JURNAL TEKNOLOGI DAN OPEN SOURCE
Vol. 4, No. 1, June 2021, pp. 08~18
e-ISSN: 2622-1659, accredited Four Grade by Kemenristekdikti, Decree No: 36/E/KPT/2019
DOI: 10.36378/jtos.v4i1. 936

HIGH AVAILABILITY IN SERVER CLUSTERS BY USING BACKPROPAGATION NEURAL NETWORK METHOD

Ahmad Heryanto[1], Aditya Gunanta[2]
[1]Laboratorium Jaringan Komputer dan Komunikasi Data, Universitas Sriwijaya
[1,2]Sistem Komputer, Universitas Sriwijaya

Article Info

Article history:

Received 30 November, 2020
Revised 27 May, 2021
Accepted 09 June, 2021

Keywords:

Client,
Server,
Kluster,
High Availability,
Neural network

ABSTRACT

Server is a host device applications to serve every request in finding information needs. The server must fully support the services used for the organization's digital needs 24 hours in a day, 7 days in a week, and 365 days in a year. The concept of High Availability is needed to maintain the quality of server services. The algorithm used to build HA can use both classical and modern algorithms. The algorithm used in this research is using backpropagation neural network. In this study, the parameter values to obtain optimal accuracy are learning rate 0.1, training data 80 and test data 20, the number of nodes in hidden layer 4, minimum error 0.0001, and the number of iterations 2500.The best accuracy value using these parameters is 93.79% .

Corresponding Author:

Ahmad Heryanto
Sistem Komputer, Universitas Sriwijaya
Palembang, Indonesia
Email: hery@unsri.ac.id

1. Introduction

Client-Server technology in the era of industrial technology 4.0 is inevitable service. The services based on server side be a main point for the company to operate the organization. The server is a host device applications to serve every request in finding needs [1]. The server must always be available and ready to serve requests to get the contents. The server must be fully support the services used for the organization's digital needs 24 hours in a day, 7 days in a week, and 365 days in a year. The running server continuously will have problems. The problem might be occur are excessive traffic, request flooding, data center downs, hacker attacks, power outages, natural disasters, and single point of failure[2]–[4]. High Availability is the quality of service to continuously operate without failing for a long period even though there are disruptions threaten the available services [1]. High Availability can use hardware or software, and one way to get High Availability on a service is clustering [1]. In computing, a clustering system consists of two or more computers (nodes) that work together to perform computational tasks.

Algorithms commonly used for High Availability on server computer clusters are round robin, static-rr, wight rr, lastconn, and source[2], [5]–[7]. This type of algorithm in the scope of the data center is classic algorithm. Classical algorithms generally have simple, fast and easy ways to implement. However, there are many weaknesses in this classical algorithm, one of them is it does not use computational parameters as a whole because usually this algorithm only uses 1 type of parameter as input in making decisions on computer clusters in choosing nodes to serve clients. Parameters in classical algorithm can be in process, source ip, load process, and cookies. Therefore, more complex modern algorithm is needed to analyze the overall server node parameters such as processor, memory, i / o, software process, disk, service limit, and operating system. In this study it used backpropagation artificial neural network. HA system used backpropagation artificial neural network algorithm will conduct training data consisting of multiple inputs, and produce targeted output [8].

In the training process there are two main steps, there are forward pass and backforward. At the forward pass, the input will be propagated to the output layer and the predicted output will be compared with the target by using Loss Function. Loss function is used to measure the performance of the neural network in predicting selected server nodes to serve client requests. The next process of artificial neural network backpropagation is the backward pass. It will adjust the weight and bias based on the errors obtained during the forward pass. So this algorithm is expected to be able to predict the best server node to serve content requests from clients in the server cluster.

2. Research Method

2.1. Server Cluster

Cluster is collection of computer algorithms can complete a job in parallel. Server clusters consist of a group of identical computers and work together to get great capabilities by combining server computers. Clustering on servers can be done by grouping two or more servers into cluster to reduce the computational load. On a server with cluster system, visitors and their requests directed to the appropriate server according to traffic type, location, server load, available network bandwidth and other system indicator measurements dinamically [9].

There are two parts of Server cluster system, namely Manager cluster (Master Node) and the node cluster (Backend). The master computer has role in distributing work to configuration worker computers as the backend. The backend computer works based on type of clustering and algorithm used. There are several types of clusters are often used, namely: Shared Processing, Load balancing, failover, High Availability. Clustering techniques are carried out to support the system in achieving high service availability [10]. It’s same opinion with in this research [11], that the objectives of clustering are:
1. High availability
Computer clusters can provide service availability.
2. Failover Server
Reduce system failure by managing failover.
3. Scalability
Additional hardware and software can be dynamically upgraded to improve system performance.

2.2. Load Balancing

System of load balancing is one of the main elements in the Server cluster system. Load balancing is the ability of the master node on a computer cluster to distribute the computational load to server nodes located in the backend area of the cluster. The master node transfers computational data traffic between several server nodes using a network-based infrastructure. This load balancing process is transparent towoards users who make requests to the server load balancer. Then [12] explained that there are five functions of a load balancer, namely:
1. Intercept network-based traffic such as web traffic destined for an address.
2. Divide traffic into request units and decide which server will accept the request.
3. Monitore the servers to ensure these Servers can respond the traffic.
4. Provide redundancy by using more than one failover scenario.
5. Provide supervision in content distribution (in network traffic), such as reading URLs, intercepting cookies, and XML parsing.

The load balancing system supports various network protocols, such as protocols based on TCP or UDP. This makes load balancing systems flexible and simple. The load balancing network topology is shown in the following Figure 1.

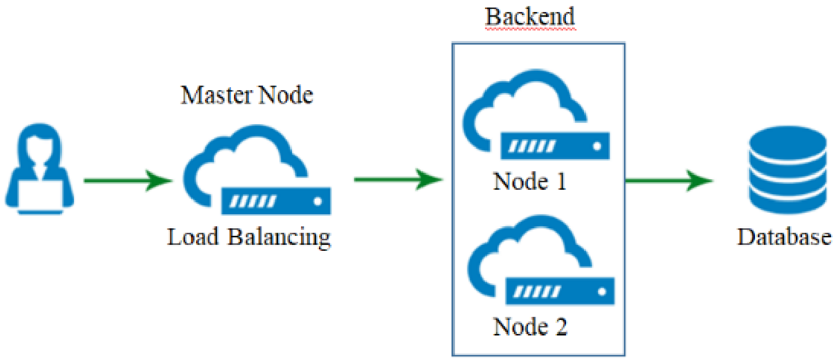


Figure 1. Load balancing Server

2.3. Artificial Neural Network (ANN)

In 1940, scientists found the psychology of the brain is the same way computers work. Humans have brains with extraordinary abilities. The structure of the human brain consists of complex neural elements. In general, the structure of the brain consists of neurons connected by synapses. The relationship between neurons and synapses is shown in the following picture.

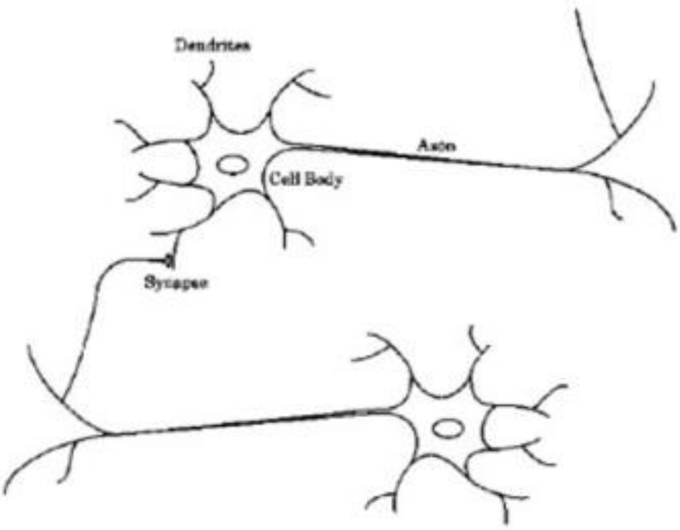


Figure 2. Simple neuron structure

Based on the neuron picture above, neurons have 3 important components, they are: dendrites, synapses and axons. Dendrites function as signal receivers from other neurons. The signal is an electrical impulse to send through the synaptic cleft. Synapses function as the sum of all incoming signals. If the sum is strong and exceeds the threshold, the signal will be transmitted to other cells via axons. It is estimated that humans have 1012 neurons that connected to neural network in humans. The neural network of the brain is able to recognize patterns, perform calculations, and control the work of the body's organs highly speed.

Based on the workings of the human brain neural network, scientists in the field of computing have developed an artificial intelligence algorithm, as known as an artificial neural network (ANN). ANN algorithms has been used in various fields of computing. ANN is determined by 3 things, namely: the relationship between neurons (network architecture), the method for assigning connecting weights (called the training / learning / algorithm method), and the activation function. The advantages of algorithms with ANN include:
a. Able to generalize and extract from a certain data pattern based on historical data in input-output process.
b. ANN can develop learning abilities (self organizing).

c. Has fault tolerance to interference.
d. Able to perform calculations in parallel.

The reason for choosing the artificial neural network (ANN) method can complete functions with more complex input and output parameters. Whereas in the previous research using the round roubin, leastconn, rr algorithm was only able to solve one input and one output.

A round roubin program, leastconn, rr, with one input and one output has the following function:

$$y = f(x)$$

ANN, for example the input parameters A, B, C, and D and the output parameters x and y, then the function becomes:

$$(xy) = f(a, b, c, d)$$

If the above equation is used using round roubin, lastconn, and rr, the results will be difficult to produce. Therefore, ANN is the right method to solve this problem.

2.3.1 Basic Principles of ANN
Generally, the artificial neural network will undergo training, so the particular input will lead to specific target output. This network will be adjusted based on the comparison of all outputs until the output network matches the target. The basic principles of ANN are described in the following.

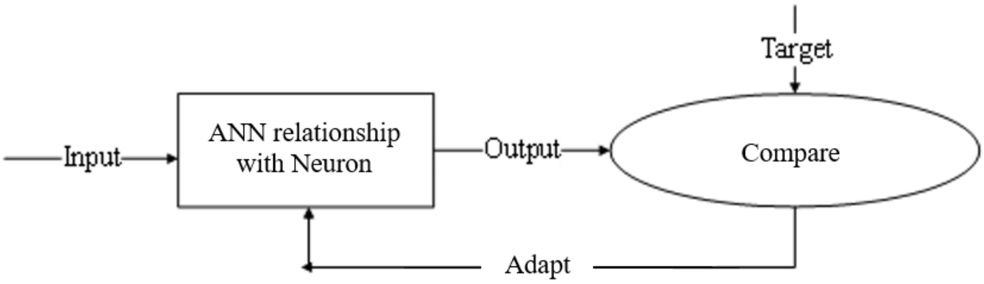


Figure 3. Basic principles of ANN

2.3.2 Network Architecture
The neurons in the ANN algorithm network form a specific network structure. In ANN, the network architecture that is commonly used is divided into two, namely single layer network and multi layer network.
a. Single layer network.
In a single layer network architecture, a set of input neurons is connected directly to a set of outputs. In this network architecture, all input units are connected directly to all target outputs, although with different weights. In this network no input unit is connected to other input units. Figure 4 below is a Single layer network architecture.

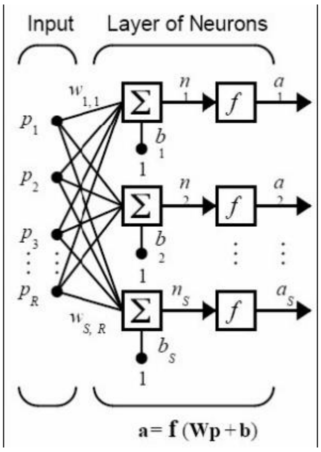


Figure 4. Single layer network

b. Multi layer Network

Multi layer network is the development of single layer network. In this network architecture, besides the input unit and output unit, there is another unit, named the hidden layer. Multi layer network can solve more complex problems than single layer network, although it takes longer time. Figure 5 shows the network architecture of a multi layer network.

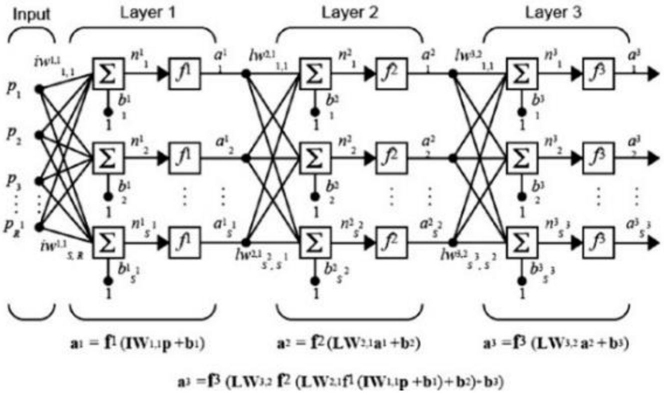


Figure 5. Multi layer network

2.3.3 Activation Function

The activation function used in backpropagation has tu need the conditions, such as continuous, can be differentiated easily, and it’s the undown function. One of the function is often used is the range of binary sigmoid function. (0, 1).

$$f(x) = \frac{1}{1+e^{-x}}$$

Dengan turunan

$$f'(x) = f(x)(1 - f(x))$$

In addition to the binary sigmoid function, bipolar sigmoid function is also often used, it almost similar in form to the binary sigmoid function. However, the bipolar sigmoid function has a range (-1, 1).

$$f(x) = \frac{2}{1+e^{-x}} - 1$$

Dengan turunan

$$f'(x) = \frac{(1+f(x))(1-f(x))}{2}$$

The maximum value of the sigmoid function has one. For the bigger patterns than one, the so the input and output patterns must first be transformed so the pattern has the same range as the sigmoid function used.

2.3.4 Backpropagation standard training

Backpropagation standard training includes 3 phases. The first phase is the forward phase. The input pattern from the input layer to the output layer using the specified activation function. The second phase is the backward phase. The difference between the network output and the desired target is an error occured. The error is propagated backwards from the line directly related to the units in the output layer. The third phase is the modification of the weights to reduce the errors that occur.

a. Phase I: forward propagation

During forward propagation, the input signal (xi) is propagated to hidden layer using the specified activation function. The output of each hidden layer unit (zj) then propagated forward to hidden up layer using the specified activation function until it produces network output (yk). Next, the network output (yk) compared with the target to be achieved (tk). The tk-yk difference is the error occured. If this error is less than the specified tolerance limit, the iteration is terminated. However, if the error is still greater than the tolerance limit, the weight of each line in the network will be modified to reduce the errors occured [13].

b. Phase II: backward propagation

Based on the tk-yk error, a factor k (k=1, 2, …, m) is calculated by used distribution of the error in the yk unit to all hidden units directly connected to yk. k is also used to change the weight of the line directly connected to the output unit. In the same way, it is calculated δj in each unit in hidden layer as the basis for the changes in weight of all lines originating from hidden unit in the down layer until the factor δ in hidden unit directly related to the calculation input unit [13].

c. Phase III: changes in weight

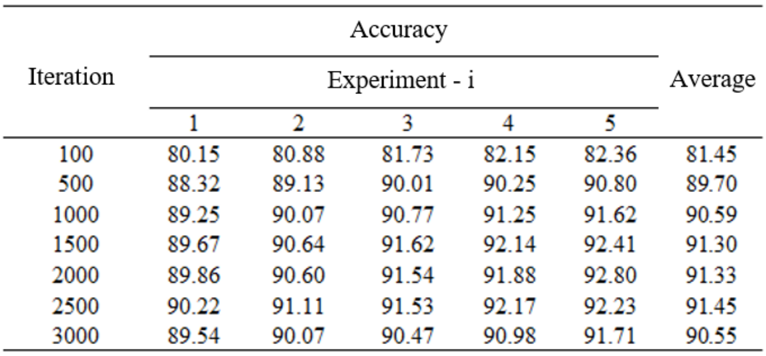
After all the factors have been calculated, the weights of all the lines are modified together. The weight change of line based on the factor of neuron in the upper layer. For example, the change in weight of the line leading to the output layer based on the basis of k in the output unit. These three phases are repeated continuously until the stopping conditions are met. Generally the stop condition is often used of iterations or errors numbers. The iteration will be terminated if the iterations numbers been carried out exceeds the maximum number of iterations set, or if the errors occured is less than the allowable tolerance limit [13].

3. Result and Discussion

3.1 Training and Testing

In this study, it was conducted five test, such as: testing the number of iterations, testing the learning rate, testing the amount of training data and test data, testing the number of node units in hidden layer, and testing the minimum error for accuracy. To perform the initial test, a learning rate of 0.5 is used, the number of nodes in hidden layer is 4, the training data is computational data for January-May 2020 and test data is computational data for June-October 2020, the minimum error is 0.001. Testing the number of iterations aims to get the optimal number of iterations for the level of accuracy. The number of iterations used is 100, 500, 1000, 1500, 2000, 2500, and 3000. The results of the iteration test are shown in table 1.

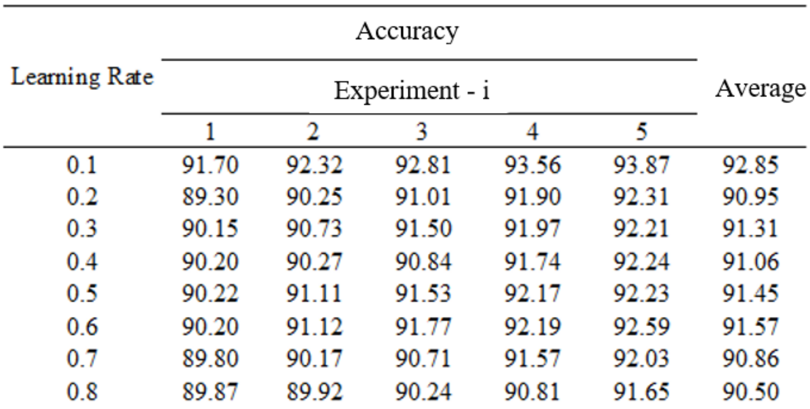
Table 1. Test Results Number of Iterations



Iteration	Accuracy					Average
	Experiment - i					
	1	2	3	4	5	
100	80.15	80.88	81.73	82.15	82.36	81.45
500	88.32	89.13	90.01	90.25	90.80	89.70
1000	89.25	90.07	90.77	91.25	91.62	90.59
1500	89.67	90.64	91.62	92.14	92.41	91.30
2000	89.86	90.60	91.54	91.88	92.80	91.33
2500	90.22	91.11	91.53	92.17	92.23	91.45
3000	89.54	90.07	90.47	90.98	91.71	90.55

The optimal iteration is shown at 2500 iterations because the highest average accuracy value is 91.45%. After obtaining the optimal number of iterations, the next test will use the number of iterations of 2500.
The learning rate test aims to get the optimal number of learning rates to the level of accuracy. The learning rate value is 0.1-0.9 with a multiple of 0.1. The results of the learning rate test are shown in table 2.

Table 2. Learning Rate Test Results

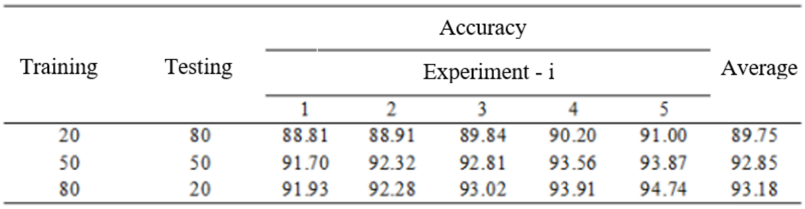


Learning Rate	Accuracy					Average
	Experiment - i					
	1	2	3	4	5	
0.1	91.70	92.32	92.81	93.56	93.87	92.85
0.2	89.30	90.25	91.01	91.90	92.31	90.95
0.3	90.15	90.73	91.50	91.97	92.21	91.31
0.4	90.20	90.27	90.84	91.74	92.24	91.06
0.5	90.22	91.11	91.53	92.17	92.23	91.45
0.6	90.20	91.12	91.77	92.19	92.59	91.57
0.7	89.80	90.17	90.71	91.57	92.03	90.86
0.8	89.87	89.92	90.24	90.81	91.65	90.50

From the results of the learning rate test, the optimal value is shown when the learning rate is 0.1, the highest accuracy value is 92.85%. After obtaining the optimal learning rate, the next test uses a learning rate of 0.1.

The test of training data and test data aims to obtain an optimal comparison of training data and test data to the level of accuracy. The combination of data used is training data 20% and test data 80%, training data 50% and test data 50%, training data 80% and test data 20%. The results of testing the training and test data are shown in Table 3.

Table 3. Test results of training data and testing data

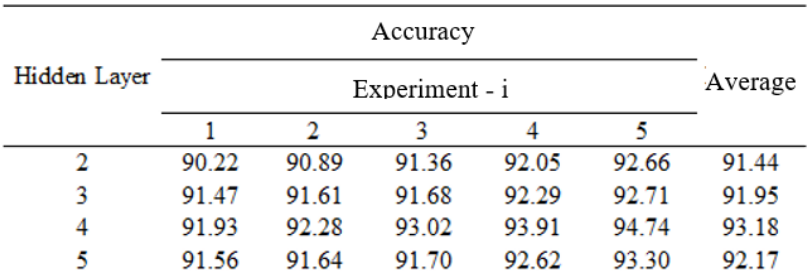


Training	Testing	Accuracy					Average
		Experiment - i					
		1	2	3	4	5	
20	80	88.81	88.91	89.84	90.20	91.00	89.75
50	50	91.70	92.32	92.81	93.56	93.87	92.85
80	20	91.93	92.28	93.02	93.91	94.74	93.18

From the test results, it is known that the more training data the better accuracy. It is because the variations in the data are increasingly diverse so that they can recognize various types of data at the testing time. The highest accuracy is obtained the training data is 80% and the test data is 20% at 93.18%

After obtaining the optimal amount of training data and test data, the next test uses the number of training data 80 and test data 20. Testing the number of nodes in hidden layer aims to get the optimal number of nodes for the level of accuracy. The number of nodes in hidden layer is defined as 1, 2, 3, 4, 5, and 6.

Table 4. Test results of Hidden Layer

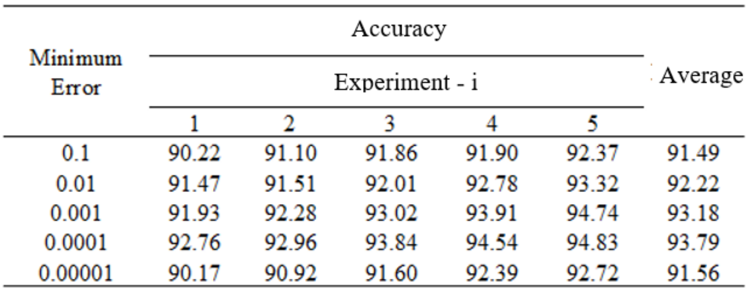


Hidden Layer	Accuracy					Average
	Experiment - i					
	1	2	3	4	5	
2	90.22	90.89	91.36	92.05	92.66	91.44
3	91.47	91.61	91.68	92.29	92.71	91.95
4	91.93	92.28	93.02	93.91	94.74	93.18
5	91.56	91.64	91.70	92.62	93.30	92.17

The test results for the number of nodes are shown in Table 4. From the test results, it is known that the optimal number of nodes is 4 with an average accuracy value of 93.18%.

After obtaining the optimal number of hidden units, the next test uses the number of hidden units 4.

Table 5. Test results of Minimum Error



Minimum Error	Accuracy					Average
	Experiment - i					
	1	2	3	4	5	
0.1	90.22	91.10	91.86	91.90	92.37	91.49
0.01	91.47	91.51	92.01	92.78	93.32	92.22
0.001	91.93	92.28	93.02	93.91	94.74	93.18
0.0001	92.76	92.96	93.84	94.54	94.83	93.79
0.00001	90.17	90.92	91.60	92.39	92.72	91.56

The minimum error test aims to get the optimal minimum error for the accuracy level. The minimum error set is 0.1, 0.01, 0.001, 0.0001 and 0.00001. From the results of Table 5, it is known that the optimal minimum error is 0.00001 with an accuracy of 93.79%.

From the training results obtained parameter values to get optimal accuracy are learning rate 0.1, training data 80 and test data 20, number of nodes in hidden layer 4, minimum error 0.0001, and the number of iterations 2500.

3.2. High Availability Web Server

This study uses two approaches, they are the simulation approach and the implementation approach. The case study in this research is a web server.

3.2.1. Simulation

The research simulation uses raspberries to form a web server cluster simulation. Raspberry is an energy efficient Small Board Computer (SBC). Raspberry is capable of running the linux operating system. Therefore, on the SDcard is embedded in the Linux OS, a web server and database server have been run. The use of SBC can save energy and space used to implement a web server cluster system. The topology shape is simulated as shown in Figure 6.

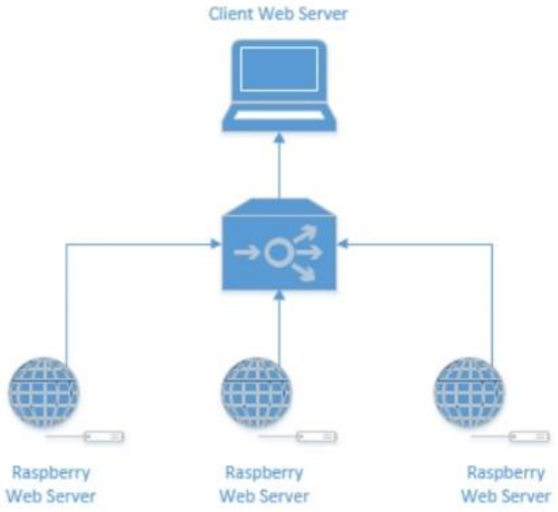


Figure 6. Cluster Web Server

The client web server uses a web browser such as Firefox, Opera, Chrome and so on. Load balancer using ZEN Lab and Cluster Web Server using apache.

At the simulation phase, the web server cluster has been formed, as evidenced by the alternate responses given by the raspberry to the request received by the balancer. Zen Lab is able to forward every HTTP request from the client to each server.

3.2.2. Implementation

Cluster data base has been implemented directly in data center. The application uses the database is sinta.ristekdikti.go.id. This application is a Science and Technology Index application to collects data from researchers in Indonesia.

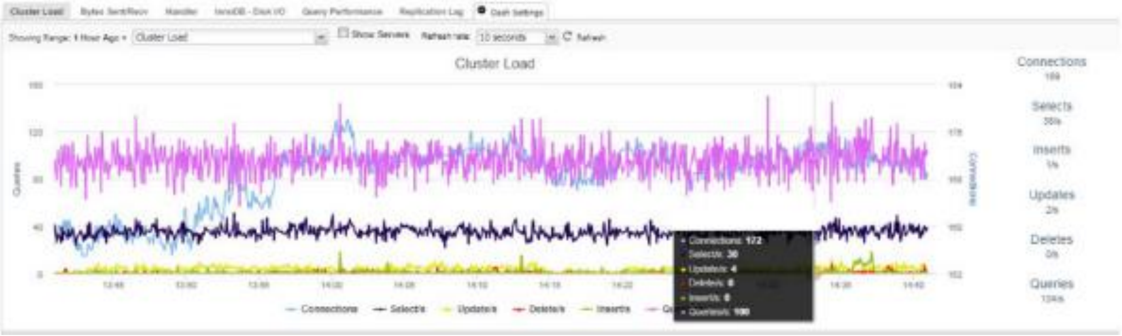
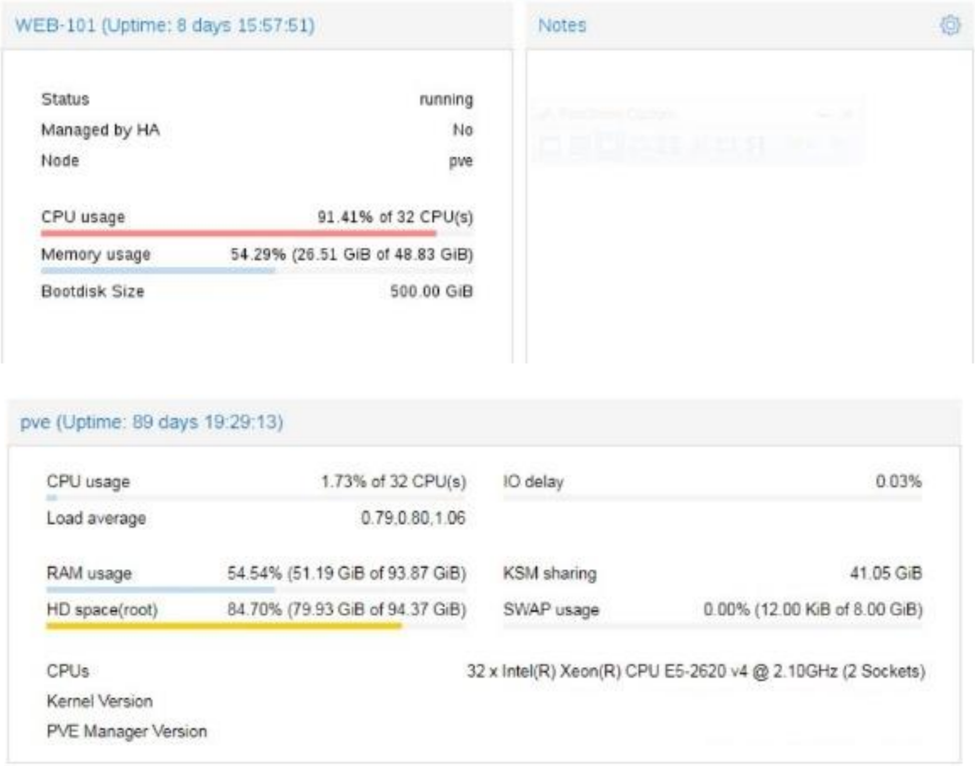


Figure 7. Load Database Server

At the beginning of the development of the server, the database server cluster was not implemented, as resulted in the Sinta server being overloaded. As a result of the overload, the CPU and memory used by the server is not able to run the application perfectly. Depleted resources cause the application to slow down and sometimes internal server errors.



WEB-101 (Uptime: 8 days 15:57:51)

Status running
Managed by HA No
Node pve
CPU usage 91.41% of 32 CPU(s)
Memory usage 54.29% (26.51 GiB of 48.83 GiB)
Bootdisk Size 500.00 GiB

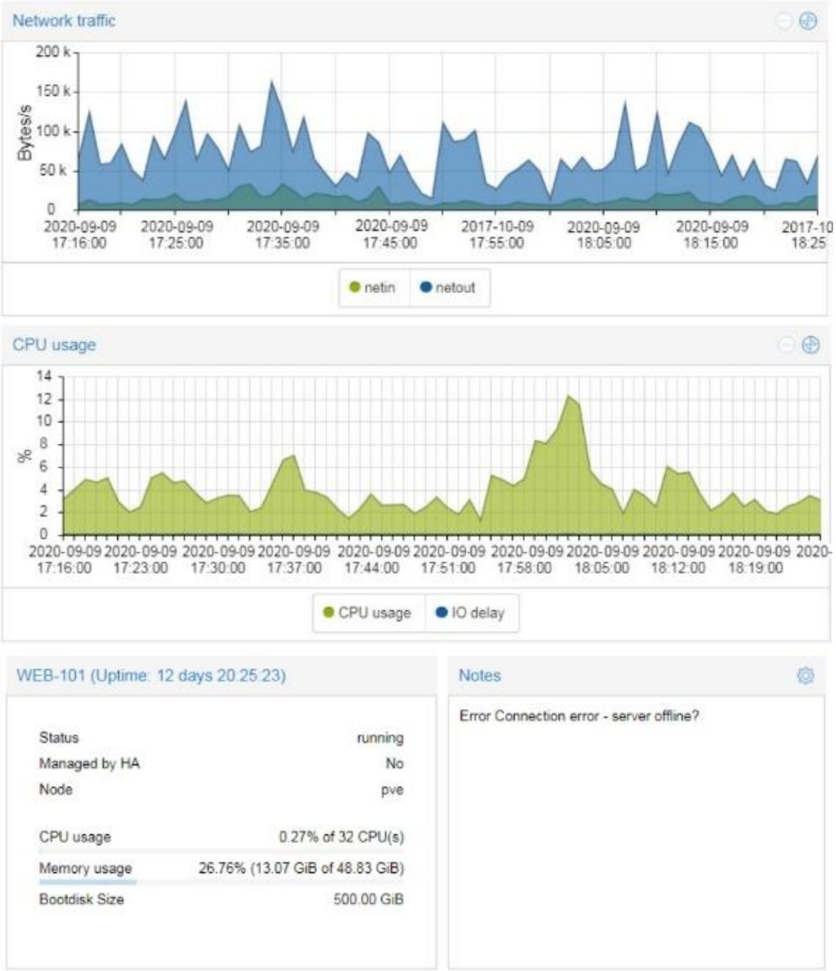
Notes

pve (Uptime: 89 days 19:29:13)

CPU usage 1.73% of 32 CPU(s)
IO delay 0.03%
Load average 0.79,0.80,1.06
RAM usage 54.54% (51.19 GiB of 93.87 GiB)
KSM sharing 41.05 GiB
HD space(root) 84.70% (79.93 GiB of 94.37 GiB)
SWAP usage 0.00% (12.00 KiB of 8.00 GiB)
CPUs 32 x Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz (2 Sockets)
Kernel Version
PVE Manager Version

Figure 8. CPU & Server Memory Consumption

Two months since this research was conducted (August-September) there has been a spike in hits add to the application load. However, since the implementation of the web server cluster, there has been a decrease in the CPU and Memory consumption queues charged to the server. The statistics regarding the resource are shown in Figure 9.



WEB-101 (Uptime: 12 days 20:25:23)

Status	running
Managed by HA	No
Node	pve
CPU usage	0.27% of 32 CPU(s)
Memory usage	26.76% (13.07 GiB of 48.83 GiB)
Bootdisk Size	500.00 GiB

Notes

Error Connection error - server offline?

Figure 9. CPU & Memory statistics after clustering

4. Conclusion

The conclusions of this study are:

1. The algorithm used to build HA can use classical and modern algorithms.
2. The algorithm used in this research is using backpropagation neural network.
3. In this study, the parameter values to obtain optimal accuracy are obtained, they are: learning rate 0.1, training data 80 and test data 20, number of nodes in hidden layer 4, minimum error 0.0001, and the number of iterations 2500.
4. The best accuracy value using the parameter is 93.79%.

References

[1] M. Ljubojevic, A. Bajic, and D. Mijic, "Implementation of High-Availability Server Cluster by Using Fencing Concept," in 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2019, pp. 1–5, doi: 10.1109/INFOTEH.2019.8717752.

[2] S. Aalto and P. Lassila, "Near-optimal dispatching policy for energy-aware Server clusters," Perform. Eval., vol. 135, p. 102034, Nov. 2019, doi: 10.1016/j.peva.2019.102034.

[3] J. von Kistowski et al., "Measuring and rating the energy-efficiency of Servers," Future Gener. Comput. Syst., vol. 100, pp. 579–589, Nov. 2019, doi: 10.1016/j.future.2019.05.050.

[4] J. David and C. Thomas, “Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic,” Comput. Secur., vol. 82, pp. 284–295, May 2019, doi: 10.1016/j.cose.2019.01.002.

[5] H. Handoko, S. M. Isa, S. Si, and M. Kom, “High Availability Analysis with Database Cluster, Load Balancer and Virtual Router Redudancy Protocol,” in 2018 3rd International Conference on Computer and Communication Systems (ICCCS), 2018, pp. 482–486, doi: 10.1109/CCOMS.2018.8463263.

[6] Formilux Staf, “HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer.” [Online]. Available: http://www.haproxy.org/. [Accessed: 23-Dec-2019].

[7] Z. Cai and Y. Ling, “Research on the Resource Shortage Scheme Based on High Availability Cluster,” in 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA), 2019, pp. 71–74, doi: 10.1109/ICCNEA.2019.00024.

[8] A. A. Lydia and F. S. Francis, “Convolutional Neural network with an Optimized Backpropagation Technique,” in 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 2019, pp. 1–5, doi: 10.1109/ICSCAN.2019.8878719.

[9] Y. Jiao and W. Wang, “Design and Implementation of Load Balancing of Distributed system based Web Server,” presented at the 2010 Third International Symposium on Electronic Commerce and Security, 2010, pp. 337–342.

[10] Savvion Team, “Clustering Guide - Load Balancing (Computing) - Computer Cluster.” [Online]. Available: http://knowledgebase.progress.com/servlet/fileField?id=0BEa0000000LPUH. [Accessed: 01-Jul-2019].

[11] F. As, “Design Of Load Balancing Scheduling Method Of Iptv Internet Protocol Television And Lms Learning Management System,” Pap. Present. Electr. Eng. RSE 621388 07 Fac R 2012, Jul. 2012.

[12] Bourke, Tony, Server Load Balancing. Sebastopol: O’Reilly & Associates, Inc., 2010.

[13] N. A. Rizal, “Implementasi Metode Hybrid Jst-Som Pada Prediksi Churn Pelanggan Seluler: Studi Kasus Pt. Telekomunikasi Seluler,” p. 125.