iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45225

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Optimized Load Balancing Using Adaptive Algorithm in Cloud Computing with Round Robin Technique

Vinay Kumar Prasad[1], Dr. Dharamveer Singh[2], Vikas Gupta[3]
[1, 2, 3]*Deptt.of Computer Science & Engineering, R.D. Engineering College, Ghaziabad, India*

Abstract: ***Developments in the field of computer networks have been carried out by several groups. However, there are still a lot of wrong problems one is the server load. For this reason, a system will be implemented Load Balancing with the aim of overcoming the server load which is not in accordance with its capacity and to optimize server load before and after the implementation of the Round robin Algorithm Load Balancing system on the cloud servers.***
The method used is the comparative method, namely researches that compares and analyze two or more symptoms, compare least connection algorithm as the previous algorithm with Round robin algorithm. Load Balancing Testing with both algorithms using a software called Httperf.
Httperf displays the value according to parameters. The parameters used are Throughput, Response Time, Error and CPU Utilization. The test results show that load balancing with the algorithm round robin is more effective to handle server load than algorithm.
The previous one was Least Connection. It is proven that in each respondent's assessment of the load balancing system. Test rating throughput obtained a percentage of 81.11% with good criteria, testing response time obtained a percentage of 81.78% with good criteria, testing error obtained a percentage of 84.67% with very good criteria, and cpu utilization testing obtained a percentage of 82% with good criteria.

I. INTRODUCTION

The existence of communication network technology allows two entities to be connected to each other. This allows computers to be associated to every one further via a communication group. With the increasing alacrity of delivery that current communication technologies can make, this has allowed computers to share resources, such as CPU, memory and storage media, to provide applications that are superior to a single system [1]. This is also driven by the problems to be solved which have become more complex and on a larger scale to be worked on by a single computer. One of the applications that take advantage of the advances in communication network technology is grid computing and cluster computing. Network computing is a structure of computing that occupies numerous machines that are typically heterogeneous and spread over different geographic locations. Meanwhile, cluster computing is a computation that involves many computers located in one place. Cluster computing is one of the constituent components of grid computing [2-4]. In cluster computing, the terms server, node, client, job and task are known. A task is a computational unit which is usually a program that cannot be broken down into small processes. Job is a computational activity that consists of one or several tasks [2-4]. Clients are entities that create jobs, servers are entities that distribute jobs and nodes are entities that perform computational tasks [2-4]. When receiving a job, there is a possibility that the computers in the parallel computing system experience an unbalanced load. This unbalanced system load condition can reduce the Quality of Service (QoS), so a load balancing method is needed. In cluster computing, there is a term load balancing policy. Policy load balancing in cluster computing can be categorized into four, namely static, dynamic, hybrid and adaptive methods [5-8]. Static policies consider the state of systems and applications statically and pertain this in sequence into pronouncement production. The foremost improvement of static policy is mediocre transparency because assessment creation has been through prior to the job is given [5-8]. Dynamic policies work by moving jobs from an overloaded computer to a lesser computer. The hybrid method is a load balancing method that combines static methods and dynamic methods. Meanwhile, the adaptive method is a method that adaptively adjusts the portion of the task using the latest information from the system and a certain threshold value. Adaptive policy for load balancing problems has been done a lot of research.

©IJRASET: All Rights are Reserved

A. Identification Of Problem

From the facts on the background, several can be identified problems, including:

1) Internet users increase along with the number of devices that continues increases, so that server performance is required to increase as well.

2) Servers that have high performance have a good price high, so it is necessary to find a solution to create a server with services high but minimal costs.

B. Formulation Of The Problem

Based on this background, it can formulate problems which exist, namely:

1) How to deal with the server load in Cloud Computing that is not in accordance with the capacity?

2) How to optimize the server load before and after application of the Round robin Load Balancing Algorithm system on Cloud servers?

C. Restricting The Problem

Given the wide scope of the formulation of the problem regarding this final project also limited time, energy, and thoughts so there is a need for limitations the problem with considering the networking aspects of the internet with using Load Balancing technology, there is a limitation of the problem are:

1) Research is not carried out directly on a Cloud Network currently running ISP.

2) How to deal with the burden of accessing web applications and applications the server will continue to rise quite high into a few server computer so that the server load does not lie on one server only.

3) Parameters used in measuring server performance are throughput, response time, error and CPU Utilization.

D. Research Purposes

The objectives of this study are:

1) To overcome the server load at Cloud Servers which is not in accordance with its capacity;

2) To optimize the server load before and after application of the Round Robin Load Balancing Algorithm system on server at Cloud.

E. Research Contribution

The contribution of this research is to improve the performance of the load balancing mechanism using an adaptive distribution strategy based on the Reinforcement Learning algorithm in dynamic conditions, including: changing throughput between nodes and servers; and the processor load on a node.

II. PROPOSED METHODOLOGY

A. System Requirements Analysis

In making this final thesis, using a qualitative approach, namely investigate that be determined to recognize the incident knowledgeable by research topic such as performance, perceptions, inspiration, actions and others. The qualitative research method in this study uses comparative research. However, comparative research is a study that compares and analyzes two or more symptoms. The comparison in this study is to compare the round robin algorithm with least connection.

1) *Literature Study:* Literature study is carried out by collecting information sources about the object to be studied and to obtain the correctness of the procedures carried out in implementing the load balancing system with the Round robin algorithm for handle the server load. Information obtained from articles and journals that support the writing of the final project.

2) *Topology Design (System):* At this stage the system used to support load balancing servers is the LINUX RHEL OS operating system. The method used for designing this topology is LVS Direct Routing, and the algorithm used to implement load balancing servers is the Round robin algorithm.

B. Software Components

The software used in this research is as follows the following:

1) Operating System

The operating system used as the load balance is RHEL Server 6.5. Meanwhile, as a virtual server using the RHEL 6.0 operating system.And for client computers using the Windows Desktop 7 operating system.

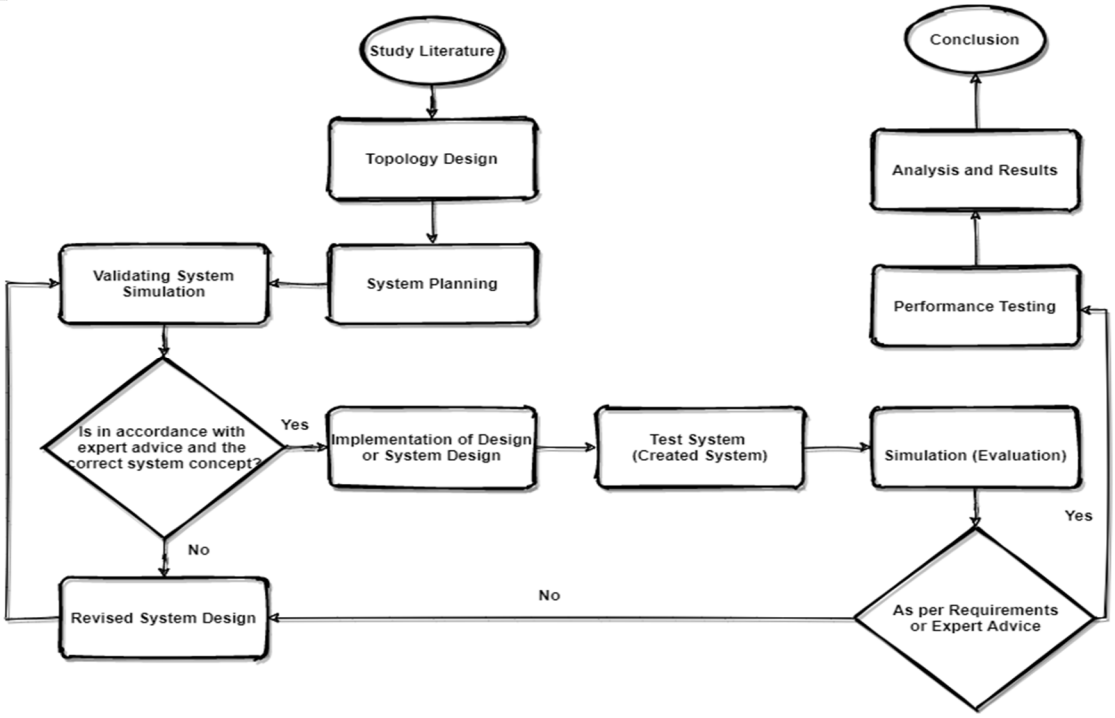


Figure 3.1: Research Procedure Flowchart

The explanation of each stage system in the flowchart above is as follows:

2) System Design

The network design used for the experiment uses a simple local network as shown in the following figure:

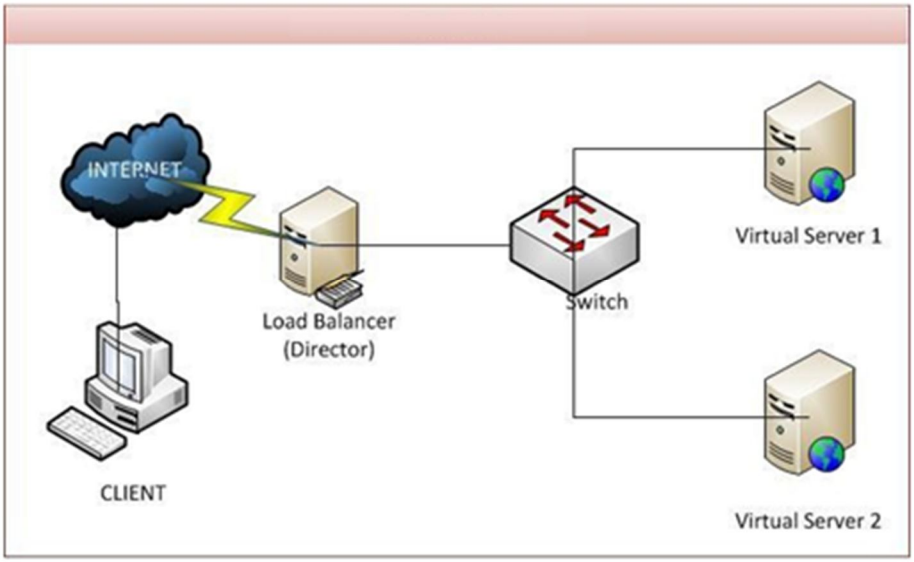


Figure 3.2: Scenarios 1 and 2 with 2 Virtual Servers

Figure 3.2 is an experimental scenario using 2 virtual servers connected to the Director (Load Balancer). Each virtual server runs apache and has a load balance using the server director. On this load balance runs the IPvsadm software which experiments to replace the load balance algorithm. Experiments were carried out with 2 algorithms, namely Least connection and Round robin with 10 trials each with a predetermined request.

Figure 3.3.is an experimental scenario using 3 virtual servers connected to the Director (Load Balancer). This experiment is the same as scenario 1 where the virtual server is added to become 3 virtual servers. Experiments were carried out with 2 algorithms, namely least connection and round robin with 10 trials each with a predetermined request. The following are details of the basic configuration of each computer in scenario 2:

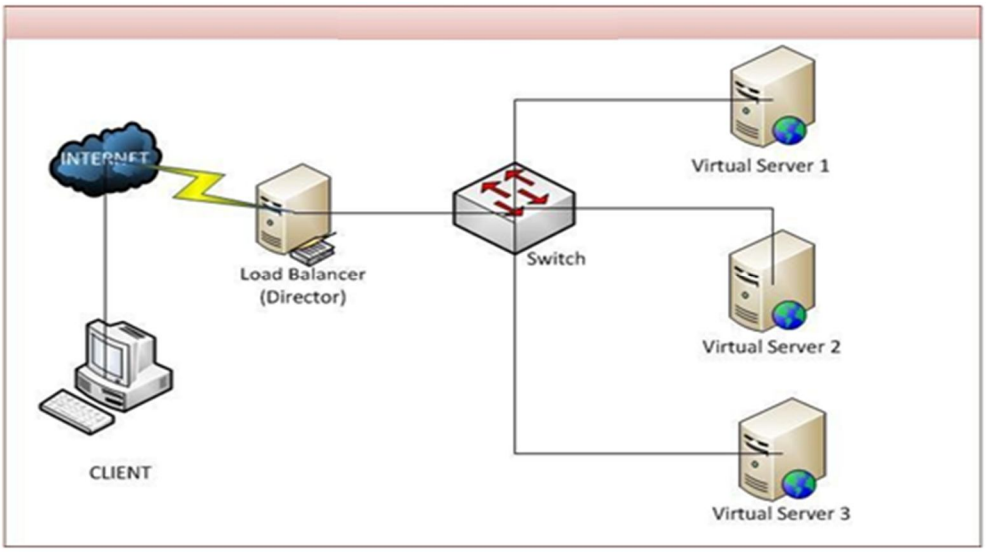


Figure 3.3: Scenario 2 with 3 Virtual Servers

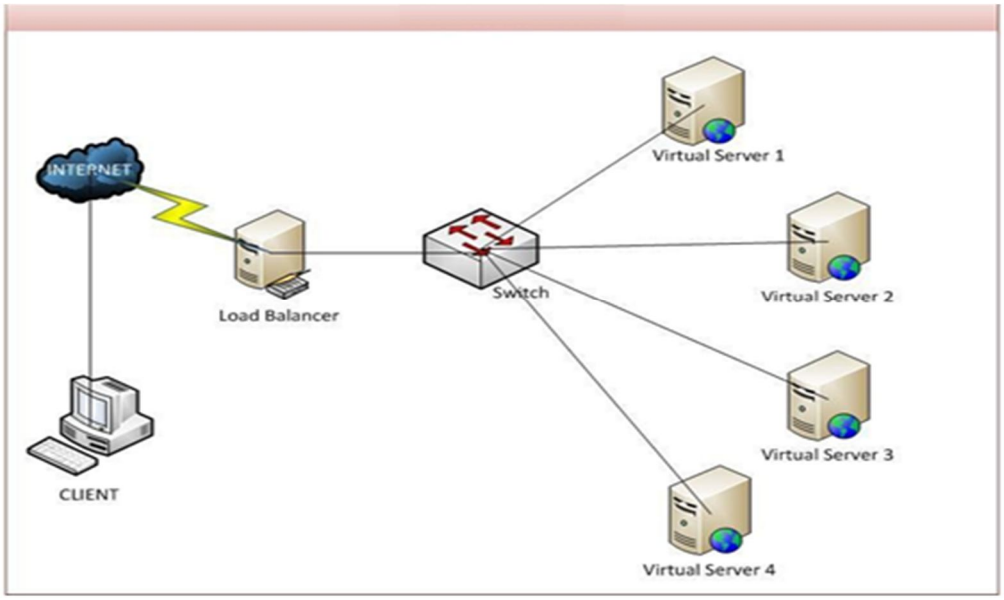


Figure 3.4: Scenario 3 with 4 Virtual Servers

Figure 3.4 is an experimental scenario using 3 virtual servers connected to the Director (Load Balancer). This experiment is the same as scenario 1 where the virtual server is added to become 3 virtual servers. Table 3.3: Configuration Scenarios 3 with 4 Virtual Servers

3) Implementation

System Kernel Linux Director: System operation Linux that use the kernel version below:-

Linux Kernel 2.4.28 doesn't support Linux virtual server yet, so patching is needed by compiling the kernel, but in this final project the author uses a kernel that supports Linux virtual server, but needs to activate the virtual server modules in the kernel using modprobe.

echo ipvs dh >> /etc/modules echo ipvs ftp >> /etc/modules echo ipavs>> /etc/modules echo ipavsalc>> /etc/modules echo ipavsarr>> /etc/modules modprobeipavsadhmodprobeaipavsaftpamodprobeaipavsamodprobeipavsalcamodprobeipavsarr

Whereas for kernels that do not yet support the Linux virtual server, compilation must be done by taking the kernel source from the kernel.org site or scheme can use the Synaptic Package Manager tool. After all the preparations are complete, then recompile the linux kernel. To recompile the kernel, the packages used for compilation are needed, namely:

kernel-packager
libnucrses5-devr
yum install kernel-packagellibncurses5-dev

After the above packages are installed, the kernel compilation is ready to do. The next steps for compiling the kernel are:-

a) Go to the /usr/src directory and extract the existing kernel-2.6.x sources.
tar –jxvfalinux-2.6.x.tar.bz2a

b) Enter the ipvs patch in the /usr/src directory and extract the patch.
tara-zxvfaipvs-1.1.7.tar.gza

c) Enter the linux-2.6.x directory as a result of the extraction that has been done, and do the kernel patching.
cd / linux-2.6.x
#patchr-pqr<../ipvs-1.1.7/linuxkernel_ksyms_c.diffr
patchr-pqr<../ipvs-1.1.7/linuxnet_netsyms_c.diffr

If the forwarding service from the real-server you want is direct routing, then add the kernel patching:

patchr-pqr<../ipvs-1.1.7/contrib/patches/hidden-2.6.xpre10-1.diffr

d) Configure the kernel.
makermenuconfigrorrmakerxconfigr

e) Enter the initial kernel configuration menu.

f) Enter the networking options menu.

g) Activate the features that support LVS, namely:
cd /linux-2.6.xr
patchr-pqr<../ipvs-1.1.7/linuxkernel_ksyms_c.diffr
patchr-pqr<../ipvs-1.1.7/linuxnet_netsyms_c.diff
patchr-pqr<../ipvs-1.1.7/contrib/patches/hidden-2.6.xpre10-1.diffr
makermenuconfigrorrmakerxconfig

Networking options --->
Network packet filtering (replaces ipchains) <m> IP: tunneling

C. Research Parameters

Based on observations and evaluations from several previous sources, research on load balancing with the Round robin algorithm will be carried out using the LVS Direct Routing method with the help of Httperf software so that the parameters to be calculated are Throughput, response time, error and CPU Utilization. According to Mulay& Jain, (2013) in measuring load balancing the parameters used include:

1) Throughput is the amount of data received in units of time.
2) Response Time is the time it takes to complete one request and send it back to the client.
3) Error are a number of requests that the web server has not responded to.
4) CPU Utilization is the amount of resources needed to carry out a computerized process.

D. Research Instruments

The form of the instrument used to determine the feasibility of the load balancing system in this study is a questionnaire. Meanwhile, the instrument used to determine the load balancing system test assessment is in the form of an observation table. Based on how to answer it, the questionnaire can be divided into 2, namely open and closed questionnaires and in this study used a closed questionnaire with an answer, namely the likert scale. A closed questionnaire is a questionnaire in which the number of items and alternative answers to whatever the response has been determined, the respondent only needs to choose the answer according to the actual situation.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

CRITERIA	SCORE3A
VERYAHIGH HIGHA	53
MEDIUMA	43
LOWA	33
VERYALOW	23
	13

Table 3.4: Criteria and Scores for the System Eligibility Test Instrument

E. Observation Table

The observation table is used to observe changes in the value of all parameters used, with this observation table it can be concluded whether the system can work properly and in accordance with the concept of the system. The grid of the observation table is as follows:

ASPECT	INDICATOR
Implementation of Load Balancing System Testing with 2 Algorithms, namely Least Connection and Round Robin	1. Prepare the equipment and materials needed. 2. Install equipment according to the required topology. 3. Implementing a load balancing system by comparing the least connection and round robin algorithms on the network system at ISP with the LVS (Linux Virtual Server) method. 4. Implement each algorithm by the number of servers different. 5. Testing with HTPperf *6.* Analyze and record the results of observations according to the observation indicators (throughput, response time, and error). 7. Conclusion

Table 3.5: Grid of Observation Sheet of Load Balancing System

F. Data Analysis Technique

Analysis of data obtained from the study includes a system feasibility test and observation tables from the results of system testing carried out by the following techniques:

Feasibility Analysis

The analysis technique used to calculate data from each aspect consists of the Throughput, Response Time, Error, and CPU Utilization tests. The calculations used to process the data from the instrument are the calculation of the average value and the calculation of the percentage score for each aspect:

1) Calculating the response value of each aspect or sub variable.
2) Recap value
3) Calculate the average value
4) Calculate the percentage using the percentage formula as the following:

$$\boldsymbol{P = \frac{n}{N} * 100\% \ (eq.3.1)}$$

Information:

P = Percentage (%)

n = Empiric score (score obtained)

N = Ideal score for each question item

5) Determine the criteria level with the following steps:
   a. Determine the highest percentage number with the formula:

$$\boldsymbol{\frac{Obtained\ Score}{Maximum\ Score} * 100\% = \frac{5}{5} * 100\% = 100\%}$$

   b. Determine the lowest percentage number with the formula:

$$\boldsymbol{\frac{Obtained\ Score}{Maximum\ Score} * 100\% = \frac{1}{5} * 100\% = 20\%}$$

To find out the criterion level, the score obtained (in%) of the percentage calculation results are consulted with the criteria table. Criteria table is used to determine the category “very good”, “good”, "good enough", and "not good enough". Maximum values, minimum values and intervals are used to create a table. The maximum value comes from the highest percentage figure, the minimum value comes from the lowest percentage figure, while the length of the interval is searched in the following way:-

- Determine the range (largest data-smallest data), namely 100-20 = 80
- Determine the assessment interval, namely 4 (very good, good, good enough, not good, not good)
- Determine the width of the interval by dividing the range by the assessment interval, namely 80/5 = 16

NO.	PERCENTAGEE	CRITERIAE
1	184% <ESCOREE ≤ 1100%	VERYEGOOD
2	168% <ESCOREE ≤ 184%	GOODE
3	152% <ESCOREE ≤ 168%	PRETTYEGOOD
4	136% <ESCOREE ≤ 152%	NOTEGOOD
5	120% <ESCOREE ≤ 136%	NOTEGOOD

Table 3.6: Interval of Qualitative Score Categorization

In the analysis of observations, there are several aspects that must be considered, namely the value that appears on the HTTF. Research is carried out to see the changes in the value that will appear, from this value the average will be sought according to the parameters used and the number of requests. After recording the results of the treatment, the results will be compared and graphed according to the algorithm under study.

III. RESULTS AND DISCUSSION

A. Result Analysis

The results of the study were carried out on two sides for the four parameters. On the client side using the Httperf software to test the parameters of throughput, response time, and error and CPU Utilization; Testing was carried out 3 times on each request to see the trend of the research results. The value of the research results is then calculated on average in 3 research trials and divided by the total number of requests given. So, the results obtained from the throughput, response time, and the chance of an error on each request. Meanwhile, for the calculation of the CPU Utilization load balancing server functions for how much power is needed when load balancing works. After the experiment is sufficient, the next thing is to distribute a questionnaire to the respondents. The questionnaire aims to determine the appropriateness of the load balancing system with the Round robin algorithm being implemented according to the predetermined indicators. The following is an explanation of the process that occurs in LVS (Linux Virtual Server) with the implementation of Direct Routing and how to get test data to compare algorithms before Round robin and after Round robin. The following is an example of taking test values from a load balancing system using the Round robin method.

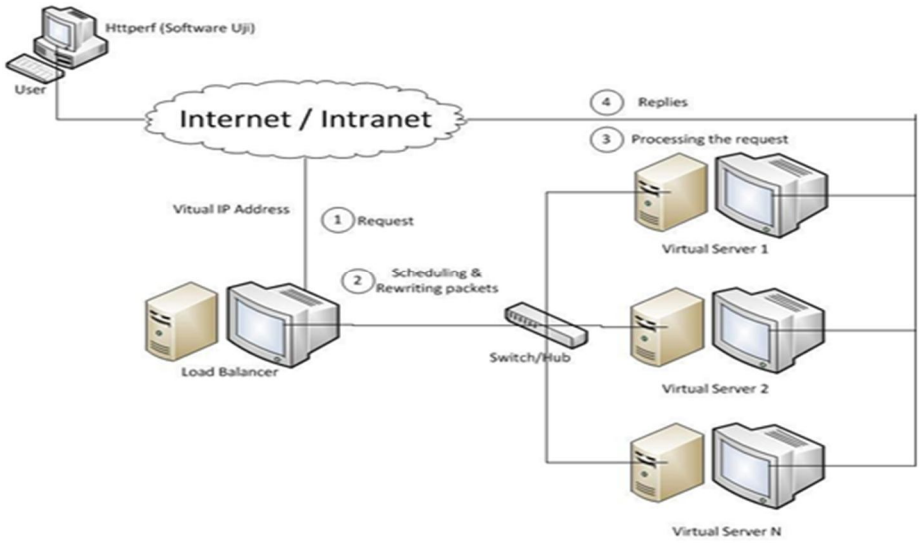


Figure 4.1: Details of the processes that occur in the LVS Direct Routing System

In process 1 in Figure 4.2., the client makes a request, where the client already has HTPperf software which functions as a test data producer. Example of a request process at HTPperf:

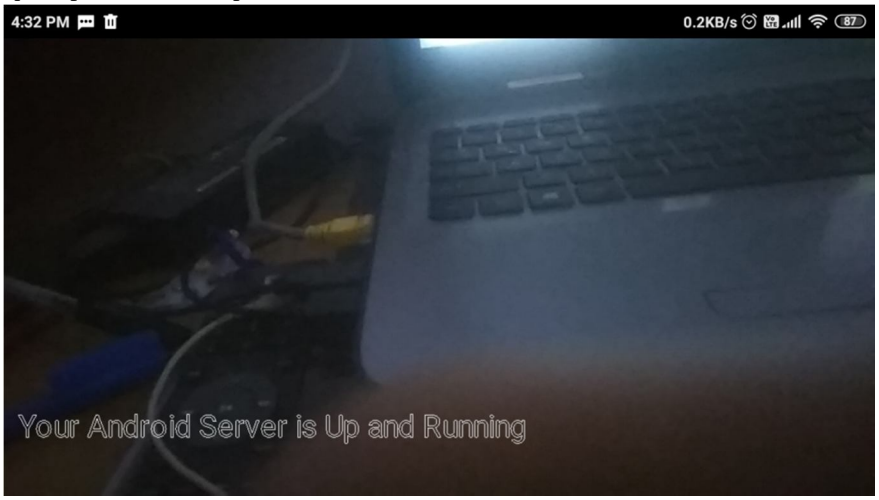


Figure 4.2: Custom Server (Web Server) over the Cloud Resource

B. Throughput Parameter Analysis

The throughput parameter in this study represents the number of requests that can be responded to by the web server at one time. This parameter is calculated in units of Kb / second. The greater the value of this parameter, the better the performance of the web server.

Figure 3.10 shows the amount of throughput using the least connection algorithm. Changes in the amount of throughput produced are not very significant. Same is the case with the graphic changes to the Round robin algorithm. The resulting changes tend to be stable and decrease according to the increasing number of requests received by the web server.

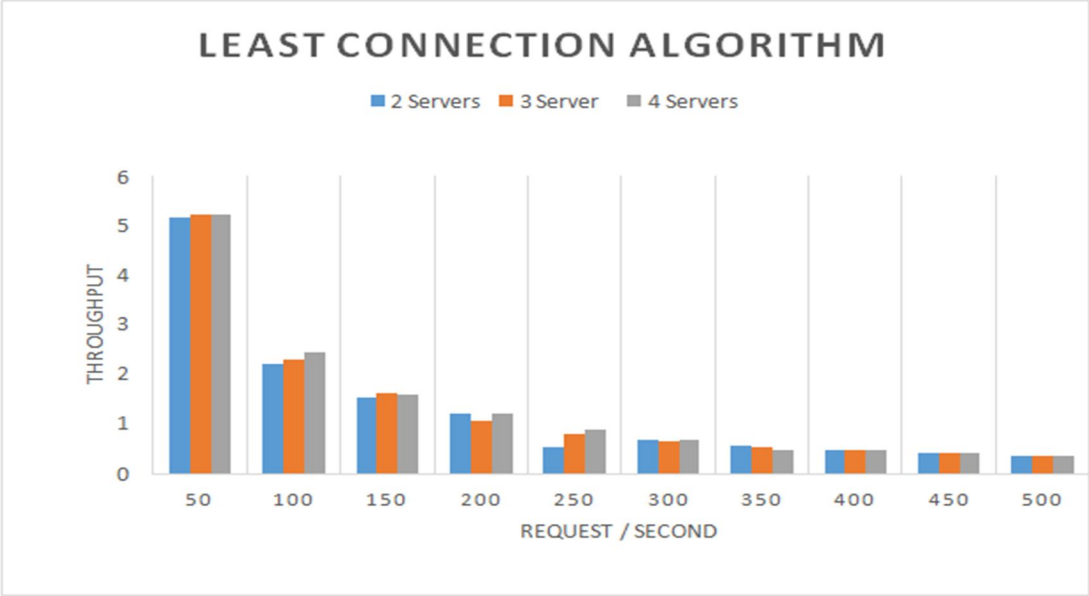


Figure 3.10: Throughput Graph of the Least Connection Algorithm

Figure 3.11 shows the throughput of the study using the Round robin algorithm. The resulting change in the amount of throughput is not very significant. This can be seen from the throughput comparison graph in Figure 4.11, where the resulting graph tends to be down and stable. Changes in the number of requests / second resulted in the value of throughput decreasing in accordance with the number of requests received more and more by the web server.

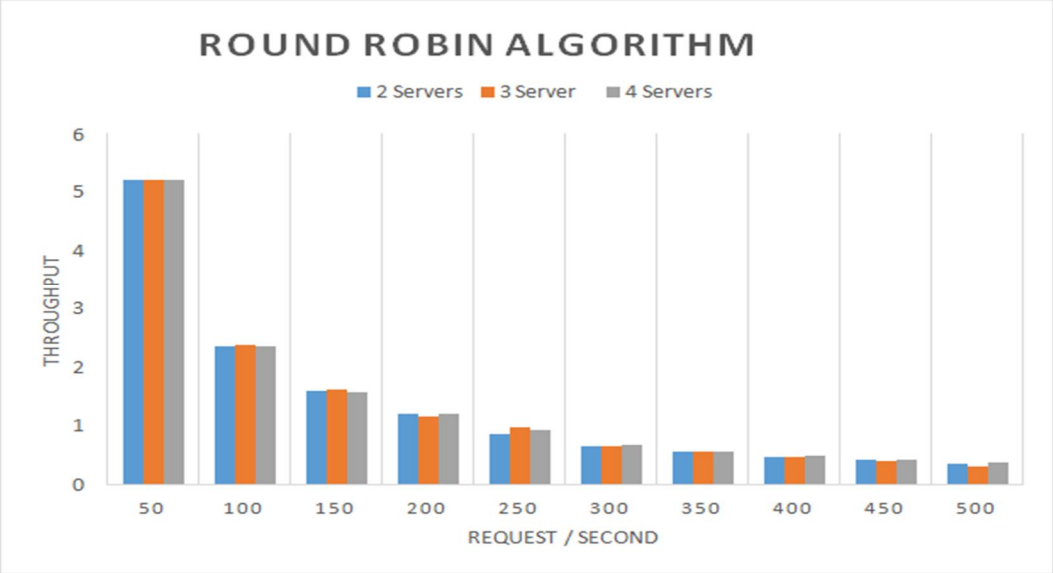


Figure 3.11: Throughput Graph of the Round Robin Algorithm

Figure 3.12 shows a comparison of the response time on the two algorithms that have been generated. The movement of the graph starts at 50 requests / second with a very low response time value, meaning that at 50 requests / second, the resulting response time is very good and fast. This is because the apache web server can handle 50 requests / second directly. Graph movement at 200 requests / second has increased very significantly, this is because the server experienced sudden surges and queues which resulted in the response time value is high and tends to be stable. The next graph movement has decreased, which means that the response time value is better, because the server has been able to adapt to the number of requests that have been given.

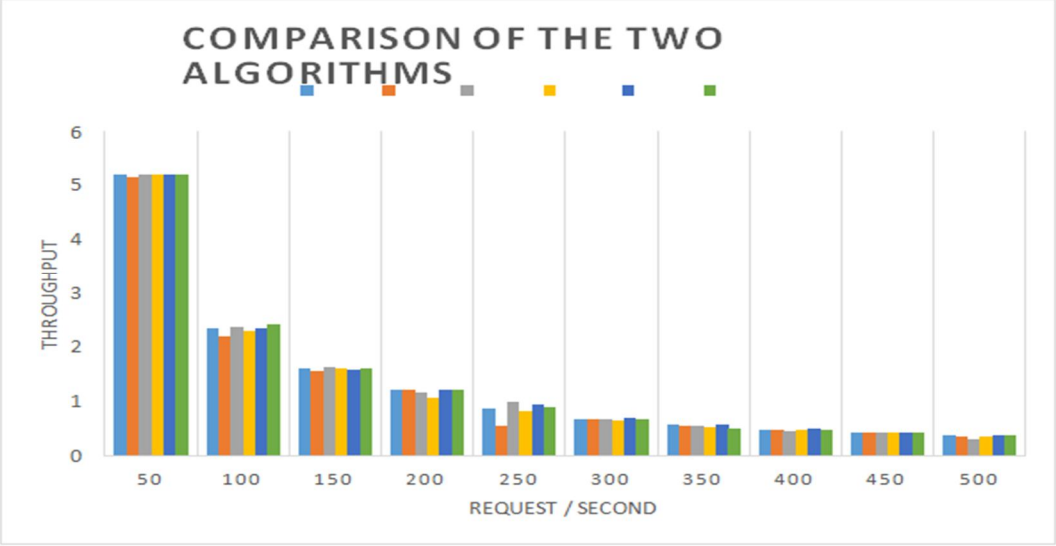


Figure 3.12: Two Algorithm Response Time Graph

C. Error Parameter Analysis

The error parameter describes the number of errors that occur when the web server responds to requests from clients. Data from the observed error parameters are attached and presented in tabular form,

Figure 3:13 shows the number of errors in the Least connection algorithm. Just like the Round robin algorithm on requests for 50, 100, 150 and 200 requests / second, there were no errors. However, in this experiment the Round robin algorithm shows the average error value is smaller than the Least Connection algorithm.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

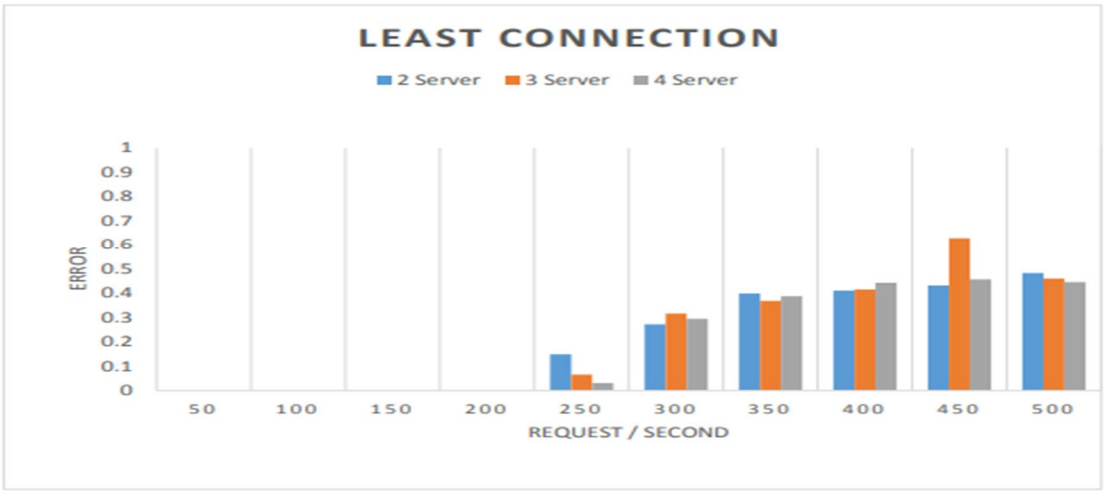


Figure 3.13: Least Connection Algorithm Error Graph

Figure 3.14 shows the number of errors that occur in the implementation of load balancing using the Round robin algorithm. Seems like on requests 50, 100, 150 and 200 requests / second there was no error. The error occurred starting at 250 requests / second. The graph above shows that the experiment tends to be stable and has an increase in the error value in the addition of the number of requests / errors.

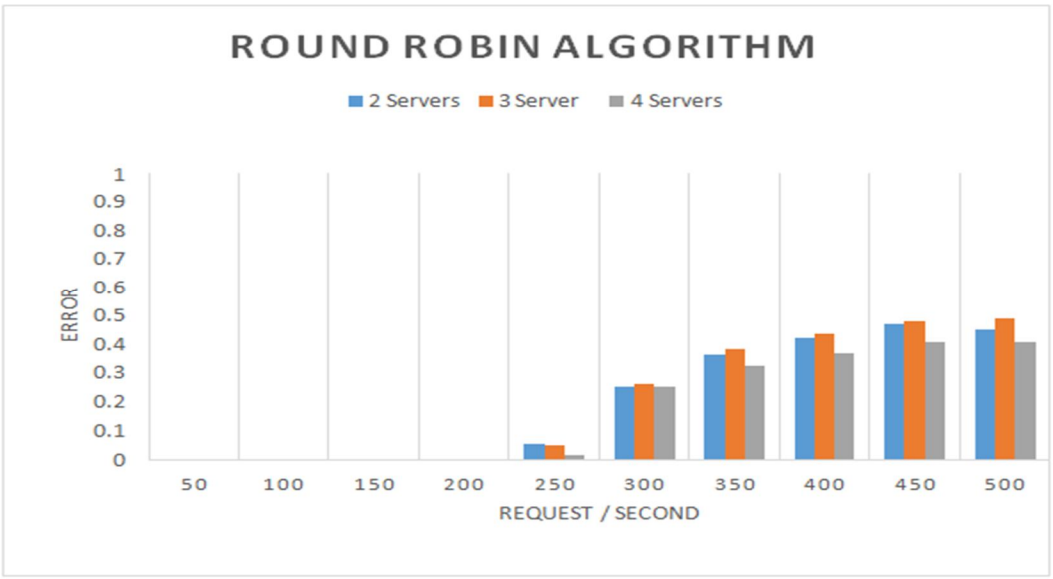


Figure 3.14: Graph of Round Robin Algorithm Error

Figure 3.15 shows a graph of the comparison of the error value between the Round robin algorithm and the least connection algorithm. The graph above shows that at 50 requests / second up to 200 requests / second shows that there are no errors. At 250 requests / second an error begins and increases with the addition of requests / second. The Least connection algorithm shows a high error spike value when a request is given equal to 250 requests / second. This happens because the server experiences an unbalanced condition with increasing requests or it is called saturated (server saturation point).

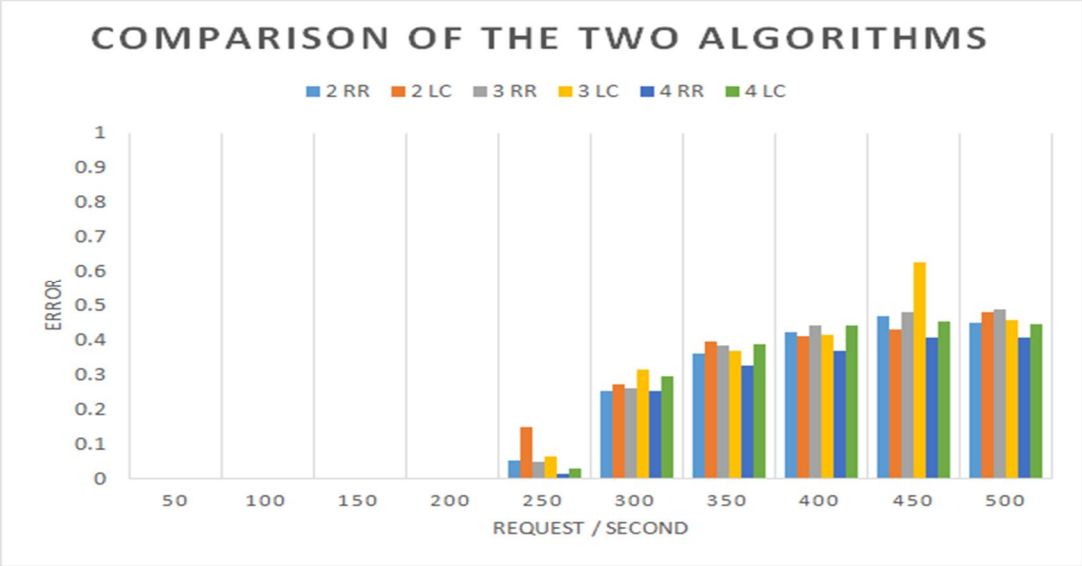


Figure 3.15 Error Graphs of Two Algorithms

D. CPU Utilization Analysis

The value to be measured is the value of the CPU load balancing server performance. CPU Utilization is calculated in percentage (%). So that the smaller the percentage value, the fewer sources will be used in the load balancing process.

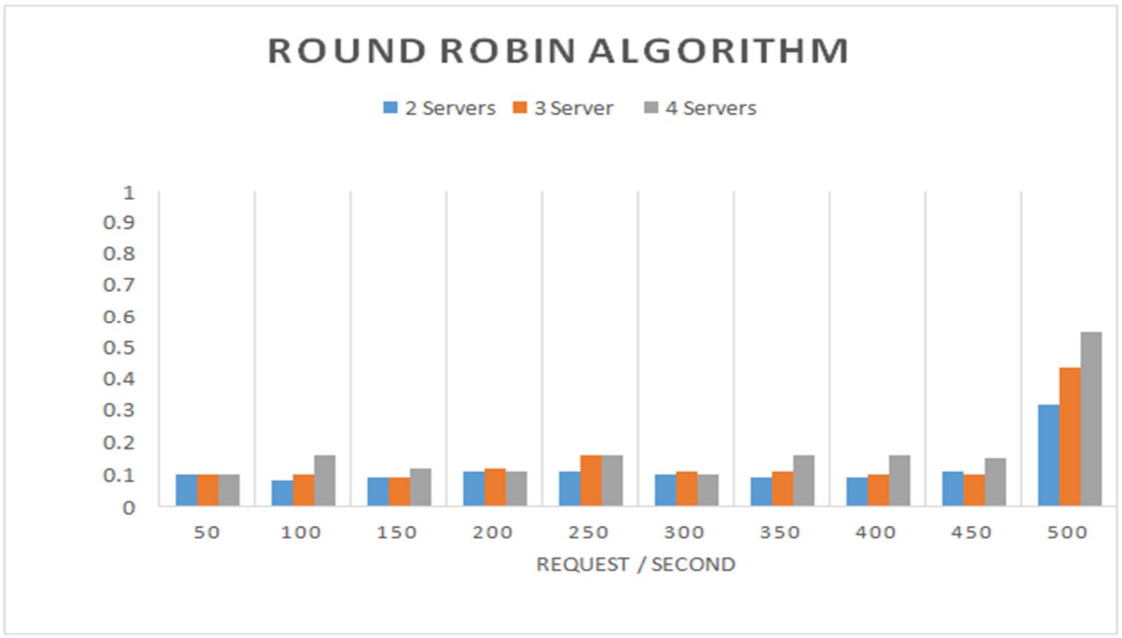


Figure 3.16 describes the results of the large CPU power usage using the Round robin algorithm.

In terms of power consumption, load balancing tends to be stable and there is no significant change. It can be seen in the number of requests from 50 to 450 requests / second and occurs on all existing servers. However, a very significant change occurred in usage on 4 servers in number request 500 request / second.

Figure 3.17 shows that CPU power usage on load balancing with least connections tends to be stable in the number of requests 50-400 requests / second with the number of servers 2 and 3 servers. On 2 server trials experienced a significant increase in 500 requests / second. Whereas server 3 experienced a very significant increase at 450 requests / second and decreased again at 500 requests / second.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

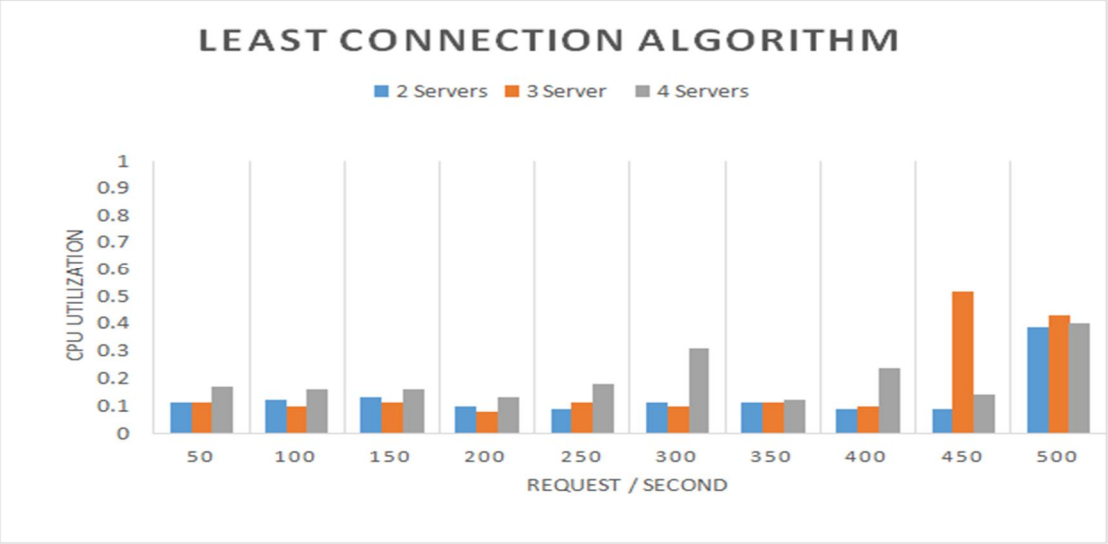


Figure 3.17 Graph of CPU Utilization Least Connection Algorithm

Figure 4.18 shows a graph of the comparison of CPU Utilization values between the two graphical algorithms generated at 50 requests / second up to 500 requests / second showing that the power required for the implementation of 4 servers in the previous algorithm or the Least connection algorithm shows a very large value. The increase tends to be volatile and volatile. This happens because the server handles the request process simultaneously with a short time interval. Then on a request to 400 requests / second, the power used has increased and decreased again at 450 requests / second. The increase in the value of CPU Utilization occurs because the process that is running the server load balancing increases according to the number of servers. This happens because the large number of queues that occur requires large resources or power to handle it.

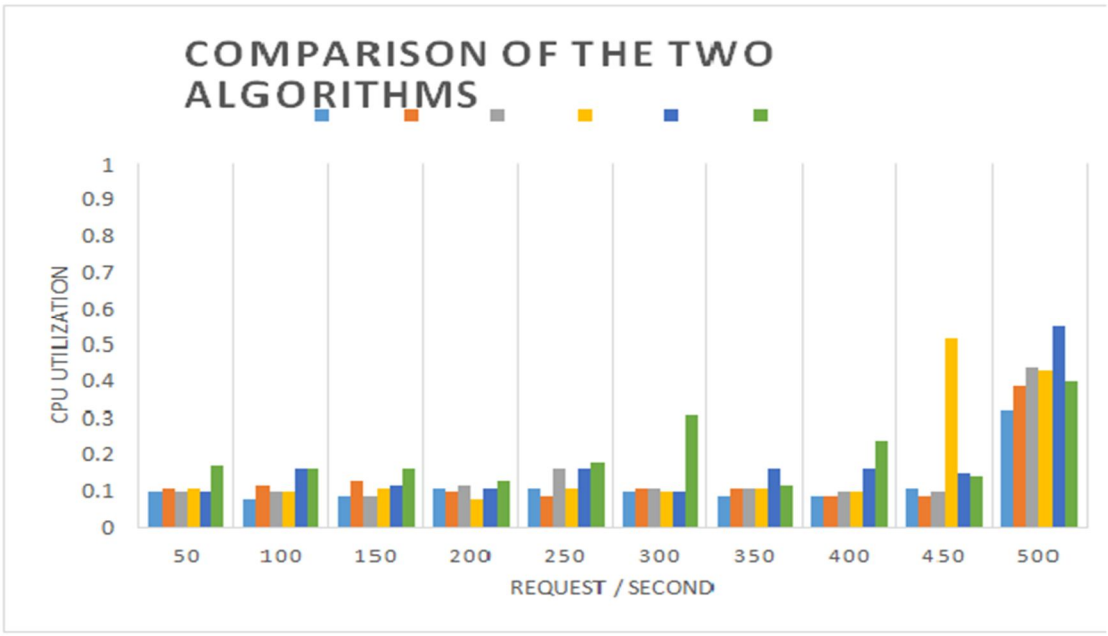


Figure 3.18: Graph of CPU Utilization of Two Algorithms

On the other hand, the Round Robin algorithm uses less power than the previous Least connection algorithm. On requests to 50-400 requests / second, power usage tends to be stable and does not experience significant changes. Significant changes only occur once, which occurs on requests to 500 requests / second when experimenting with 4 servers.

E. System Feasibility Test

System feasibility testing aims to get a direct assessment of the response to the resulting system. The stages of the system feasibility test are questionnaires and data tabulation.

1) Questionnaire

The questionnaire made contains questions related to the resulting system. The system feasibility test questionnaire uses a linkert scale by selecting five available answers, namely Very Good, Good, Good Enough, Less Good and Not Good. The weights for each answer are shown in table 3.6. After the questionnaire / questionnaire data is obtained, it will be analyzed by calculating the amount of each variable. For the complete process using a questionnaire calculation is listed in attachment 5. The following is an assessment of the test of the user for each aspect.

It can be seen that the assessment of the throughput testing aspect has a percentage value of 81.11%. Based on the assessment category in the table 4.1 the percentage values are in the qualitative interval 68% - 84%. So it can be concluded that the assessment from the throughput testing aspect is included in the "Good" category. The picture of the percentage of throughput testing aspects in the diagram is shown in Figure 3.19.

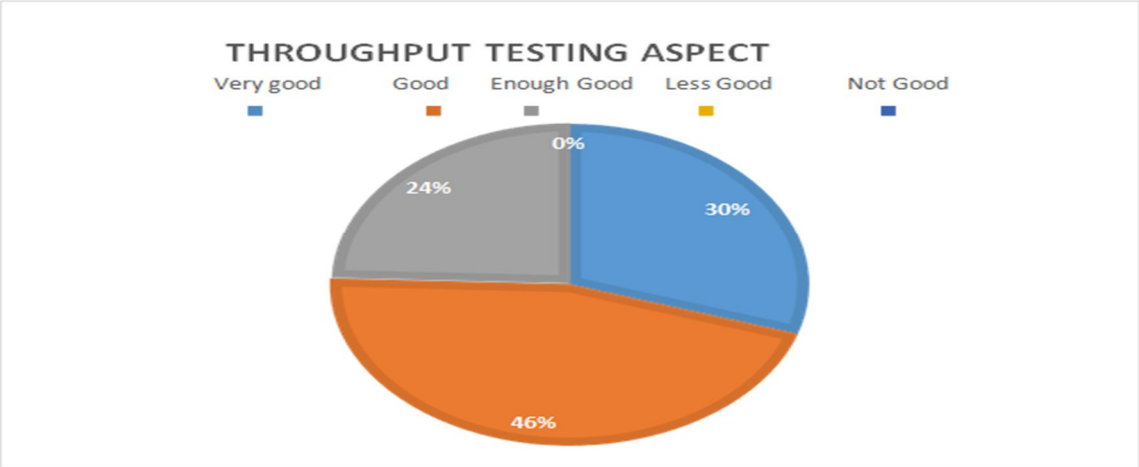


Figure 4.19: Percentage Diagram of Throughput Testing Aspects Questionnaire Results

2) Aspect of Response Time Testing

In the aspect of testing the response time in this test, which is to see the respondent's assessment of the performance of the system to serve users according to the time that has been needed. From the table, it can be seen that the assessment of the response aspect time has a percentage value of 81.78% which is in the interval 68% - 84% and is included in the "Good" category. The percentage of throughput aspects in the diagram is shown in Figure 3.20.

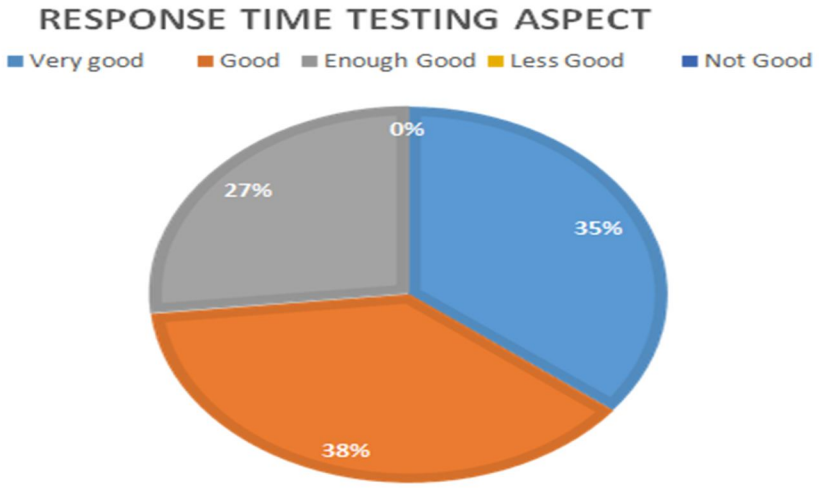


Figure 3.20: Percentage Diagram for the Aspect Response Time Questionnaire Result.

F. Research Limitations

This study has several limitations that were found. This limitation cannot be denied, even though we have made every effort to carry out valid research and data collection. The limitations of this study are as follows:

1) Weaknesses in data collection through questionnaires include respondents who seem to answer at will without reading the questions carefully and do not understand clearly the questions being asked.
2) Limitations of teachers in schools who are still unfamiliar with the workings of the Load Balancing System.
3) Limited knowledge, time, energy, and reviews in research. This makes an obstacle for writers in carrying out software development and perfect research.

IV. CONCLUSIONS AND SUGGESTION

A. Conclusion

After conducting research on load balancing with Linux Virtual Server using the Round robin algorithm at Cloud by comparing the previous algorithm and comparing the performance between 2 servers, 3 servers and 4 servers, the following conclusions were drawn:

1) The implementation of load balancing with the round robin algorithm is more reliable in optimizing throughput, response time, CPU utilization, and reducing the number of errors from the web server. Meanwhile, using the previous algorithm, namely least connection, is more reliable in optimizing the response time of the web server.
2) The addition of the number of servers can increase the value of throughput, response time and reduce errors in the implementation of the load balancing system with the round robin algorithm, whereas the previous algorithm did not increase significantly.

B. Suggestion

In further research, research on load balancing with other algorithms can be developed by synchronizing database. Testing is done by considering the data flow that is on the database server. In addition, testing can be done using a type of web server application other than Apache as a comparison on the application side.

REFERENCES

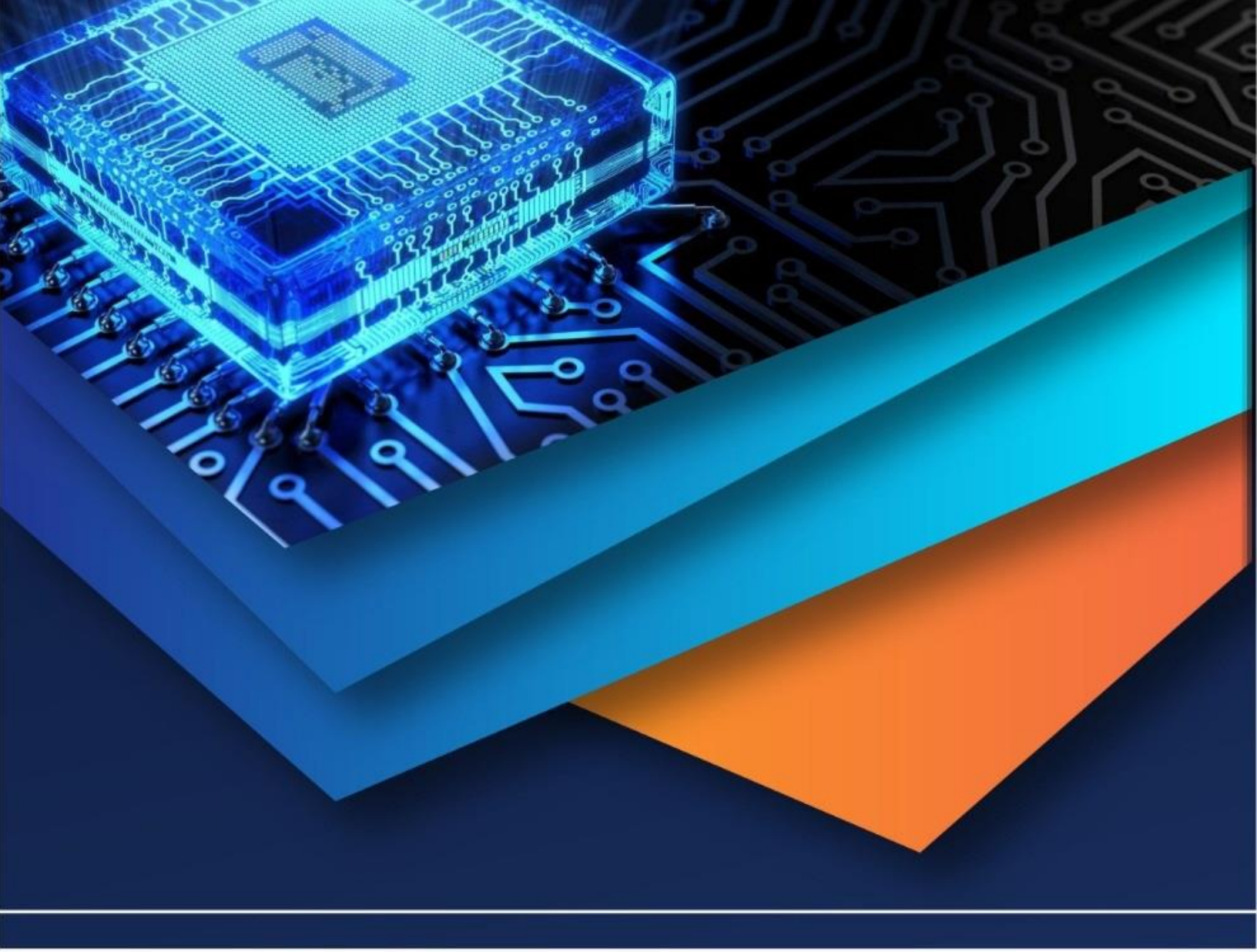
[1] https://en.wikipedia.org/wiki/Computer_network
[2] Nwobodo, Ikechukwu. (2015). Cloud Computing: A Detailed Relationship to Grid and Cluster Computing. International Journal of Future Computer and Communication. 4. 82-87. 10.7763/IJFCC.2015.V4.361.
[3] Shiv Shankar, Ashish Kumar Sharma, 2017, A Comparative Performance Analysis of Cloud, Cluster and Grid Computing over Network, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) ICADEMS – 2017 (Volume 5 – Issue 03),
[4] Haider, S., Nazir, B. Fault tolerance in computational grids: perspectives, challenges, and issues. SpringerPlus 5, 1991 (2016). https://doi.org/10.1186/s40064-016-3669-0
[5] Becker A., Zheng G., Kalé L.V. (2011) Load Balancing, Distributed Memory. In: Padua D. (eds) Encyclopedia of Parallel Computing. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-09766-4_504
[6] Garg, Atul. (2014). A comparative study of static and dynamic Load Balancing Algorithms. IJARCSMS. Volume 2. Page 386-392.
[7] Chamoli, Sushil&Rana, Deepak. (2015). Various Dynamic Load Balancing Algorithms in Cloud Environment: A Survey. International Journal of Computer Applications. 129. 14-19. 10.5120/ijca2015906927.
[8] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi and J. Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms," 2012 Second Symposium on Network Cloud Computing and Applications, London, UK, 2012, pp. 137-142, doi: 10.1109/NCCA.2012.29.
[9] Müller, Sune& Holm, Stefan &Søndergaard, Jens. (2015). Benefits of Cloud Computing: Literature Review in a Maturity Model Perspective. Communications of the Association for Information Systems. 37. 10.17705/1CAIS.03742.
[10] Qi, Lianyong&Khosravi, Mohammad &Xu, Xiaolong& Zhang, Yiwen&Menon, Varun. (2021). Cloud Computing. 10.1007/978-3-030-69992-5.
[11] Daigrepont, Jeffery & Morrow, James. (2020). Cloud Computing. 10.4324/9780429356674-2.
[12] Beigrezaei, Mahsa. (2020). cloud computing.
[13] Ali, Omar. (2020). Cloud Computing.
[14] Beri, Rydhm& Singh, Jaspreet. (2020). Cloud computing.
[15] P, Krishna Sankar& N P, Shangaranarayanee&Saravanan, K.. (2020). CLOUD COMPUTING.
[16] Manvi, Sunilkumar&Shyam, Gopal. (2021). Cloud Computing: Concepts and Technologies. 10.1201/9781003093671.
[17] Mammadova, Narmin&Akkuş, Banu&Ppp, Ppp. (2020). GRID VS CLUSTER COMPUTING. 10.13140/RG.2.2.29134.08009.
[18] Buyaa(Ed, R.. (1999). High Performance Cluster Computing: Architectures and Systems.
[19] Gupta, Sohan. (2020). Computer Architecture and organizational.
[20] Dinquel, Jerry. (2021). NETWORK ARCHITECTURES FOR CLUSTER COMPUTING.
[21] Kaushik, Shweta& Gandhi, Charu. (2021). Fog vs. Cloud Computing Architecture. 10.4018/978-1-7998-5339-8.ch020.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[22] Ağalarov, Mehran&Elizade, Cebrayil&Ppp, Ppp. (2020). Cluster Computing vs. Cloud Computing.
[23] Singh, Uhlmann, John. (2021). Efficient Job Scheduling on Cluster Computers.
[24] Boveiri, Hamid Reza. (2020). An enhanced cuckoo optimization algorithm for task graph scheduling in cluster-computing systems. Soft Computing. 24. 1-19. 10.1007/s00500-019-04520-3.
[25] Tannenbaum, Ariela&Tannenbaum, Zvi& Minor, Bryan &Frajka, Tamas&Westerhoff, John &Dauger, Dean. (2014). Clustered computer system.
[26] Manopulo, Niko. (2021). Seminar in Cluster Computing: Cluster Computing – A Step Forward to Grid Computing.
[27] Taufer, Michela&Balaji, Pavan& Matsuoka, Satoshi. (2016). Special Issue on Cluster Computing. Parallel Computing. 58. 25-26. 10.1016/j.parco.2016.09.001.
[28] Rodionov, Alexey. (2019). On Evaluating a Network Throughput. 10.1007/978-3-030-19063-7_3.
[29] Weinberg, Frankie. (2018). A Process Model of Network Throughput. Academy of Management Proceedings. 2018. 12877. 10.5465/AMBPP.2018.12877abstract.
[30] Persico, Valerio &Marchetta, Pietro&Botta, Alessio&Pescapè, Antonio. (2015). On Network Throughput Variability in Microsoft Azure Cloud. 1-6. 10.1109/GLOCOM.2015.7416997.
[31] Jechlitschek, Christoph. (2021). A Survey Paper on Processor Workloads.
[32] Adessa, Frank. (2013). System and method for processor workload metering.
[33] Martin, Philippe. (2021). The Workloads. 10.1007/978-1-4842-6494-2_5.
[34] https://www.jppf.org/about.php
[35] https://www.vogella.com/tutorials/JavaConcurrency/article.html
[36] Launay, Pascale &Pazat, Jean-Louis. (2006). A Framework for Parallel Programming in Java. 10.1007/BFb0037190.
[37] Maeda, Shin-ichi. (2021). Reinforcement Learning. 10.1007/978-3-030-03243-2_859-1.
[38] Jo, Taeho. (2021). Reinforcement Learning. 10.1007/978-3-030-65900-4_16.
[39] Yan, Wei. (2021). Reinforcement Learning. 10.1007/978-3-030-61081-4_5.
[40] Plaat, Aske. (2020). Reinforcement Learning. 10.1007/978-3-030-59238-7_3.
[41] Coqueret, Guillaume &Guida, Tony. (2020). Reinforcement learning. 10.1201/9781003034858-20.
[42] Abundo, Marco & Valerio, Valerio &Cardellini, Valeria & Lo Presti, Francesco. (2014). Bidding Strategies in QoS-Aware Cloud Systems Based on N-Armed Bandit Problems. Proceedings - IEEE 3rd Symposium on Network Cloud Computing and Applications, NCCA 2014. 10.1109/NCCA.2014.15.
[43] Ludolph, Nicolas & Ernst, Thomas & Mueller, Oliver &Goericke, Sophia & Giese, Martin &Timmann, Dagmar &Ilg, Winfried. (2019). Cerebellar involvement in learning to balance a cart-pole system. 10.1101/586990.
[44] Khoshnevisan, Davar& Schilling, René. (2016). On the Markov Property. 10.1007/978-3-319-34120-0_4.
[45] Kronfeld, B.. (2021). Relative Markov property. GlasnikMatematički. Serija III.
[46] Lecture, &Lauritzen, Steffen. (2021). More on Markov Properties.
[47] Kaushik, Shweta& Gandhi, Charu. (2021). Cloud Computing Technologies. 10.4018/978-1-7998-2764-1.ch011.
[48] Labhade, Chetan. (2018). Cloud Computing Technologies: an Overview. Journal of Advances and Scholarly Researches in Allied Education. 15. 43-48. 10.29070/15/57787.
[49] Kirubakaramoorthi, R. &Arivazhagan, D. & Helen, D.. (2015). Analysis of Cloud Computing Technology. Indian Journal of Science and Technology. 8. 10.17485/ijst/2015/v8i21/79144.
[50] Ma, Huan&Shen, Gaofeng& Chen, Ming & Zhang, Jianwei. (2015). Technologies based on Cloud Computing Technology. 1-5. 10.14257/astl.2015.82.01.
[51] S Kumar and D. Singh, Energy and exergy analysis of active solar stills using compound parabolic concentrator, International Research Journal of Engineering and Technology (IRJET), 6 (2019) 12.
[52] R. Shanker, D. Singh, D. B. Singh "Performance analysis of C.I. engine using biodiesel fuel by modifying injection timing and injection pressure" International Research Journal of Engineering and Technology(IRJET) 6 (2019) 12.
[53] A. K. Anup and D. Singh, FEA analysis of refrigerator compartment for optimizing thermal efficiency, International Journal of Mechanical and Production Engineering Research and Development, 10 (2020) 3, 3951-3972.
[54] S Kumar and D. Singh, Optimizing thermal behavior of compact heat exchanger, International Journal of Mechanical and Production Engineering Research and Development, 10 (2020) 3, 8113-8130.
[55] Dharamveer and Samsher, Comparative analyses energy matrices and enviro-economics for active and passive solar still, materialstoday: proceedings, https://doi.org/10.1016/j.matpr.2020.10.001 (2020).
[56] Dharamveer, Samsher, Anil Kumar, Analytical study of N^{th} identical photovoltaic thermal (PVT) compound parabolic concentrator (CPC) active double slope solar distiller with helical coiled heat exchanger using C_uO Nanoparticles, Desalination and water treatment, 233 (2021) 30-51, https://doi.org/10.5004/dwt.2021.27526
[57] Dharamveer, Samsher, Anil Kumar, Performance analysis of N-identical PVT-CPC collectors an active single slope solar distiller with a helically coiled heat exchanger using CuO nanoparticles, Water supply, October 2021, SCI-E Index, IWA Publication. I.F 1.275,https://doi.org/10.2166/ws.2021.348
[58] M. Kumar and Dharamveer Singh, Comparative analysis of single phase microchannel for heat flow Experimental and using CFD, International Journal of Research in Engineering and Science (IJRES), 10 (2022) 03, 44-58.
[59] ZHOU, Xue-Song & MI, Jia-Wei & MA, You-Jie& GAO, Zhi-Qiang. (2017). Cloud Computing Technology in Smart Grid. DEStech Transactions on Engineering and Technology Research. 10.12783/dtetr/icmeca2017/11958.
[60] AL-Harthy, Khoula&Shimaa, &Khadjiab, & a, Duaa. (2014). Cloud Computing Technology: SWOT Analysis.
[61] Srujana, R. &Dr.Y, Mohana& Mohan, M.. (2019). Sorted Round Robin Algorithm. 968-971. 10.1109/ICOEI.2019.8862609.
[62] A Stephen &Shanthan, Hubert &Ravindran, Daks. (2018). Enhanced Round Robin Algorithm for Cloud Computing. International Journal of Scientific Research & Management Studies. 7.
[63] M. Mostafa, Samih& Amano, Hirofumi. (2020). An Adjustable Variant of Round Robin Algorithm Based on Clustering Technique. Computers, Materials & Continua. 66. 10.32604/cmc.2021.014675.

[64] Youm, Dong. (2016). Load Balancing Strategy using Round Robin Algorithm. Asia-pacific Journal of Convergent Research Interchange. 2. 1-10. 10.21742/apjcri.2016.09.01.
[65] Zhu, Liangshuai& Cui, Jianming&Xiong, Gaofeng. (2018). Improved dynamic load balancing algorithm based on Least-Connection Scheduling. 1858-1862. 10.1109/ITOEC.2018.8740642.
[66] O'Rourke, Patrick & Keefe, Mike. (2001). Performance Evaluation of Linux Virtual Server.. 79-92.
[67] Gwena, Tinashe. (2021). Gollach : configuration of a cluster based linux virtual server
[68] Ungermann, F., Kuhnle, A., Stricker, N. and Lanza, G., 2019. Data Analytics for Manufacturing Systems–A Data-driven Approach for Process Optimization. Procedia CIRP, 81, pp.369-374.
[69] Azahara. 2016. How Data mining is used to generate Business Intelligence. Geographica. [online] retrieved from http://www.blog-geographica.com/2016/11/15/how-data-mining-is-used-to-generate-business-intelligence/ [Accessed on 3 march 2021]
[70] Jenke, R., 2018. Successful data applications: a cross-industry approach for conceptual planning. Journal of Business Chemistry, 15(2).
[71] Diba, K., 2019. Towards a comprehensive methodology for process mining. In Proceedings of the 11th Central European Workshop on Services and their Composition, Bayreuth (pp. 9-12).
[72] Massmann, M., Meyer, M., Frank, M., von Enzberg, S., Kühn, A. and Dumitrescu, R., 2020. Framework for Data Analytics in Data-Driven Product Planning. Procedia Manufacturing, 52, pp.350-355.
[73] Ghavami, P., 2019. Big data analytics methods: analytics techniques in data mining, deep learning and natural language processing. Walter de Gruyter GmbH & Co KG.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)