Jurnal Mantik
Volume 4, Number 3, Nov 2020, pp. 1588-1596 E-ISSN 2685-4236
https://iocscience.org/ejournal/index.php/mantik

Load Balancing Method Performance Analysis on Haproxy and Router OS

Yoga Permana[1], Ritzkal[2], Yuggo Afrianto[3]

[1,2,3]Technical Information, Faculty Of Engineering And Sanis, Ibn Khaldun University Bogor,Indonesia

E-mail: [1]yoga.260497@gmail.com ,[2*]ritzkal@ft.uika-bogor.ac.id, [3]yuggo@uika-bogor.ac.id.

ARTICLE INFO

Article history:
Received: 12/07/2020
Revised: 22/08/2020
Accepted: 30/09/2020

Keywords:
System Load Balancing, Haproxy, Router OS

ABSTRACT

Increasing requests for client connections to servers on a network can cause a problem with server performance to decrease and too many user requests make the workload on the server increase rapidly resulting in the server being down, so we need technology that handles complex network usage. Then it can be solved with a load balancing system. Several studies on load balancing on haproxy and router OS with various algorithms applied will produce different performance, including round robin, least connections, and source algorithms on haproxy load balancing, while NTH, PCC, and ECMP on load balancing routers OS. Testing is done with 3 scenarios, scenario 1 with a load of 10 users, scenario 2 with a load of 20 users, and scenario 3 with a load of 30 users. The test parameters used are throughput, delay, and jitter using the webserver stress tool and wireshark. The results of the performance measurement of the load balancing method on haproxy generally show that the least connection is better than the round robin and source methods, this can be seen from the measurement figures for delay and jitter parameters that are best found in the loast connection method. While the results of the performance measurement of the load balancing method on the OS router, there are two, one shows that PCC is better, this can be seen from the measurement numbers of the delay and jitter parameters, then the second shows that the ECMP method is better, this can be seen from the throughput parameter measurement figures and delay.

1. Introduction

Increasing requests for client connections to servers on a network can cause a problem with server performance to decrease and too many user requests make the workload on the server increase rapidly resulting in the server being down, so we need technology that handles complex network usage. Then it can be solved with a load balancing system. Load balancing is a technique of distributing traffic loads on two or more lines, so that a balanced connection is obtained, more optimal traffic, maximum data throughput, minimal delay, and no overload occurs [1]. There are several algorithms that can be implemented in load balancing systems, such as round robin, least connections, and source in haproxy load balancing, while NTH, PCC, and ECMP are used in router OS load balancing.

There are several studies that have been done on load balancing systems, including: Husain Nasser (2016) has conducted research on round robin, least connection and ratio algorithm analysis on load balancing using opnet modeler. Muhammad Iqbal Firdaus (2017) conducted a comparative analysis of the load balancing performance of the ECMP (equal cost multi-path) method with the PCC (per connection classifier) method on Mikrotik RouterOS. Hasta Triaga (2019) conducted a comparative analysis study of the static round-robin algorithm with least-connection on the efficiency of load balancing on the haproxy load balancer. Abe Wisnu Syaputra (2017) conducted an analysis and implementation of load balancing using the NTH method in the Jambi provincial education office network. Pratama Ihsan (2019) implemented server load balancing using haproxy with the roundrobin algorithm and implementing file cache.

From several studies that have been done, this research is looking for a load balancing system that is more optimal to keep server services reliable, by analyzing several load balancing methods, namely round robin, least connection, and source on haproxy load balancing, while PCC, NTH, and ECMP on load balancing router OS. Load balancing operating systems that will be tested in this study are haproxy and router OS. Testing is carried out on a virtualization network with an analysis that will be carried out on the loading of traffic between two or more connection lines on the web server service. After each method is run on each operating system, the results of testing and analysis will be obtained to see which method is optimal for load balancing systems on haproxy and router OS. The formulation of the problem of this research are (i) How to implement load balancing haproxy and router OS on a web server kite? The objectives of this study

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

are (i) Obtain the results of implementing load balancing on haproxy and router OS. (ii) Get results of the workings and performance of each method that supports load balancing of haproxy and router OS.

2. Research Method

This research method is a framework for carrying out an action or frame of mind, to compile an idea that is directed and related to the aims and objectives. The method used in this research is as shown in Figure

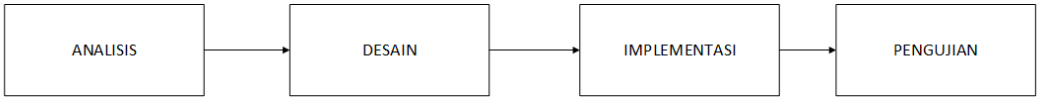

Fig 1. Research Method

3. Result

3.1 Analysis

In the analysis of how it works, it will explain how the system works in this study. The following image will explain the analysis of how the system works:

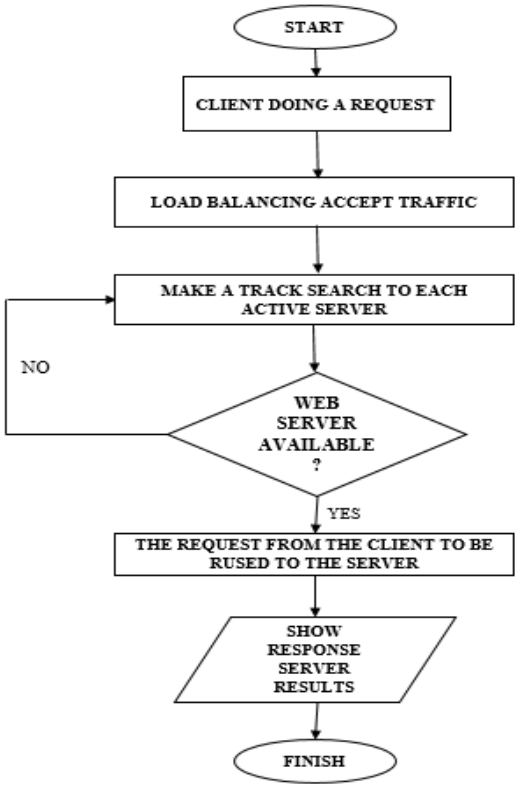

Fig 2. Analysis of how the system works

System design will be done by the client making a request to the load balancing system, then load balancing receives traffic from the client, after that load balancing searches the path to each active server based on the method, if the web server is available then the request from the client is forwarded to the server then it will display server response results.

3.2 Design

At this stage, a topology design for implementing load balancing on haproxy and router OS is carried out which is described as shown below:

a) Physical and logical topology design in haproxy load balancing implementation.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

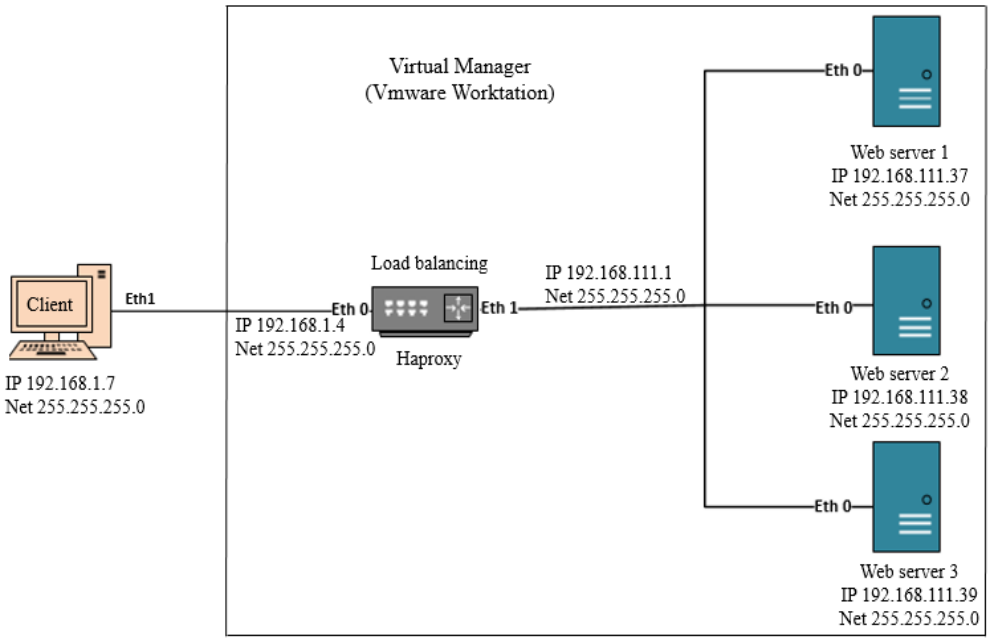

Fig 3. Physical and logical topology design in haproxy load balancing implementation.

From Figure 3 the PC client is physically connected directly to the haproxy load balancing system and three web servers built into the virtual machine, a logical topology for sharing IP and ethernet addresses, namely the client PC using eth1 with IP address addressing 192.168.1.7 network 255.255.255.0 , load balancing haproxy uses eth0 with IP addressing 192.168.1.4 network 255.255.255.0 and eth1 with IP addressing 192.168.111.1 network 255.255.255.0, then web server 1 uses eth0 with IP experience 192.168.111.37 network 255.255.255.0, web server 2 uses eth0 with IP addressing 192.168.111.38 network 255.255.255.0 and web server 3 using eth0 with IP addressing 192.168.111.39 network 255.255.255.0.

b) Physical and logical topology design on the implementation of load balancing router OS

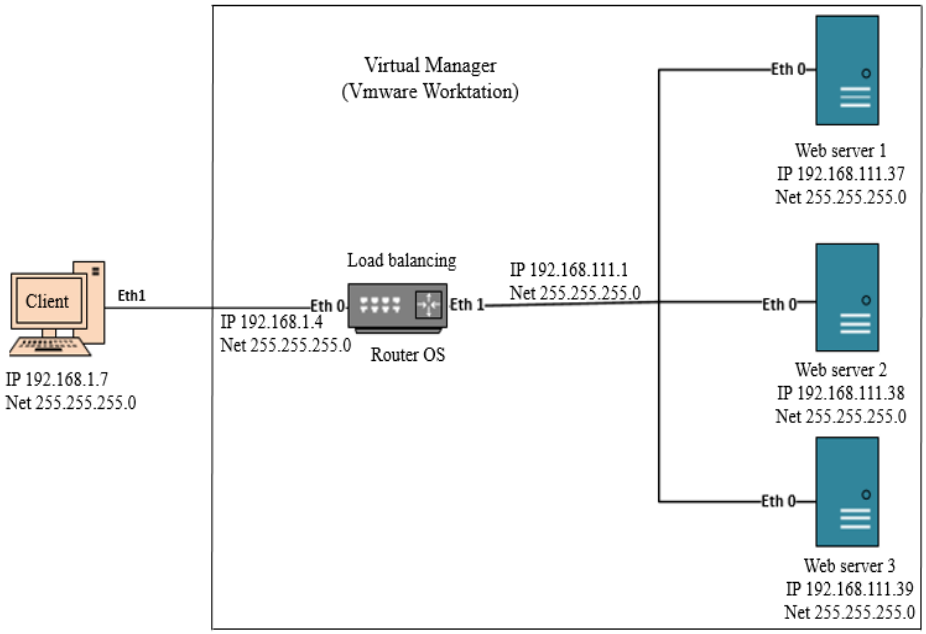

Fig 4. Physical and logical topology design on the implementation of load balancing router OS

From Figure 4, the client PC is physically connected directly to the router OS load balancing system and the three web servers built into the virtual machine, with a logical topology for sharing ip and ethernet addresses, that is, the client PC uses eth1 with IP address addressing 192.168.xx.xx network 255.255. 255.xx, load balancing router OS uses eth0 with ip addressing 192.168.xx.xx network 255.255.255.xx and eth1 with ip addressing 192.168.xx.xx network 255.255.255.xx, then web server 1 uses eth0 with ip experience

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

192.168.xx.xx network 255.255.255.xx, web server 2 using eth0 with ip addressing 192.168.xx.xx network 255.255.255.xx and web server 3 using eth0 with ip addressing 192.168.xx.xx network 255.255.255.xx.

3.3 Implementation

At this stage, implementation is carried out which is a continuation of the analysis and design stages. At this stage it consists of several parts.

a) Web Server Configuration.

Web server configuration is carried out as a marker of each web server 1, web server 2 and web server 3 when accessing IP load balancing on the browser, as shown below:

Fig 5. Web Server Configuration.

The result of the configuration above is by changing the default display on Apache2 with the name of each server, then the results are as shown below:

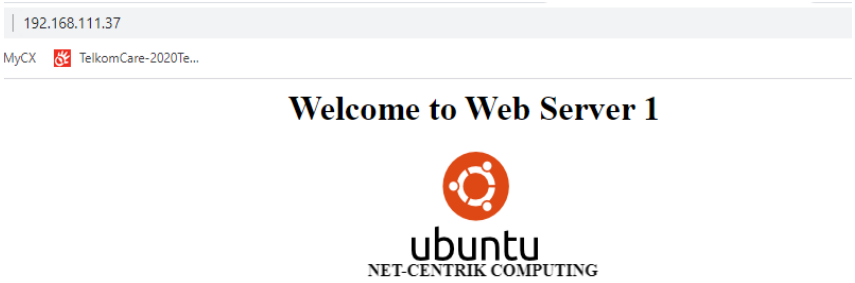

Fig 6. The results of the web server configuration

b) Haproxy configuration.

In the haproxy configuration, changing the IP forward to 1 is done to activate each load balancing method on haproxy with the nano /etc/sysctl.conf command, the results can be seen in the following figure:

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

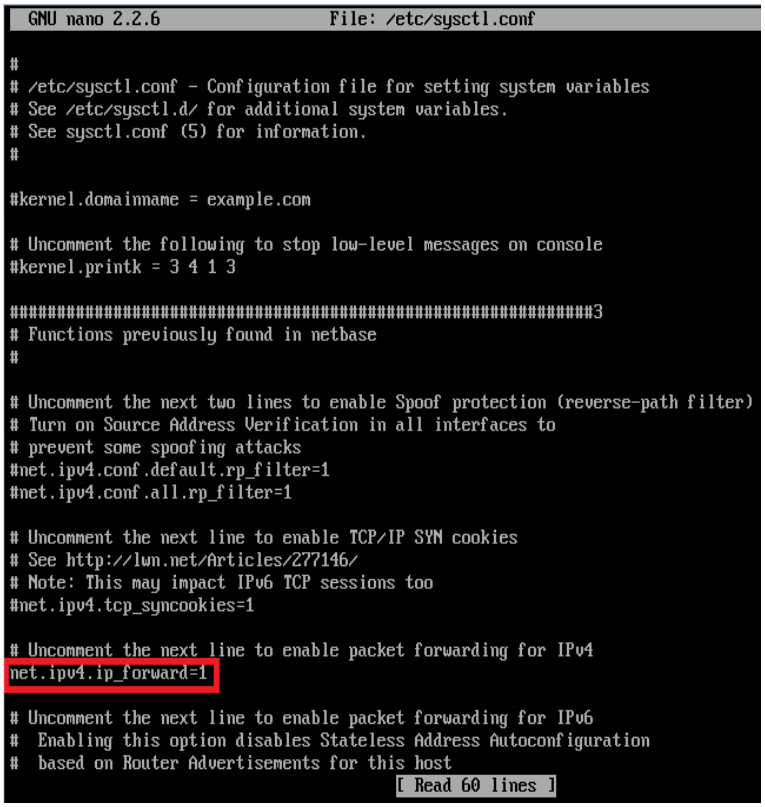

Fig 7. Haproxy configuration

c) Implement Configuration Each Method On Haproxy.

I configure each method that supports load balancing on haproxy as follows:

The round robin method configuration is performed to enable load balancing directed at the web server service with the command nano /etc/haproxy/haproxy.cfg, can be seen in the following figure.

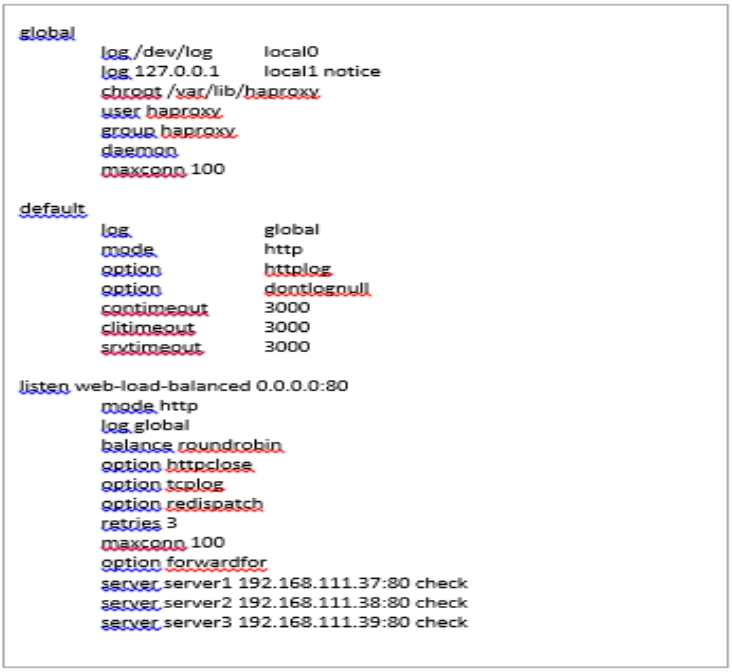

Fig 8. Configuration Round Robind

The least connection method is configured to enable load balancing on the web server service with the command nano /etc/haproxy/haproxy.cfg. The results can be seen as follows:

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

```
global
        log /dev/log    local0
        log 127.0.0.1   local1 notice
        chroot /var/lib/haproxy
        user haproxy
        group haproxy
        daemon
        maxconn 100
default
        log             global
        mode            http
        option          httplog
        option          dontlognull
        contimeout      3000
        clitimeout      3000
        srvtimeout      3000
listen web-load-balanced 0.0.0.0:80
        mode http
        log global
        balance leastconn
        option httpclose
        option tcplog
        option redispatch
        retries 3
        maxconn 100
        option forwardfor
        server server1 192.168.111.37:80 check
        server server2 192.168.111.38:80 check
        server server3 192.168.111.39:80 check
```

Fig 9. Result of Least Connection Method Configuration

3.4 Testing

The discussion of testing is to determine the results of the overall performance parameters for each load balancing method and without load balancing. The performance results are as in the following table:

Table 1
Overall Performance Results of Haproxy Load Balancing Methods and No Load Balancing

Measurement Factor	Testing scenario 1 with 10 users									
	Round Robin			Least Connection			Source			No load balancing
Throuhput (Kbps)	83			76			63			67
Delay (Ms)	0.21462			0.154385			0.43153			0.135662
Jitter (Ms)	0.42249			0.223409			0.43162			0.271364
	Server1	Server 2	Server 3	Server 1	Server 2	Server 3	Server 1	Server 2	Server 3	Server
Use CPU (%)	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	1.0	1.0
Use Memory (%)	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
Measurement Factor	Testing scenario 2 with 20 users									
	Round Robin			Least Connection			Source			No load balancing
Throuhput (Kbps)	125			172			134			155
Delay (Ms)	5.20988			5.10832			5.04929			3.5487
Jitter (Ms)	0.10415			0.10209			0.10092			7.09794
	Server1	Server 2	Server 3	Server 1	Server 2	Server 3	Server 1	Server 2	Server 3	Server
Use CPU (%)	0.7	0.7	0.7	0.7	0.7	0.7	0.3	0.3	1.3	1.7
Use Memory (%)	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.7	0.6
Measurement Factor	Testing scenario 3 with 30 users									
	Round Robin			Least Connection			Source			No load balancing
Throuhput (Kbps)	225			220			223			226

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Measurement Factor	Testing scenario 1 with 10 users									
	Round Robin			Least Connection			Source			No load balancing
Delay (Ms)	2.67626			2.60922			2.66928			1.72711
Jitter (Ms)	5.34832			5.21376			5.33157			3.44689
	Server1	Server 2	Server 3	Server 1	Server 2	Server 3	Server 1	Server 2	Server 3	Server
Use CPU (%)	1.0	1.0	1.0	1.0	1.0	1.0	0.3	0.3	1.3	2.3
Use Meory (%)	0.6	0.6	0.6	0.6	0.6	0.6	0.5	0.6	0.7	0.6

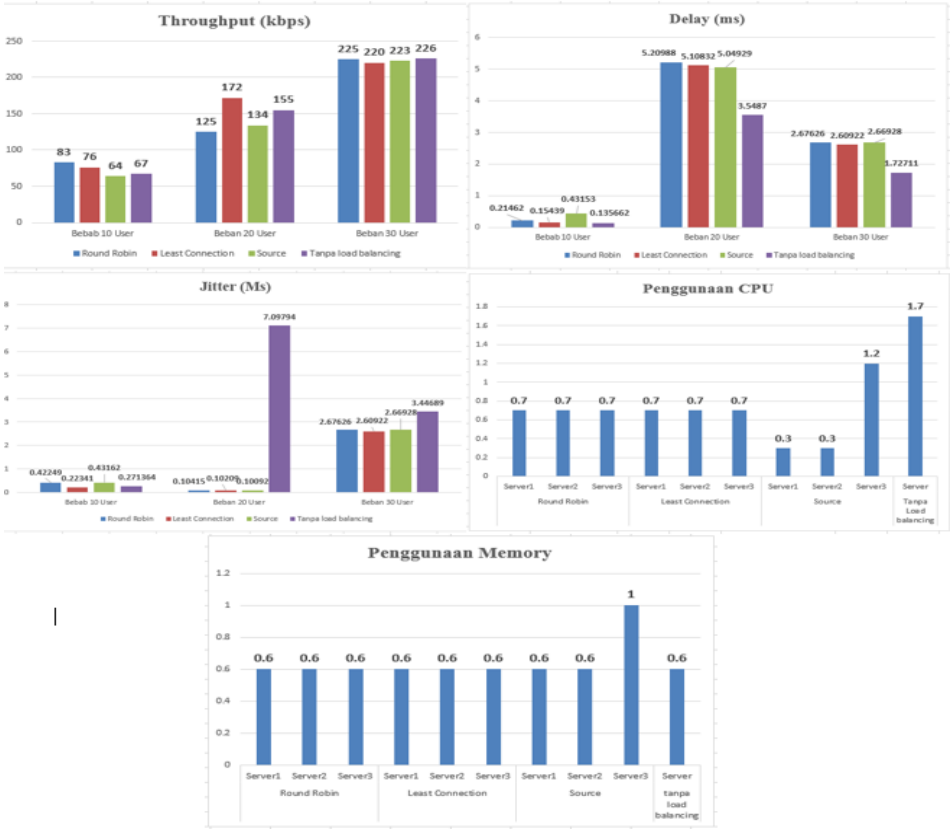

Fig 10. Results of Haproxy Load Balancing Overall Parameters and No Load Balancing

Table 2
Testing Scenario

Measurement Factor	Testing Scenario 1 dengan 10 User									
	NTH			PCC			ECMP			No balancing
Throuhput (Kbps)	105			71			224			67
Delay (Ms)	0.414304			0.886161			0.114634			0.135662
Jitter (Ms)	0.828721			0.017728			0.229094			0.271364
	Server1	Server 2	Server 3	Server 1	Server 2	Server 3	Server 1	Server 2	Server 3	Server
Use CPU (%)	0.3	0.3	0.3	0.3	0.3	0.3	0.7	0.3	0.3	1.0
Use Memory (%)	0.7	0.6	0.7	0.7	0.6	0.7	0.7	0.6	0.7	0.6
Measurement Factor	Testing Scenario 2 dengan 20 User									
	NTH			PCC			ECMP			No balancing
Throuhput (Kbps)	146			81			**272**			155
Delay (Ms)	0.176674			0.88315			3.66453			3.5487
Jitter (Ms)	0.353376			0.17664			0.70138			7.09794
	Server1	Server 2	Server 3	Server 1	Server 2	Server 3	Server 1	Server 2	Server 3	Server
Use CPU (%)	0.7	0.7	0.7	0.7	0.3	0.3	1.3	0.3	0.3	1.7
Use Memory	0.7	0.6	0.7	0.6	0.6	0.7	0.7	0.6	0.7	0.6

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

Measurement Factor	Testing Scenario 1 dengan 10 User									
	NTH			PCC			ECMP			No balancing
(%)										
Measurement Factor	Testing Scenario 3 dengan 30 User									
	NTH			PCC			ECMP			No load balancing
Throuhput (Kbps)	158			100			360			226
Delay (Ms)	2.92407			0.134239			1.66013			1.72711
Jitter (Ms)	5.84396			0.268315			3.31232			3.44689
	Server1	Server 2	Server 3	Server 1	Server 2	Server 3	Server 1	Server 2	Server 3	Server
Use CPU (%)	1.0	0.7	0.7	0.3	0.3	0.3	1.7	0.3	0.3	2.3
Use Memory (%)	0.7	0.6	0.7	0.6	0.7	0.7	0.7	0.6	0.6	0.6

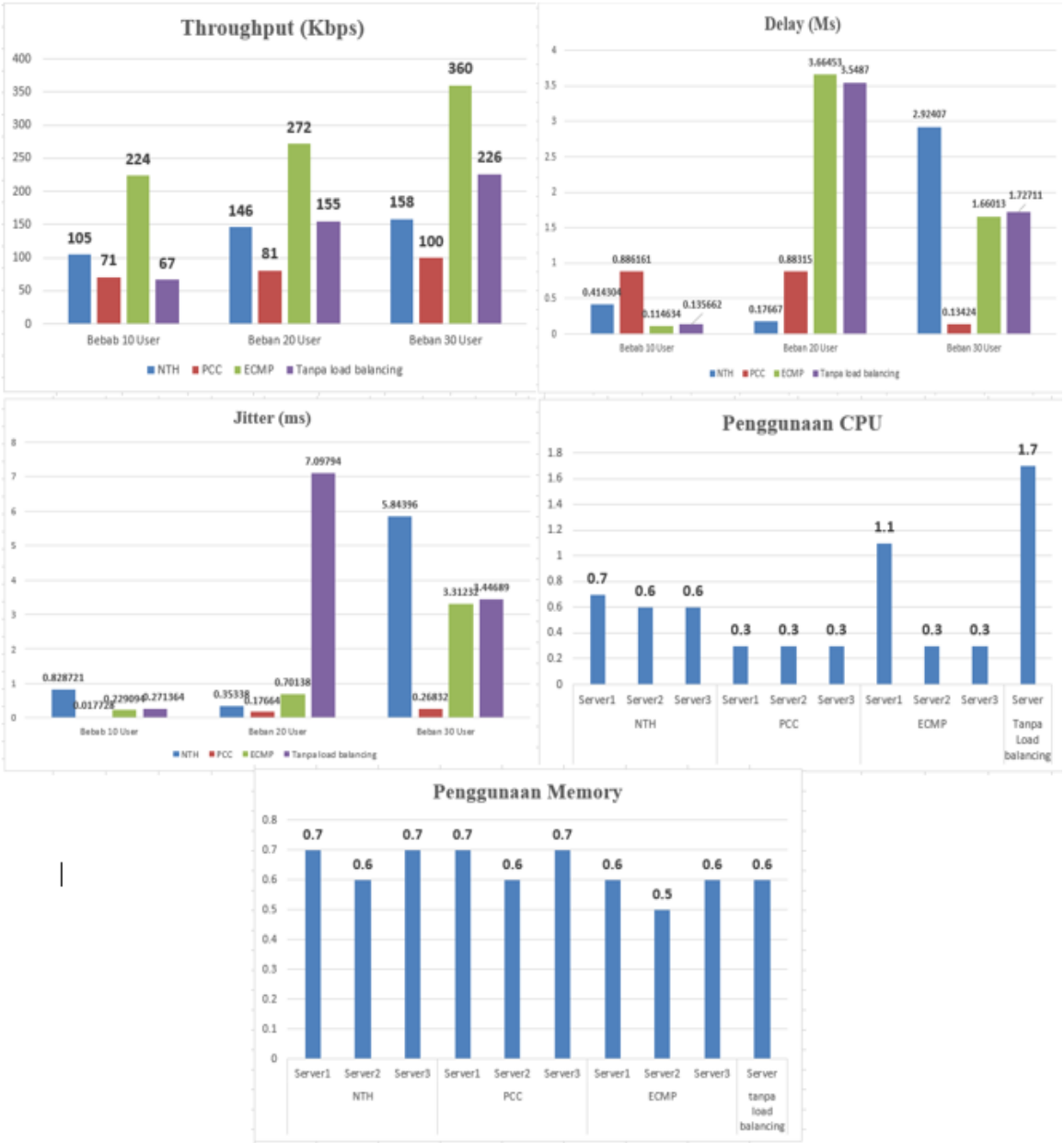

Fig 11. Results of Overall Router OS Load balancing Parameters and without Load balancing

The benefit of load balancing is to increase the speed of web access when opened, with two or more servers sharing web traffic, each of which will run faster because the load is not on one server alone but is divided into several servers, without load balancing in the division of work of the server is likely to

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

experience waiting or loading times because the load is borne by only one server and the CPU usage is quite large. From the hardware side, using one server, there is a possibility that the server will experience downtime or die, therefore, it is necessary to use more than one server and equipped with load balancing is very necessary so that work in the network continues. Load balancing has the ability to know whether a server can be used or not, a server that is down does not respond so that load balancing will divert work on other servers. Load balancing has a certain algorithm to share the work between servers with one another. For example, if there is a service request while server A is busy, work will be transferred to server B or C. There are also those who share the work of the server in turns. Each time there is a request, load balancing will point to server A, then B, and then C. All this is done so that no server has too large a workload.

4. Conclusion

Based on the research conducted, the results and discussion in the previous chapter, the following conclusions can be drawn: (1) The application of each load balancing method on haproxy and router OS can run well with each character of the method, and divides the load on three web servers. (2) Load balancing of the source method on haproxy is the same as the PCC method on the router OS, load balancing will go to the active web server first regularly except with a different IP, load balancing will lead to the next web server, this method is suitable for many client. The round robin method on haproxy is the same as the nth method on the OS router, load balancing will go to web server 1, web server 2, and web server 3 in turns, but the nth method is not sequential but random. The results of the performance measurement of the load balancing method on haproxy generally show that the least connection is better than the round robin and source methods, this can be seen from the measurement figures for delay and jitter parameters that are best found in the least connection method. While the results of the performance measurement of the load balancing method on the OS router, there are two, one shows that PCC is better, this can be seen from the measurement numbers of the delay and jitter parameters, then the second shows that the ECMP method is better, this can be seen from the throughput parameter measurement figures and delay.

5. References

[1] Nasser, Husain, Timotius Witono. 2016. Analisis Algoritma Round Robin , Least Connection , dan Ratio Pada Load Balancing Menngunkan Opnet Modeler. INFORMATIKA, 12(1), 1-8.

[2] Pramono, Luthan H, Robby Cokro B, dan ynuar Galih W. 2018. Round-roubin Algorithm in Haproxy and Nginx Load Balancing Performance Evaluation: Review. International Seminar On Research Of Information Teknology And intelegent Systems (ISRITI), 367-372.

[3] Nugroho, Agung, Widhi Yahya, dan Kasiful Amron. 2017. Analisis Perbandingan Performa Algoritma Round Robin dan Least Connection untuk Load Balancing pada Software Defined Network. Pengembangan Teknologi Informasi dan Ilmu Komputer, 1(12), 1-10.

[4] Triangga, Hasta, Ilham Faisal, dan Imran Lubis. 2019. Analisis Perbandingan Algoritma Static Round-Robin Dengan Least-Connection Terhadap Efisiensi Load Balancing Pada Load Balancer Haproxy.NASIONAL INFORMATIKA DAN TEKNOLOGI JARINGAN, 4(1), 1-6.

[5] Pratama, Ihsan, Dewi W S. 2019. Implementasi Server Load Balancing Menggunakan Haproxy dengan Algoritma Roundrobin Serta Penerapan Cache File. 1-7.

[6] Ritzkal. 2020. "Tick Waste Application in Houses With Warning of Microcontroller Assistant Social Media.," Jurnal MANTIK Vol 3, hlm. 559-568.

[7] Ritzkal. 2018. "Manajemen jaringan untuk pemula.," Bogor: UIKA PRESS.

[8] M Subchan, Dedi Setiadi. 2020. "Information System For Sale Of Muslim Clothes Based On E-Commerce Technology.," Jurnal MANTIK Vol 4, hlm. 311-318.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.