CHAPTER 1
INTRODUCTION

1.1. Background

In a distributed architecture, optimal load balancing needed when network overhead problems arise. In this case, an adaptive threshold adjustment strategy is needed based on the cell load and the proportion of various types of cross-tissues. It can not only solve the problem of heavy load and uneven traffic, but also get better network throughput [1]. Server limitations are generally an obstacle affecting the quality of service (QoS) due to increased traffic levels. Load Balancing is needed to manage service requests to the optimal application server [2]. It is important to implement a load balancing strategy to increase performance and resources in the Cloud Data Center. There is a modeling approach to the virtual machine scheduling problem with capacity sharing by modifying the traditional interval scheduling problem [3]. However, the increased costs and technical complications in the deployment of such hardware systems often require human intervention to ensure the consistent functioning of the strategy [4]. Therefore, Load Balancing technology is used to distribute network resources evenly in order to increase network performance.

A new concept arise, namely Software Defined Network (SDN). SDN is a new architecture for networks that is dynamic, manageable, cost-effective, and adaptable. SDN architecture aims to make the network more flexible and make it easier to control the network. The capabilities of the SDN network can also be used to change network behaviour and can make changes automatically such as maximizing traffic load sharing using load balancing.

On SDN, the Load Imbalance problems can still occur. If the network load distribution process is uneven, it will affect the overall performance and efficiency of the network. In general, a balanced traffic load in the network helps optimize the utilization of available resources by maximizing throughput, minimizing response time, and avoiding load overloading on the network. Therefore, Load Balancing technology is used to distribute network resources evenly in order to increase network performance.

Load Balancing technology in SDN can be classified as Server Load Balancing and Link Load Balancing. When there is a link congestion caused by disturbances in Traffic Scheduling Management on the network, the network performance is decreased and the transmission delay increased. By using Server Load Balancing strategy, we can allocate traffic load to different servers to avoid link congestion [5]. The method used in the Load Balancing system usually uses algorithms such as randomized round robin which is a static algorithm. [6] The research distributes the server performance load based on the server performance weight to determine the server that serves the request.

From the results of this study, the distribution of the load will be directed to a predetermined server based on server weight with a value of 60% to server 1, 30% to server 2 and 10% to server 3. Server 1 will get a greater performance load than other servers. Therefore, the Weighted Round-robin method is less efficient and does not consider the actual conditions of each server and can cause server overload. Then server performance refers to a lot of resource usage for example, CPU usage, memory usage, network and disk usage. [7]

The least-bandwidth algorithm selects the server with the lowest network traffic consumption to respond the next request. Like the round-robin method, the least-bandwidth can also accept weights on the servers if w_i is the weight of the server i and b_i the network traffic consumption of it, then the machine with the lowest ratio bi is selected to respond the next request, thus, machines with larger b_i/w_i weights receive more load than machines with smaller weights. [8]

POX Controller which can distribute the server performance load according to the state or server status. The state of the server is the weight of the performance set by the Controller in accordance with the capabilities of each server used. The controller determines the weight value (weight), and the Controller can determine the load from each server to handle requests. [9]

An agent that is planted on the server. Agent functions to send the results of CPU usage data used by each server. It then distributes traffic based on the lowest CPU usage that each server has. The function of the agent will be used so that the needs of this research are met. Therefore, agents are used in this study to send server resource information to the controller. [10]

The load balancing system is applied to a Software Defined Network using a Fuzzy algorithm that can distribute the workload based on the lowest server weight. Then a comparison is made with the Response time (RT) algorithm and the Least Connection (LC) algorithm with the parameters of traffic distribution, CPU and Memory Usage, response time and throughput.

In testing with a request load with a total rate of 360, the Fuzzy algorithm imposes more traffic distribution on the server with the lowest server weight. Then in the memory usage test, the response time algorithm has the lowest memory usage on each server, namely 18.7% server1, 32.4% server2 and 62.2% server3. Furthermore, in testing with a load at a rate of 360 the RT algorithm has the smallest response time, namely 393 m / s. In testing the response time algorithm throughput has the largest throughput, namely 93.6 Kbyte/s. [11]

The research introduces an enhanced ant colony algorithm by including a fuzzy logic module to calculate the pheromone value and the Taguchi concept to optimize the algorithm parameters. The experimental results achieved by the Cloud Analyst simulator have confirmed that the algorithm is more appropriate for handling complex network. The main contributions of the proposed solution still the application of fuzzy logic to calculate the pheromones and the use of the Taguchi concept for selecting the best ACO parameters. This algorithm includes a procedure of evaporation from the trial of pheromones to avoid earlier convergence towards non-optimal solutions. The achieved simulations within the Cloud Analyst platform showed that the proposed approach allows improving load balancing in the Cloud architecture while reducing response time by up to 82%, the processing time by up to 90% and total cost by up to 9% depending on the applied scenario. [12]

In research conducted by Wang and Guo, 2017. This research discusses the load balancing system with Fuzzy Logic in SDN architecture for connection line weighting mechanism which refers to the weight of the connection to determine the closest route. This study aims to optimize the performance of the load balancing system to dynamically distribute traffic loads based on the shortest path. For this reason, to meet the needs of this research system, it will use the same algorithm for server weighting mechanisms in distributing traffic loads based on server resource usage such as CPU usage, memory usage and disk usage. [13]

Basic SDN feature for flow control can be used for load balancing. In load balancing process the inbound IP (internet protocol) traffic can be divided into different flows to multiple servers, is also improve the working capacity of servers. It forwards the request of the client to the backend servers and gets replies from servers, which it replies to the client . While the fuzzy method has the advantage that it is suitable to be applied to problems that contain an element of uncertainty like RTT delay, throughput, round trip time (RTT) and device conditions from each server.

The Performance for load balancing will be better because SDN can direct control the traffic flow to multiple server. It can also be improved by selecting the server with minimum load using Fuzzy Logic Algorithm [14]. Stability in the Server Load Balancing process is deemed necessary with the intention that the flow direction in the data flow table does not change too often.

The LBBSRT method provides a Load Balancing solution using the Load Balancing Based Server Response Time (LBBSRT) method. LBBSRT uses SDN controllers to get response time values from each server in order to select a server with the best response time value. The SDN controller sends multiple packet-out messages to the switch with a time interval t and stores the result as the transmission time. But the LBBSRT method does not consider the energy and resource consumption side of the server [15].

This research aim is to propose a new solution to solve these problems with an SDN-based Load Balancing system using the Fuzzy Logic Algorithm method. It aims to be able to distribute the server load based on the server's capabilities. Each server will be weighted by the controller dynamically based on CPU, memory and network utilizations parameters that have gone through the fuzzy logic calculation stage. Then the distribution of the traffic load is determined based on the smallest server load window.

Contribution of the paper include:

a) An effective load balancing scheme based on SDN architecture, using the real-time CPU, Memory usage and network utilization of each server measured by an SDN controller.
b) Realizing the implementation of our design by using a Ryu controller.
c) Proving the effectiveness of our design by evaluating the resource utilization, throughput, fairness index and Request distribution metrics against the traditional schemes.

The rest of the Book is organized as follows: Chapter 2 reviews the existing load balancing schemes and introduces the background of SDN. Chapter 3 consists of our proposed scheme Fuzzy Logic Load Balancing. Chapter 4 consists of performance evaluation. Chapter 5 concludes the book.

1.2. Problem Identification

When the network load distribution process between servers destination is uneven, it will affect the overall performance and efficiency of the network. Traditional Load Balancing methods like random and Round Robin didn’t divide the workload of the servers evenly, especially if the initial conditions of the server when loaded are not the same. Another example Load balancing on SDN using the LBBSRT algorithm did not consider resource utilization variable on the server (CPU Utilization, RAM Utilization, Network Utilization). [15].

1.3. Scope and Delimitations

A custom design for the research has been made with simple topology design run on Mininet. Ryu controller is used because this controller is opensource. The HTTP Request generated using Apache AB Load Testing and Network Traffic Injection generated using iperf3. Fuzzy Mamdani method is used for fuzzy logic algorithm. Measurement metrics are evaluating the resource utilization, throughput, fairness index and Request distribution metrics.

There are several types of variables in the research :

a) Dependent variables: Throughput, Round Trip Time, number of successful requests, CPU usage, memory(RAM) usage.

b) Independent variables: number of nodes, number of links, number of the switch, number of server, number of concurrent request.

c) Control variables: total number of request such as HTTP Request.

1.4. Research Purpose

The primary objective is through measuring the performance of Fuzzy Logic Algorithm with Server Resource Variable for Load Balancing on SDN in Low Load Scenario and High Load Scenario. The second is to compare with traditional Load Balancing algorithm. The simulation results and the conclusions of this thesis contribute to making the decision for deploying Load Balancing Scheme on SDN environment.

1.5. Hypothesis

Table 1 explains about basic features of SDN and advantages for load balancing. Basic SDN feature for flow control can be used for load balancing. It is as an aware routing protocol that can maximizing the throughput and minimizing the interval of round trip time and reducing the jam. In load balancing process the inbound IP (internet protocol) traffic can be divided into different flows to multiple servers, is also improve the working capacity of servers. It forwards the request of the client to the backend servers and gets replies from servers, which it replies to the client [16].

While the fuzzy method has the advantage that it is suitable to be applied to problems that contain an element of uncertainty like RTT delay, throughput, and device conditions from each server. Usage of Server's resource utilization data like CPU usage, memory usage, network utilization could also improve performance of load balancing

The Performance for load balancing will be better because SDN can direct control the traffic flow to multiple server. It can also be improved by selecting the server with minimum load using Fuzzy Logic Algorithm.

Table 1. SDN Feature and Advantage for Load Balancing [14]

Feature	Advantage
Dynamically programmable	They can be programmed before deployment and while the network is running.
Centralised control plane	It simplifies provisioning while optimizing performance and provides management for granular policy
API (Application Programming Interface)	It simplify configuration as per the needs
Flow Control	It can be used for load balancing

1.6. Definition of Terms

For the better understanding of this study, the following terms are defined in the context of this research.

a) Network Utilization – byte receive per sec : Network Adapter performance counter which is measured by rate at which bytes are received over each network adapter, including framing characters.

b) CPU Usage : current system CPU utilization which is measured as a percentage.

c) RAM Usage : current system memory (RAM) utilization which is measured as a percentage.