, Yuanhao, et al. "A method for load balancing based on software defined network." Advanced Science and Technology Letters 45 (2014): 43-48.

[9] Kaur, Sukhveer, et al. "Round-robin based load balancing in Software Defined Networking." Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on.IEEE, 2015.

[10] Ghaffarinejad, Ashkan, and Violet R. Syrotiuk. "Load balancing in a campus network using software defined networking." Research and Educational Experiment Workshop (GREE), 2014 Third GENI.IEEE, 2015