JOIN (Jurnal Online Informatika)
Volume 6 No.1 | June 2021: 103-112
DOI: 10.15575/join.v6i1.672

p-ISSN: 2528-1682
e-ISSN: 2527-9165

Computer Networks Optimization using Load Balancing Algorithms on the Citrix ADC Virtual Server

Hardiyan K. Ramadhan[1], Sukma Wardhana[2]
[1,2,]Informatics Engineering, School of Computer Science, Universitas Mercu Buana, Jakarta, Indonesia

Article Info

Article history:

Received December 06, 2020
Revised February 09, 2021
Accepted February 18, 2021
Published June 17, 2021

Keywords:

Citrix ADC VPX
Least connection
Least packet
Least response time
Round-robin

ABSTRACT

In the digital era and the outbreak of the COVID-19 pandemic, all activities are online. If the number of users accessing the server exceeds IT infrastructure, server down occurs. A load balancer device is required to share the traffic request load. This study compares four algorithms on Citrix ADC VPX load balancer: round-robin, least connection, least response time and least packet using GNS3. The results of testing response time and throughput parameters show that the least connection algorithm is superior. There were a 33% reduction in response time and a 53% increase in throughput. In the service hits parameter, the round-robin algorithm has the evenest traffic distribution. While least packet superior in CPU utilization with 76% reduction. So algorithm with the best response time and throughput is the least connection. The algorithm with the best service hits is round-robin. Large scale implementation is recommended using the least connection algorithm regarding response time and throughput. When emphasizing evenest distribution, use a round-robin algorithm.

Corresponding Author:

Hardiyan K. Ramadhan,
Informatics Engineering,
Universitas Mercu Buana,
Jl. Raya Meruya Selatan, Kembangan, Jakarta, 11650
Email: 41518320034@student.mercubuana.ac.id

1. INTRODUCTION

The increasing number of internet users must be balanced with the readiness of network and server infrastructure capacity. On the other hand, people want the maximum access speed [1]. The problem arises when the server's capabilities are unable to handle requests from the client. During the COVID-19 pandemic, all activities are currently carried out from home, lecturing activities. If, in terms of the e-learning infrastructure, the number of servers is not optimized, the e-learning server performance will decrease and even down.

A load balancer device is required to share the request traffic load from clients to the server. The load balancer device is divided into hardware (hardware) and software (software/appliance). Load balancers work by dividing the load request traffic from clients to the server. To be able to share the load traffic, the load balancer works using several algorithms. The algorithm is divided into two groups, namely Static Load Balancing Algorithm and Dynamic Load Balancing Algorithm [2]. According to Nguyen Xuan Phi in his research, 2017, some of the algorithms used in load balancers are Round-robin, Weighted Round Robin, Least Connection, Weighted Least Connection, and Least response time [3]. Algorithms included in the Static Load Balancing Algorithm include Round-robin, Central Manager, Randomized, and Threshold. Meanwhile, the algorithms that are included in the Dynamic Load Balancing Algorithm include Least connection, Least response time, Local Queue, and Central Queue [2].

The research will perform comparisons of the round-robin, least connection, least response time, and least packet algorithms by examining the aspects of network performance relevant to the server load balancing network when these three methods are implemented to obtain an optimal algorithm related to server load sharing according to project requirements.

Many previous studies have discussed the implementation of load balancing and algorithm comparisons in it. In the implementation of load balancing itself, various tools are used, such as HA Proxy [4], Nginx [5][6]. For the comparison of the algorithm itself, in previous studies that have been done, among others, the

comparison of the round-robin algorithm with the least connection [4] [7], the comparison of the round-robin algorithm, the least connection, and the ratio [8], the comparison of the round-robin, least packet and least connection [9]. From some of these studies, the researcher tries to raise the least response time algorithm as one of the algorithms tested for performance by being compared with round-robin and least connection algorithms then implemented in Citrix ADC VPX.

The test method will be carried out using GNS3 as the environment for simulation. With using five different scenarios with single webserver topology and multi-webserver, which consists of four webservers. In the multi-web server topology, the researcher focuses on finding out which algorithm has the best performance on response time, throughput, CPU utilization and service hits parameters by doing load testing in the GNS3 environment.

2. LITERATURE REVIEW

Load balancing is a technique for dividing the workload on two or more servers when a client requests a request. This aims to make traffic run optimally [10]. The load balancer is needed to do load balancing. The load balancer acts as a traffic cop positioned in the topology in front of the backend server farm.

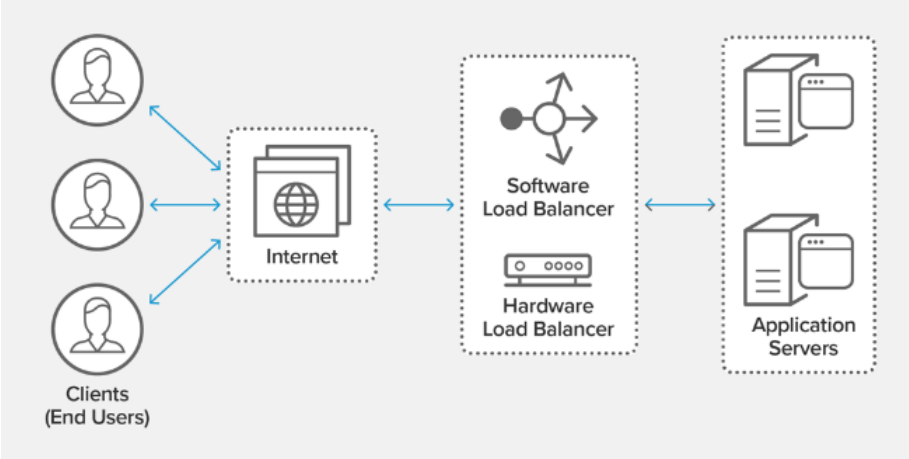

Figure 1. How the load balancer works

So the load balancer functions to efficiently distribute client or network requests to multiple servers as we can see at the Figure 1, ensure high availability and reliability by sending requests only to online servers, and provide the facility to increase or decrease the number of servers.

Citrix Application Delivery Controller (ADC) VPX provides complete, secure, secure, and remote access to web and application load balancing, acceleration, security, and offload features in a set of simple, easy-to-install virtual appliances. IT organizations, cloud, and telecommunications service providers can deploy Citrix ADC VPX to industry-standard hypervisors anytime, anywhere in a data center.

Citrix ADC VPX is included in ADC devices that work up to layer seven load balancing [11]. Because of its virtual form, it means reducing project and maintenance costs to increase the profit of companies using Citrix ADC VPX. Citrix ADC VPX can be installed on top of a hypervisor, or you can use Citrix ADX SDX.

The round-robin algorithm divides all server nodes to be treated equally according to the load assigned by each server but does not allow dynamic load switching due to the nature of static loads. There is no limit to the number of active servers on the backend [4]. The least response time uses the round-robin method plus a parameter to select the server with the least connection and shortest response time [3].The least connection works by sending a request to the server, where the selected server is the server with the least number of connections. The least packet algorithm distributes traffic or service, which currently is receiving the fewest packets. The load balancer will monitor the number of server connections and send requests to the server with the fewest connections [11].

Throughput is the actual measured bandwidth at a certain time in transmitting data or files. Throughput describes the actual bandwidth at a time and in certain conditions and networks used to download a certain size file [12]. This research will use the Response Time using the standards from TIPHON (Telecommunications and Internet Protocol Harmonization Over Network) and included in the standardization delay refer to the Table 1. The delay parameter value is obtained by managing the response time, which consists of average, minimum, and maximum [13].

Table 1. Standardization of delay according to tiphon

Delay Category	Delay (ms)
Excellent	< 150
Good	150 – 300
Moderate	300 - 450
Poor	> 450

Apache JMeter is an open-source, 100% pure Java application designed to load functional behavior tests and measure performance. It was initially designed for testing Web Applications but has since been extended to other test functions [14]. Apache Jmeter will be used to measure the performance of the CPU utilization parameter on the webserver.

Apache Benchmark is a tool from the Apache organization used to measure performance on a Hypertext Transfer Protocol (HTTP) webserver. This tool is used to calculate how many requests per second the webserver used can serve. Some of the features of the Apache Benchmark are: open source, simple command line, platform-independent, load, and performance tests, not extensible [15]. Apache Benchmark will measure the performance of the response time parameters and the throughput of the webserver.

The webserver is software that serves HTTP requests, and the web browser also sends dynamic codes to the application server. This application server translates and processes dynamic codes into static HTML codes in a static page, sent to the browser by the webserver [10].

3. TEST DESIGN

To determine the webserver's performance before and after optimization will use two topologies: single webserver topology and multi-webserver topology with the load balancer. Both topologies will be simulated in Graphical Network Simulator 3 (GNS3) software. GNS3, at a glance, is a network simulation software that was first released in 2008. It can combine native and virtual devices for simulating complex networks. GNS3 is widely used by many companies such as Exxon, Walmart, AT&T, and NASA. Specifications for laptops that run GNS3 devices and as testing laptops or refer to the Table 2.

Table 2. Laptop specification

Category	Specification	Description
Hardware	System Manufacturer	Lenovo
	System Model	Thinkpad L540
	Processor	Intel® Core™ i5-4300M 2.6 GHz
	Operating System	Xubuntu Linux 16.04
	Memory	16 GB
	Storage Media	HDD 500 GB
Software	Network Simulator	GNS3 2.2.1
	HTTP Load Testing	Apache Jmeter and Apache Benchmark

The following single webserver topology will be simulated:

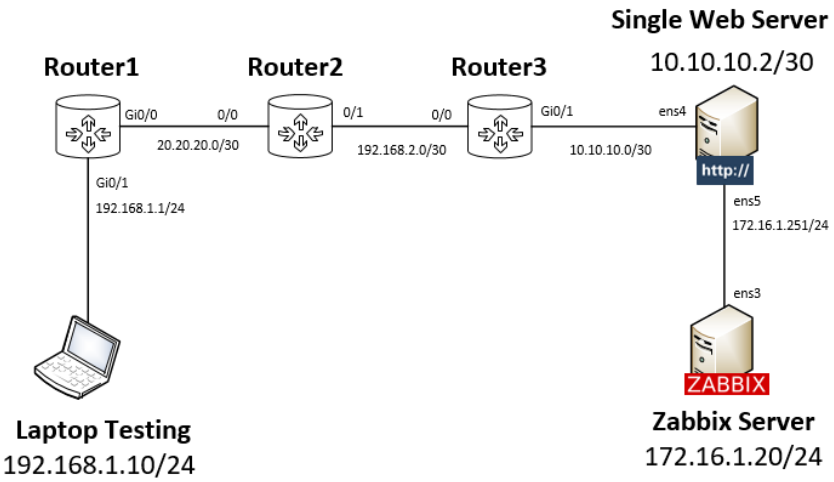

Figure 2. Single webserver topology

Figure 2 shows the network devices used are router, webserver, client, and NMS server. Meanwhile, for a multi-webserver topology, the following will be simulated:

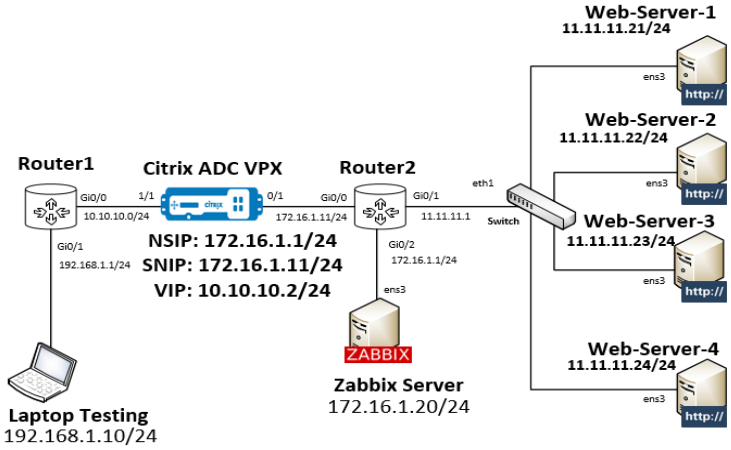

Figure 3. Multi-webserver topology

Figure 3 shows the network devices used are router, load balancer, webserver, client, and NMS server. The following load balancer device specifications refer to Table 3.

Table 3. Load balancer specification

Type	Specification
vCPUs	2
RAM	4096 MB
Network adapter	Paravirtualized Network I/O
Disk image	NSVPX-KVM-12.0-56.20_nc_32.qcow2
Disk interface	Virtio
Brand	Citrix ADC VPX
Version	12

Performance measurement of the four algorithms (round-robin, least connection, least response time, and least packet) is done by measuring several parameters: response time, throughput, service hits, and CPU utilization. Later, HTTP requests from Laptop Testing will be made using the Apache Benchmark software and for CPU utilization testing using Apache Jmeter. Service hits are obtained from the results of the parameters of the four algorithms that have been implemented alternately (round-robin, least connection, least response time, and least packet), which appear on the Citrix ADC VPX virtual server load balancing statistics dashboard.

There are five experimental scenarios used as referred to the Table 4. The first is to test the traffic load on a single webserver topology, the second to test the traffic load on a multi-webserver topology combined with a load balancer device using a round-robin algorithm, the third change the algorithm to the least connection, the fourth using least response time, and the last using least packet. The number of users charged in the four scenarios is 3k, 6k, 9k, and 12k and uses level 10 concurrency.

Table 4. Experiment scenario

Scenario	Total webserver	Algorithm	Total user
1	1	-	3k, 6k, 9k and 12k
2	4	Round-robin	3k, 6k, 9k and 12k
3	4	Least connection	3k, 6k, 9k and 12k
4	4	Least response time	3k, 6k, 9k and 12k
5	4	Least packet	3k, 6k, 9k and 12k

In the first scenario, the researcher only needs one web server. Even if we use four web servers, a load balancer is still required to distribute traffic, if there is no load balancer, traffic distributed on one webserver only. The first scenario will test the performance of a single webserver without using a load balancer. The

number of users tested was 3k, 6k, 9k, and 12k. The first parameters tested are response time and throughput using Apache Benchmark. In the Figure 4, the test load request is 12k with concurrency level 10 and 51.96 requests per second. The command used is "ab -n 12000 -c 10 http://10.10.10.2/".

```
Server Software:        Apache/2.4.29
Server Hostname:        10.10.10.2
Server Port:            80

Document Path:          /
Document Length:        834 bytes

Concurrency Level:      10
Time taken for tests:   230.969 seconds
Complete requests:      12000
Failed requests:        33
   (Connect: 0, Receive: 0, Length: 33, Exceptions: 0)
Keep-Alive requests:    11866
Total transferred:      13649938 bytes
HTML transferred:       9980478 bytes
Requests per second:    51.96 [#/sec] (mean)
Time per request:       192.474 [ms] (mean)
Time per request:       19.247 [ms] (mean, across all concurrent requests)
Transfer rate:          57.71 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0  16.2      0    1026
Processing:     0  192 455.5      8    6520
Waiting:        0  192 455.5      7    6520
Total:          0  192 455.7      8    6520
```

Figure 4. Apache Benchmark test results 12k users on a single webserver topology

Next is the CPU utilization test using Apache Jmeter as we can see at the Figure 5. Using 12k users and a ramp-up period of 3 minutes.

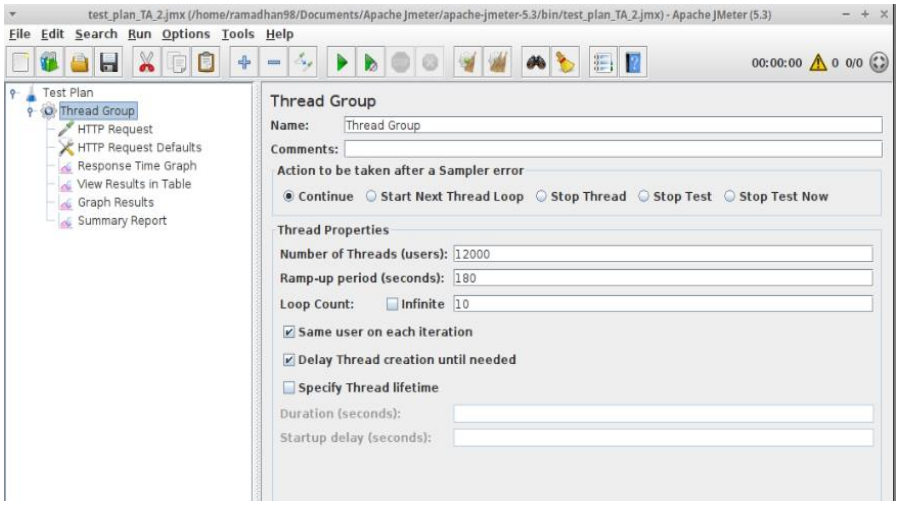

Figure 5. Testing the CPU utilization using Apache Jmeter

The results of CPU utilization with 12k users on a single webserver topology are seen from Zabbix as referred to the Figure 6. The maximum value of CPU utilization is 8.3135%.

Figure 6. Graph of CPU utilization topology single webserver with 12k users

The second scenario tests multiple webserver performance with a round-robin algorithm on the Citrix ADC VPX load balancer. The number of users tested was 3k, 6k, 9k, and 12k. The first parameters tested are response time and throughput using Apache Benchmark. In Figure 7, the test load request is 12K with concurrency level 10 and 75.17 requests per second. The command used is "ab -n 12000 -c 10 http://10.10.10.2/".

```
Server Software:        Apache/2.4.29
Server Hostname:        10.10.10.2
Server Port:            80

Document Path:          /
Document Length:        864 bytes

Concurrency Level:      10
Time taken for tests:   159.628 seconds
Complete requests:      12000
Failed requests:        0
Total transferred:      14048598 bytes
HTML transferred:       10368000 bytes
Requests per second:    75.17 [#/sec] (mean)
Time per request:       133.024 [ms] (mean)
Time per request:       13.302 [ms] (mean, across all concurrent requests)
Transfer rate:          85.95 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    6  69.7      1    1031
Processing:     1  123 530.2      2   30140
Waiting:        1  123 530.2      2   30140
Total:          2  129 534.1      3   30141
```

Figure 7. 12k user Apache Benchmark test results with a round-robin algorithm

The Apache Benchmark test results showed 13.302 ms for the response time value and 85.95 Kbps for the throughput value as we can see at Figure 7. Next is the CPU utilization test using Apache Jmeter and 12k users and a ramp-up period of 3 minutes. The CPU utilization results with 12k users on a multi-web server topology with a round-robin algorithm are seen from Zabbix, please refer to the Figure 8. The maximum value of CPU utilization is 2.4485%.

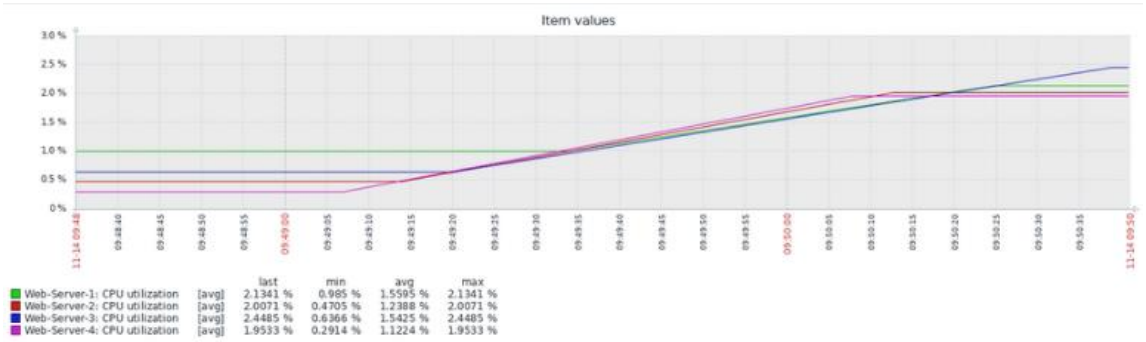

Figure 8. 12k user CPU utilization chart with round-robin algorithm

Meanwhile, for service hits on the dashboard statistics, load balancing virtual server Citrix ADC VPX round-robin algorithm refers to Figure 9 below.

Bound Service(s) Summary

Name	IP address	Port	Service type	State	Service hits
Service-1	11.11.11.21	80	HTTP	UP	3,000
Service-2	11.11.11.22	80	HTTP	UP	3,000
Service-3	11.11.11.23	80	HTTP	UP	3,000
Service-4	11.11.11.24	80	HTTP	UP	3,000

Figure 9. Service hits 12k user round-robin

Figure 9 shows that the distribution of traffic in the round-robin algorithm is evenly distributed from 12k traffic divided into 3k traffic each. The third untill fifth scenario tests multi-web servers performance with the least connection, least response time and least packet algorithm on the Citrix ADC VPX with the same method.

4. RESULT AND DISCUSSION

After testing five scenarios using the parameters of Response Time, Throughput, and CPU Utilization, here are the results of comparing the five scenarios.

RESPONSE TIME

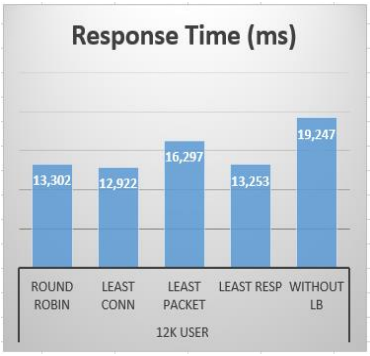

Figure 10. 12k user response time comparison graph

In testing the response time parameters, it was carried out on 3k, 6k, 9k, and 12k users. The response time comparison graph for 12k user can refer to the Figure 10. For details as follows.

Table 5. Response time comparison results

Algorithm	Load Test	Response time (ms)	Load Test	Response time (ms)
Round-robin	3k user	13,649	9k user	13,513
Least connection		13,548		13,041
Least response time		14,45		14,154
Least packet		15,309		15,621
Without LB		19,072		19,15
Round-robin	6k user	13,701	12k user	13,302
Least connection		13,449		12,922
Least response time		15,047		16,297
Least packet		15,642		13,253
Without LB		19,146		19,247

From Table 5, it can be seen that there is an optimization of response time after using a load balancer. Meanwhile, the least connection algorithm has the best response time (lowest in ms units).

THROUGHPUT

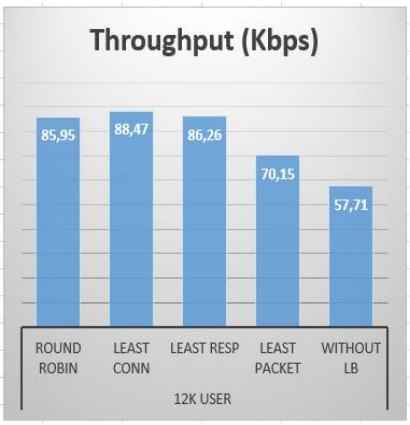

Figure 11. 12k User throughput comparison graph

In testing the throughput parameters, it is carried out on 3k, 6k, 9k, and 12k users. The throughput comparison graph for 12k user can refer to the Figure 11. For the details as follows.

e-ISSN 2528-1682
p-ISSN 2527-9165

Table 6. Throughput comparison results

Algorithm	Load Test	Throughput (Kbps)	Load Test	Throughput (Kbps)
Round-robin	3k user	83,77	9k user	84,61
Least connection		84,39		87,66
Least response time		79,12		80,77
Least packet		74,68		73,19
Without LB		58,39		58,06
Round-robin	6k user	83,44	12k user	85,95
Least connection		85,01		88,47
Least response time		75,98		86,26
Least packet		73,09		70,15
Without LB		58,12		57,51

Table 6 shows throughput optimization after using a load balancer, while the least connection algorithm has the best throughput (highest in Kbps units).

CPU UTILIZATION

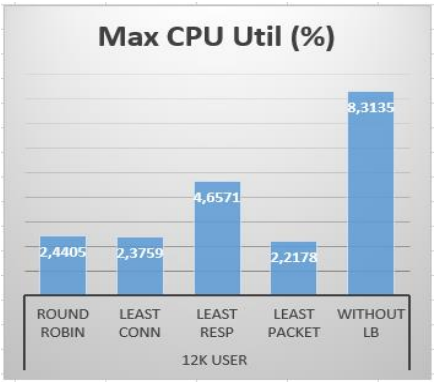

Figure 12. 12k User max CPU util comparison graph

In testing the CPU utilization parameter, it was carried out at 3k, 6k, 9k, and 12k users. The CPU utilization comparison graph for 12k user can refer to the Figure 12. For the details regarding CPU utilization please refer to the Table 7 below.

Table 7. Max CPU utilization comparison results

Algorithm	Load Test	Max CPU util (%)	Load Test	Max CPU util (%)
Round-robin	3k user	1,3724	9k user	2,3027
Least connection		1,2765		1,8411
Least response time		1,4042		2,6191
Least packet		1,1228		1,7222
Without LB		4,6858		7,2668
Round-robin	6k user	1,7452	12k user	2,4405
Least connection		1,3758		2,3759
Least response time		2,9082		4,6571
Least packet		2,025		2,2178
Without LB		5,339		8,3135

In the era of cloud computing, the industry is required to use hardware optimally. The higher CPU utilization on a server, the server will overload. In the CPU utilization parameter, the algorithm with the lowest CPU utilization is the least packet algorithm. When choosing the least packet algorithm, the CPU utilization on the virtualization server hardware can accommodate an optimal number of virtual machines.

SERVICE HITS

In the service hits parameter, traffic distribution results are displayed on the Citrix ADC VPX dashboard from the testing laptop to the webserver. From here, we will see the algorithm's behavior in dividing traffic.

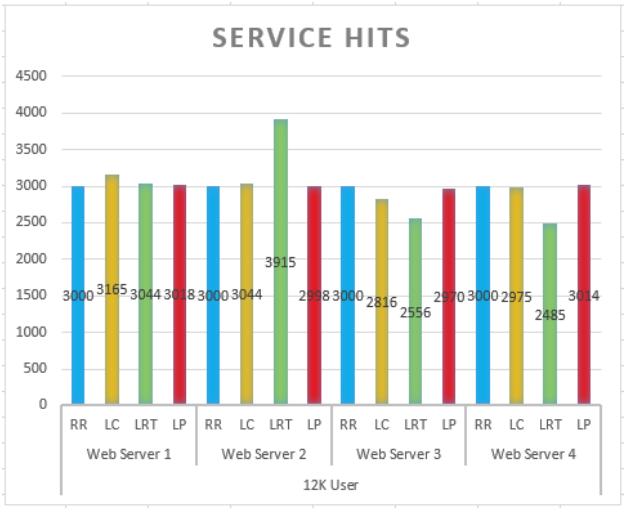

Figure 13. Service comparison chart hits 12k users

In Figure 13, the traffic distribution on the round-robin algorithm is even. This algorithm rotates the processes that are in the queue, so all servers get the same number of hits. The other algorithms such as least response time and least connection share traffic based on the lowest active connection, so traffic distribution sometimes uneven.

5. CONCLUSION

Each algorithm has its own characteristics and advantages based on response time, throughput, CPU utilization, and service hits parameters. According to the two parameters tested, the algorithm with the best performance was the least connection with a 33% decrease in response time and an increase in throughput of 53%. The best CPU utilization parameter is in the least packet algorithm, with a decrease of 76%. In contrast, for the service hits parameter, the round-robin algorithm has an even distribution of traffic than other algorithms. When viewed from the performance of the response time and throughput parameters, the least packet algorithm is in the lowest position compared to the other three algorithms. Tests that use four numbers of users, respectively 3k, 6k, 9k, and 12k, produce consistent results with round-robin algorithms that are always even in distributing traffic. At the same time, the least connection is best in response time and throughput performance. For CPU utilization, the best is in the least packet algorithm. In large-scale implementations, if there is a need to distribute traffic to all existing back-end servers evenly, it is recommended to use a round-robin algorithm. In contrast, the need to focus on response time and throughput parameters is recommended to use the least connection. Finally, when focusing on CPU utilization, use the least packet algorithm.

For further research, it is recommended to conduct research in a production environment and focus on the latest technologies, such as load balancing in cloud computing and micro-services because the current industry trend has changed from previously hardware-based to software-based (NFV and SDN).

6. REFERENCES

[1] F. P. Perdana, B. Irawan, R. Latuconsina, F. Teknik, U. Telkom, and L. Balancing, "Analisis Performansi Load Balancing Dengan Algoritma Weighted Round Robin Pada Software Defined Network (SDN)," *e-Proceeding Eng.*, vol. 4, no. 3, pp. 4161–4168, 2017.

[2] P. Beniwal, "A comparative study of static and dynamic Load Balancing Algorithms," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 2, no. 12, pp. 386–392, 2014.

[3] N. Xuan Phi and T. Cong Hung, "Load Balancing Algorithm to Improve Response Time on Cloud Computing," *Int. J. Cloud Comput. Serv. Archit.*, vol. 7, no. 6, pp. 01–12, 2017, doi: 10.5121/ijccsa.2017.7601.

[4] H. Triangga, I. Faisal, and I. Lubis, "Analisis Perbandingan Algoritma Static Round-Robin dengan Least-Connection Terhadap Efisiensi Load Balancing pada Load Balancer Haproxy," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 4, no. 1, pp. 70–75, 2019, doi: 10.30743/infotekjar.v4i1.1688.

[5] W. Chen, A. Noertjahyana, and J. Andjarwirawan, "Analisis Perbandingan Kinerja Algoritma Load Balancer

NGINX pada Studi Kasus PRS," *J. Infra,* vol. 7, no. 2, pp. 60–64, 2019.

[6] D. K. Hakim, D. Y. Yulianto, and A. Fauzan, "Pengujian Algoritma Load Balancing pada Web Server Menggunakan NGINX," *JRST (Jurnal Ris. Sains dan Teknol.*, vol. 3, no. 2, p. 85, 2019, doi: 10.30595/jrst.v3i2.5165.

[7] A. Nugroho, W. Yahya, and Kasyful Amron, "Analisis Perbandingan Performa Algoritma Round Robin dan Least Connection untuk Load Balancing pada Software Defined Network," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1568–1577, 2017, [Online]. Available: http://j-ptiik.ub.ac.id.

[8] H. Nasser and T. Witono, "Analisis Algoritma Round Robin, Least Connection, Dan Ratio Pada Load Balancing Menggunakan Opnet Modeler," *J. Inform.*, vol. 12, no. 1, pp. 25–32, 2016, doi: 10.21460/inf.2016.121.455.

[9] Y. K. Ningsih and N. Norman, "KLUSTER DNS DALAM TELEKOMUNIKASI UNTUK MANAJEMEN SKALABILITAS," *J. Elektro,* vol. 11, no. 2, pp. 93–100, 2018.

[10] D. Rahmana, R. Primananda, and W. Yahya, "Analisis Load Balancing Pada Web Server Menggunakan Algoritme Weighted Least Connection," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 3, pp. 915–920, 2017, doi: 10.1016/0028-3932(72)90008-5.

[11] I. Ivanisenko, "Methods and Algorithms of Load Balancing," *Int. J. "Information Technol. Knowledge,"* vol. 9, no. 4, pp. 340–375, 2015.

[12] Hasanul Fahmi, "Analisis Qos (Quality of Service) Pengukuran Delay, Jitter, Packet Lost Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik," *J. Teknol. Inf. dan Komun.*, vol. 7, no. 2, pp. 98–105, 2018.

[13] M. F. W. Simanjuntak, O. D. Nurhayati, and E. D. Widianto, "Analisis Quality of Service (QoS) Jaringan Telekomunikasi High-Speed Downlink Packet Access (HSDPA) pada Teknologi 3.5G," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 1, p. 67, 2016, doi: 10.14710/jtsiskom.4.1.2016.67-76.

[14] D. I. Permatasari, "Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian," *J. Sist. dan Teknol. Inf.*, vol. 8, no. 1, p. 135, 2020, doi: 10.26418/justin.v8i1.34452.

[15] A. Y. Chandra, "Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request," *J. Sist. dan Inform.*, vol. 14, no. 1, pp. 48–56, 2019, doi: 10.30864/jsi.v14i1.248.