Ilkogretim Online - Elementary Education Online, 2021; Vol 20 (Issue 5): pp. 9032-9054
http://ilkogretim-online.org
doi: 10.17051/ilkonline.2021.05.994

Load Balancing Optimization Using Adaptive Algorithm with Round Robin Technique In Cloud Computing

Jyoti Rai, Department of Computer Science and Engineering, RD Engineering College Ghaziabad, Uttar Pradesh, India

Appurva Tripathi, Department of Computer Science and Engineering, RD Engineering CollegeGhaziabad, Uttar Pradesh, 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.

Keywords: Cloud Computing, Computer Networking, Server, Load Balancing, Linux Virtual Server, Direct Routing, Round robin, Httperf.

1. 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 ajob, 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 ofsystems 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 balancingmethod 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. The development of technology, especially in the field of information technology, has made all areas in an economic system no longer only need information technology as a supporting tool, but has become one of the main pillars in designing an economic system as a whole. Information technology has metamorphosed into an important basis on which matters - substantial companies, whether small companies, to multinational companies, are documented and stored in a database unit. In practice, this database, along withother applications, often requires a lot of CPU resources, and requires high-value maintenance. For all of these things, the Cloud Computing system is considered very useful and useful for acompany, as well as personal users. This is due to the cloud computing system in the form of a third party system resource that can be accessed online, so the use of these applications can be rented from third parties with a very flexible system.

The development of conventional resource use with the system buying a resource, installinga resource, and continuing with the execution and maintenance of a resource, can be cut significantly by the use of cloud computing. Resources are no longer classified as "Resource Ownership", but with this technology a resource can be classified as "Resource Use", because the user is no longer required to have a specific resource. All required resources can be utilized with a service based system, where users only need to rent these resources according to the needs and requirements of the users themselves [9].

In its use, cloud computing can be classified by [10-16]:

1. Infrastructure as a Service (IaaS) is the oldest concept in which the implementation is mostly done, starting from using or leasing a network for Internet access, Disaster Recovery Center services, etc.
2. Platform as a Service (PaaS), which is a concept similar to IaaS. However, the platform here is the use of the operating system and its supporting infrastructure. What is quite famous is the service from the Force.com site and the services of the server vendors.
3. Software as a Service (SaaS), which is one level above PaaS and IaaS, where what is offeredis software or a certain business application. The most recent examples are SalesForce.com, Service-Now.com, Google Apps, etc.

The greater the number of users who access a cloud, the heavier the burden on web serversis in receiving requests. The large number of requests on web servers will result in overloadingand even the possibility of the server going down. On the other hand, a reliable web server should be able to handle requests from large users. Therefore, a server load balancer is needed that will regulate and share the server workload. This round robin method is a method for dividing the server load in turns and sequentially from one another to form a loop. By using theLoad Balancing method, it is the right and effective solution to handle busy server loads. So forthat the Least Connection method will be compared with the new method, with the Round Robinmethod.

RELATED RESEARCH

CLUSTER COMPUTING

Initially, software was written for serial computing. Serial computing works by breaking the problem into sequential instructions. These instructions are executed sequentially one by one. Usually execution is carried out on a solitary central processing unit and only one order is accomplished at a time [17].

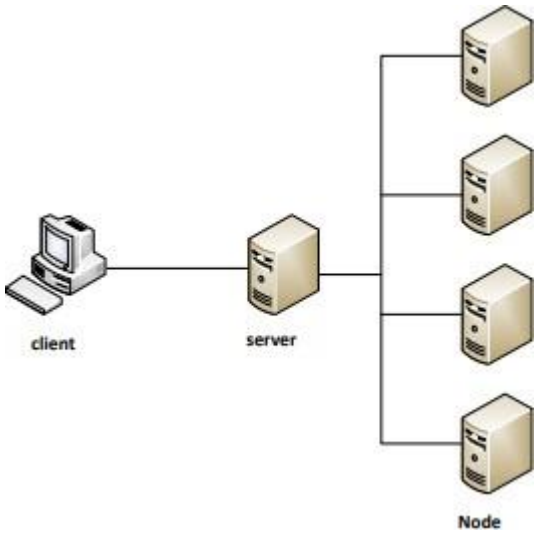

Figure 1 Cluster Computing Architecture

The first approach in parallel computing is to create a computer that consists of many processors. However, this approach is costly. Another approach is to use multiple computers connected by a LAN network. Parallel computing resides in the computing area of a single machine with multiple processors or processors with multiple cores. Meanwhile, computing that involves multiple computers to work with each other to handle the same task is called cluster computing. Figure 1 shows a general cluster computing architecture. Cluster computing is formed by several computers that have the same or almost the same specifications, which arelocated in a place connected by a fast LAN network connection. Cluster computing aims to provide more availability, reliability, and scalability when compared to a single system (Microsoft, 2013). In grid computing, each node may execute different tasks. However, in cluster computing, each node performs the same task. Some of the advantages of using cluster computing are that it saves money and time. In theory, shifting computation that requires a long computation time to multiple computers will shorten the time it takes to solve a problem. For example, in a program there is a procedure p which uses 90% of the program's execution time. The high execution time of procedure p can be hidden by

breaking the procedure into 10 sub procedures, each of which will take 9% execution time. With a shorter time, the costs used arealso smaller. In addition, cluster computing can also solve complex problems that cannot be solved by computing with a single computer, especially on computers with limited RAM [18-23].

Scheduling in Cluster Computing

Scheduling problems in cluster computing and grid computing have become a research topic by many researchers. Job scheduling itself is the most important component in cluster computing. In a grid computing environment, scheduling is said to have NP-hard computational complexity [23-25]. With the complexity of NP-hard computation, various researchers proposes heuristic and meta-heuristic methods for scheduling jobs on grid computing [26]. Theheuristic method was chosen because it does not require an optimal solution, but allows gettingan efficient solution in a short time. With scheduling problems as complex problems, researchers are moving into the areas of NP-completeness, approximation algorithms, and heuristic methods to obtain optimal solutions. First-come-first-serve (FCFS) is a simple job scheduling that is often found in real cluster computing systems in the field, but researchers usually consider this method to be inadequate [23-32]. In JPPF, FCFS is represented by a manual distribution strategy. Since job scheduling performance in cluster computing is highly dependent on workloads, adaptive scheduling allows the best results for scheduling problems. According to [23], the most important matrix for interactive jobs is response time. While the system utilization matrix is the most important matrix for batch jobs;

Network Throughput

The network throughput parameter indicates the speed at which data transfers can be made between the server and the node. This becomes important because there is a condition in the network in the form of a different throughput on each computer. The difference can be caused by differences in the hardware used or from the application side. This throughput parameter is obtained by calculating the amount of time used to send job data, both when sending and receiving job results. To calculate throughput the following equation is used [24-26]:-

$$Throughput = \left(\frac{b}{s}\right) = \frac{amount\ of\ data\ (bits)}{ransmission\ time\ (\mu s)}\ (eq.1)$$

One method for envisage the throughput of a outsized number of TCP shift is to measure the throughput of preceding transmit in the identical passageway. The History-Based (HB) technique of prediction is almost the same as traditional time series forecasting, in which unknown random samples are worn to envisage the worth of the future development. The HB method allows to be applied in the case of large numbers of TCP transfers that occur repeatedlyon the same path [24-26]. Moving Average (MA) is a family of simple linear predictors. MA isdefined by :-

$$X_{i+1} = \frac{1}{n} \sum_{k=i-n+1}^{i} X_k \quad (eq.2)$$

Where X_{i+1} is the predicted value, X_i is the actual observed value. If n is too small, the predictor cannot produce good predictions, meanwhile if n is too huge, the forecaster cannot acclimatize properly.

Processor Workload

The workload of a node shows the computational load that is being done handled by a node. The workload can be caused by the existence other applications that are also running on a node.Workload can also caused by another task the node is working on. If the workload of a node ishigh, it will affect performance compute the node against the given task. In the dynamic scheduling process, goals of importance have knowledge of a node's CPU usage is to measure how much a process is able to use CPU resources on fixed time intervals [27-29]. For example, if provided 50% CPU resources, a process should be able to get 50% of the available time-slice. If only 50% of the time-slice is available, a process is estimated to take 2 times longer if compared to the condition the CPU is not currently having a heavy compute load other. The percentage of CPU resources is defined as:-

$$available_{cpu\_load\_averag} = \left(\frac{1.0}{load_{avg} + 1.0}\right) * 100\% \ (eq.3)$$

which shows the percentage of CPU time available for new processes.

Cloud Computing Mechanism

Infrastructure as a Service (IaaS)

This concept is normally distinct as the use of computer infrastructure as a service. This service includes the basic capabilities of computers, databases, networks, load balancers, and more. This reduces the need for a company to have a datacenter. Instead of buying a set of servers, software, datacenter rooms, or network devices, a company can access those needs from a cloud computing service provider. Services sold by this company are usually in the form of pay as you go concept and the amount of resources used. IaaS can run on clients with the help of software APIs. In general, RESTful and SOAP APIs software are used to access this infrastructure. IaaS relies heavily on virtualization technology on the platform to run a particular virtual machine. The implementation of IaaS can include computer networks in the form of firewalls, load balancers needed at the datacenter to ensure a high level of security and performance of an application. Cloud service providers such as Amazon create large user communities where Amazon service users have created an API concept that runs on the web service. Other companies such as Google provide a browser interface, which provides flexibility for users to build an infrastructure with a drag-and-drop widget system that represents the CPU, database, storage media, load balancer, firewall, and others [30-36]. These various service providers differentiate their services on the number of Operating Systems on the platforms they support, software installed on the operating system, prices, and service agreement levels.

Platform as a Service (PaaS)

Platform as a service (PaaS) is a computing concept where all existing facilities are needed to fulfill the entire life cycle of an application, namely building applications, conducting application tests, and finalizing and completing an application on an Internet network from a cloud, without requiring a software download and installation process. from the developer and the user side. According to wikipedia, this is called cloudware. PaaS is the next step of the system we know as mashups. Users can create applications by adding features available from google maps, google calendar and other web services to make this feature part of their application. In PaaS a developer does not

need to write every source code for every application he wants to develop. The PaaS system contains the facilities and tools needed for designing an application, developing applications, testing, installing applications, and hosting, in line with application services such as team collaboration, web service integration, database integration, security, scalability, storage, and application management. This concept is similar to the web version of the visual basic language, where the developer can visually design an application and add code as needed. In a traditional concept of developing a software, an application is written in one system, goes through a testing phase on another system, and is attached to anothersystem for distribution. Apart from the huge funds required to build, configure, and maintain these different systems, applications must always be monitored, as other funds are required for this. In a PaaS system, the entire software lifecycle is carried out on the same system, and funds spent are significantly reduced in the areas of development and maintenance, distribution, and workmanship risk. PaaS gives flexibility to developers to create software they want, without having to think about the support systems that have to be created, complicated configurations, and large costs [30-36]. Another characteristic of PaaS is the fact that PaaS provides integration to other Web Services that are not covered by PaaS. For example, applications developed on the basis of Google AppEngine can be linked with other applications on the Cloud.

Software as a Service (SaaS)

This concept can be defined as the delivery process of an application via the Internet. Applications created as web applications or services can be accessed by users through a browserinterface or an interface provided by the service provider. Some of the services offered are available free of charge. Most of the client server based applications can be created through thisconcept. The browser functions as a client. Other services, and non-browser applications also act as clients. A service can be placed in any datacenter as long as it is not connected to the network. Companies such as Google, Microsoft and Yahoo provide general services such as mail, calendar, mapping, and others from their datacenters [30-36]. The SaaS concept eliminates the need to install and run an application on the user side. This concept also reducesthe burden on users in terms of application maintenance and support. The downside lies in the loss of the user's ability to change the software version. The main problem with upgrading from a conventional software version is the data compatibility problem with the new software version. In the SaaS concept, the data is stored together with the software on the supercomputer owned by the service provider. It is the responsibility of the service provider to ensure that if there is a change in the version of an application, the existing data can be fully compatible with the new version of the application. SaaS reduces the costs that users have to buy a software withthe concept of monthly rental owned by service providers. If the software is used as a service, the software cannot be hijacked. From the vendor side of SaaS has the perfect ability to provide maximum protection to intellectual property owned [30-36]. Applications such as Customer Relationship Management (CRM), video conferencing, human resources, IT service management, accounting, IT security, web analytics, web content management, e-mail, and calendars are applications that are the pillars of success of the SaaS concept. Several SaaS service providers have developed new applications such as Billing as a Service, and Monitoringas a Service. The difference between SaaS and Internet-based applications prior to the SaaS generation is that SaaS was developed specifically to maximize web technologies such as browsers, and provide users with an alternative interface. Users do not need to install the source code for each

application. With SaaS, the application is now a service that can be accessed from a browser and via the APIs of other services.

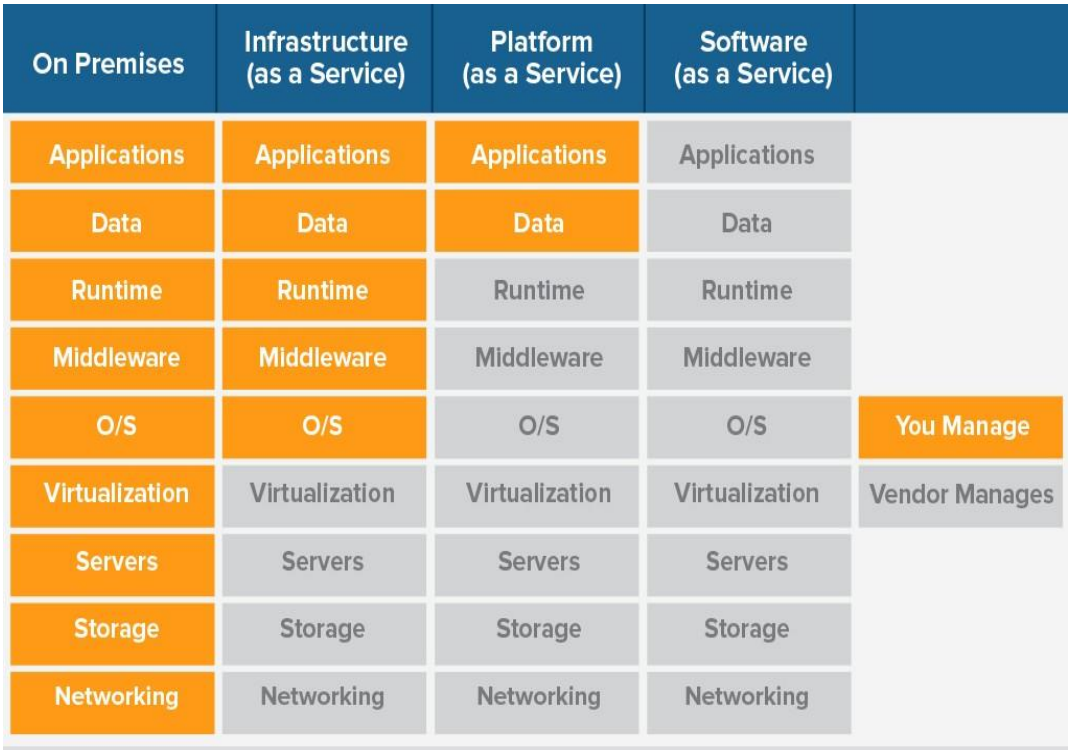

Figure 2. Cloud Computing Structure

With the classification of the cloud computing mechanism of the provided resource types as mentioned above, we can see that the implementation of SaaS can be placed on the basis ofIaaS or PaaS services. With the higher the level of service, from PaaS to IaaS the level of resource sharing and the level of resource utilization increases. At the IaaS level, users can create and run any supported operating system, and then install all the software they need to use. At the PaaS level, developers use services to create applications using the required software development kit but limited in terms of interface, language and features offered by PaaS serviceproviders. At the SaaS level, end users are limited to using specific applications offered by theprovider, and have limited capabilities to modify the service interface [30-36].

In terms of the benefits of each level, we can see that the biggest advantage is at the SaaS level. The SaaS level provides the highest possibility of sharing computer resources, from hardware to software that is shared among multiple users. Users don't need to buy a license at a fixed price, and only need to pay a license per use when they use the application. There are no upgrades or patches for users to worry about and no maintenance required for security backups and more. From the application developer point of view, patching and upgrading becomes easier because only a centralized upgrade needs to be done. This enables the patch and upgrade process to be more effective and efficient [30-36]. At the IaaS level, users benefit from being able to use hardware resources according to their usage needs. At this level it is the user'sright to use the software needed to configure and use this method as desired.

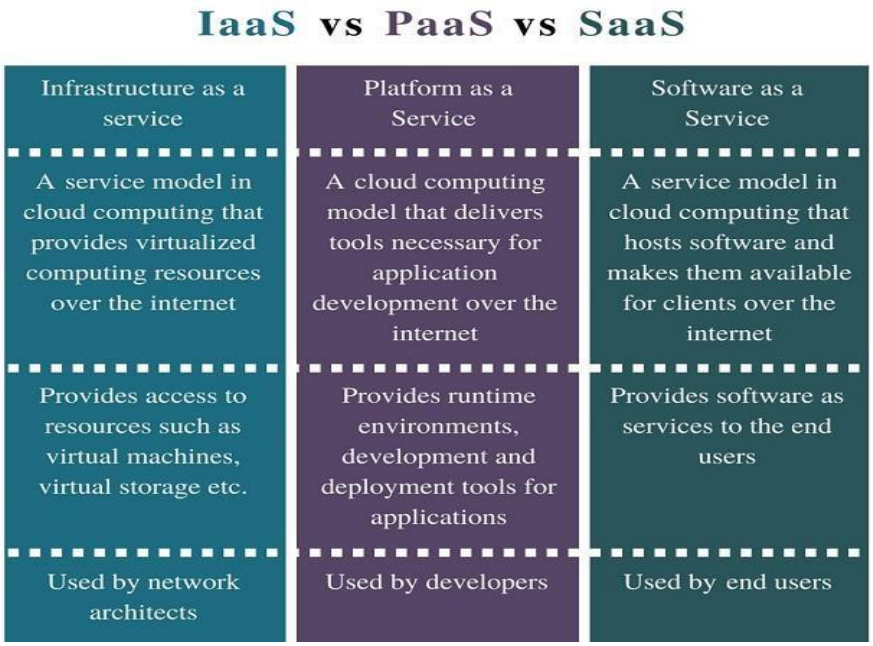

Figure 3. Comparison of the use of PaaS, SaaS, IaaS

The picture above describes the comparison between the benefits obtained by using thecloud compared to the highest capabilities that can be done in a cloud architecture.

Load Balancing

Load balancing is the capability to decrease the burden from processes for an request to numerous diverse systems so as to augment processing capabilities on arriving requirements. Load balancing determination throws a quantity of dispensation from requirements to one system to other systems which will be handled concurrently. Load balancing has the benefits of dropping the quantity of dispensation that must be executed by the major receiving server, allows the receiving server to handle extra requests and improving its performance due to less resources on the main receiving server, and more devices processing all loads. Load balancing implements several scheduling methods that determine which direction the server requests fromthe client to be forwarded. The following are the advantages of load balancing techniques [5- 8]:

- Flexibility: The server is not the core of the system and the main resource, but is part ofthe many servers that form the cluster. This means that the per-unit performance of thecluster is not taken into account too much, but the performance of the cluster as a whole.Meanwhile, to improve performance and clusters, new servers or units can be added without changing units.
- Scalability: The system does not require redesign of the entire system architecture to adapt the system when it becomes a change in system components.
- Security: For all traffic that passes through the load balancer, security rules can be implemented easily. With private networks used for real servers, IP addresses will not be accessed directly from outside the cluster system.
- High-availability: Load balancers can know the real server condition in the system automatically, if there is a real server that dies it will be removed from the real server, and if the real server is active again it will be included in the real server list.

Round Robin Algorithm

Round robin algorithm is the simplest algorithm and is widely used by load balancing tools. This algorithm divides the load by taking turns and sequentially from one server to another so as to form a cycle. This algorithm processes the queues used by load balancing devices. This algorithm divides the load by taking turns and sequentially from one virtual server to another virtual server to form a cycle. This algorithm processes the queue and rotates it in turn. The process will get a quota of time quantum. Time quantum is the time period allowed in a process that is running in a system or the amount of time used in queuing data processing. Of course this process is very fair because there is no priority process, all processes get the same time quota, namely (1 / n) the value of n is a queuing process, and does not wait long from (n-1) q where q is 1 quantum long. The provisions of the Round robin algorithm are as follows [37- 40]:

- If the previous process has not been completed, the process becomes runnable or the processing is transferred to another or the next process.
- If the quantum time has not run out and the process is still running (completion of the I / O operation), the process will be blocked and processed will be carried over to the nextprocess.
- If the quantum time has not run out and the process has finished then process terminatedand continue to other processes.

The problem that occurs in the Round robin algorithm is determines the time quantum magnitude. If time quantum is specified too small, then part of the process is not completed in 1 quantum. That matter causes a lot of switches (process switching that occurs), in fact The CPU takes time to switch from one process to another (context switches time). Conversely, if the time quantum is too large then, the round robin algorithm will run like the First Come Firstalgorithm Served. The ideal time quantum is that 80% of the total process has a CPU burst timesmaller than 1 time quantum.

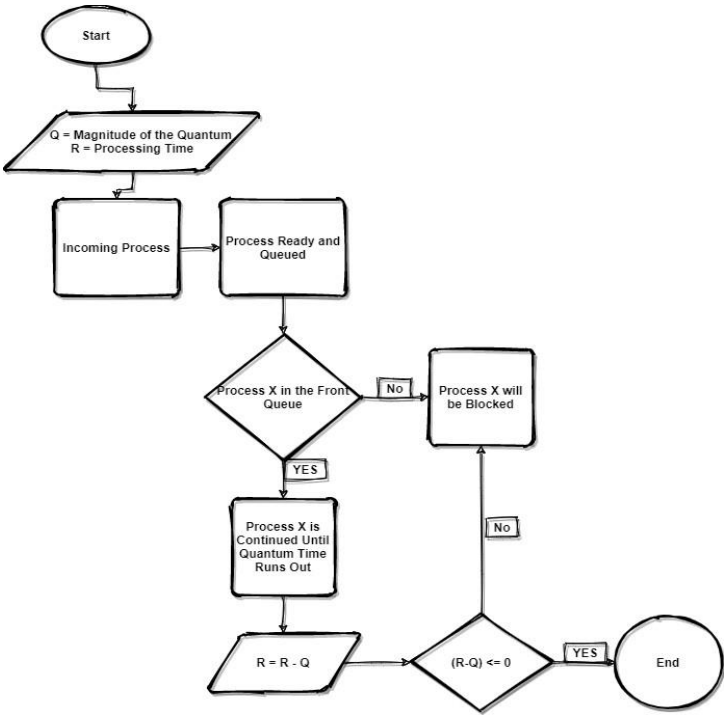

Figure 4. Round Robin Flow Model

From the flowchart above, it can be explained that the value of Q is the value the magnitudeof the quantum and the value of R is the value of processing time. When the process is entered then the process will be sorted in queue (ready), if the queue has reached the front queue then the process will be executed until the quantum time is up, if the queue hasn't reached the forefront then the queue will blocked and will be reordered into the queue. After the queuing process run until the quantum time runs out then the value of R (processing time) = R (time process) - Q (quantum time), if the value (R-Q) is less than or equal to 0 eat the queuing processis complete. However, if the value (R-Q) is more than or equal with 0 then the process will be blocked and will be sorted back to queue. To know how to calculate the Round scheduling robin. The following is an example of a Round robin scheduling.

Table 1. Example of Calculation of the Round robin Algorithm where Quantum = 2

PROCESS	ARRIVAL TIME	BURST TIME
P1	0	7
P2	2	5
P3	6	4
P4	10	2

In the table above that there are 4 processes each process has Arrival Time and Burst Time each have a value. Time comes is the time where the process begins queuing, whereas Burst Time is a possibility time used to process queues in one process. Value on Quantum is the realprocessing time.

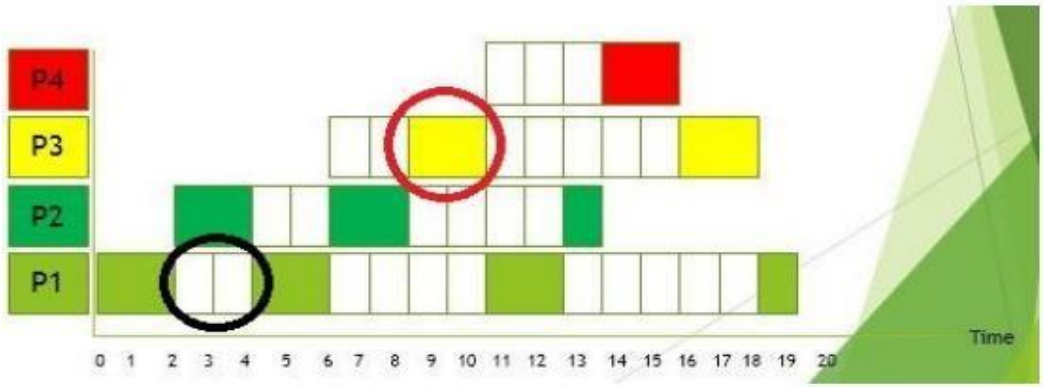

Figure 4. Queuing Results from the Round Robin Algorithm

From the image of the queuing result of the Round robin algorithm, it can be explained thateach process gets a chance of 2 milliseconds of quantum time, because P1> = time quantum then P1 will be processed during quantum time, the rest of the P1 -time quantum will be queued back and then it will switch to P2, P2 will be processed like P1 and the rest will be queued backand so on up to P4 to P <time quantum. In the Round robin algorithm, no process is carried outin more time than the time quantum provided. If there are n processes with a time quantum of q, then each process will get a time of 1 / n with a process of q. And the process will wait at least (n-1) x q for the next process. For example, there are 4 processes with a time quantum of20 ms, so each process gets a time of 20 ms every 80 ms. So the performance of the Round robin itself depends on the size of the time quantum.

Least Connection Algorithm

Least Connection Algorithm is a scheduling algorithm by directing network connections from servers that are currently active with the least number of connections. This scheduling is one ofthe dynamic scheduling algorithms, because real servers require calculations of dynamically active connections. This algorithm is good to use for smooth distribution when there are manyrequests [41].

PROPOSED ALGORITHM

Architecture Description

Using a qualitative approach and investigating 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.

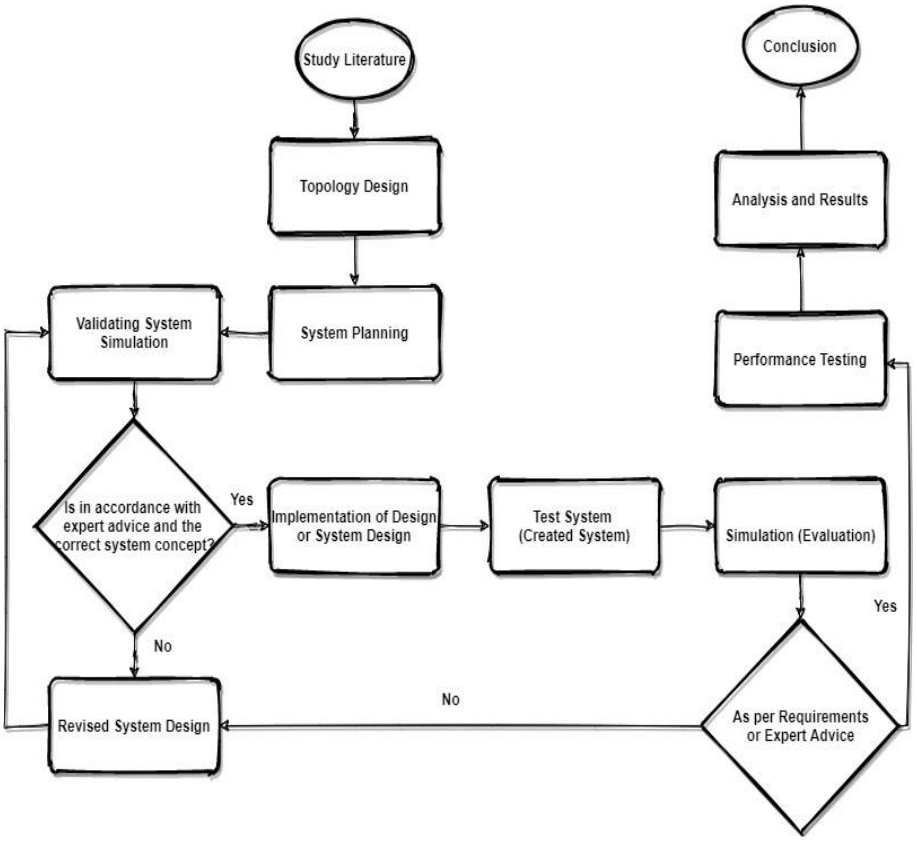

Figure 5. Research Procedure Flowchart

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 accordancewith the concept of the system. The grid of the observation table is as follows:

Table 2. Grid of Observation Sheet of Load Balancing System

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

Data Analysis Technique

Analysis of data obtained from the study includes a system feasibility test and observation tablesfrom 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:

$$P = \frac{n}{N} * 100\% \ (eq.4)$$

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:

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

b. Determine the lowest percentage number with the formula:

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

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

Table 3. Interval of Qualitative Score Categorization

NO.	PERCENTAGE	CRITERIA
1	84% < SCORE ≤ 100%	VERY GOOD
2	68% < SCORE ≤ 84%	GOOD
3	52% < SCORE ≤ 68%	PRETTY GOOD
4	36% < SCORE ≤ 52%	NOT GOOD
5	20% < SCORE ≤ 36%	NOT GOOD

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.

RESULTS AND SIMULATION

Results 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.

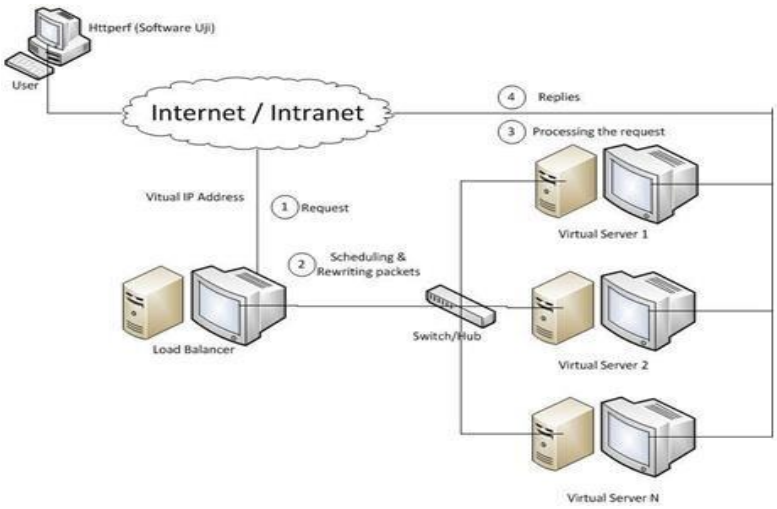

Figure 6. Details of the processes that occur in the LVS Direct Routing System

- In process 1 in Figure 7 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:

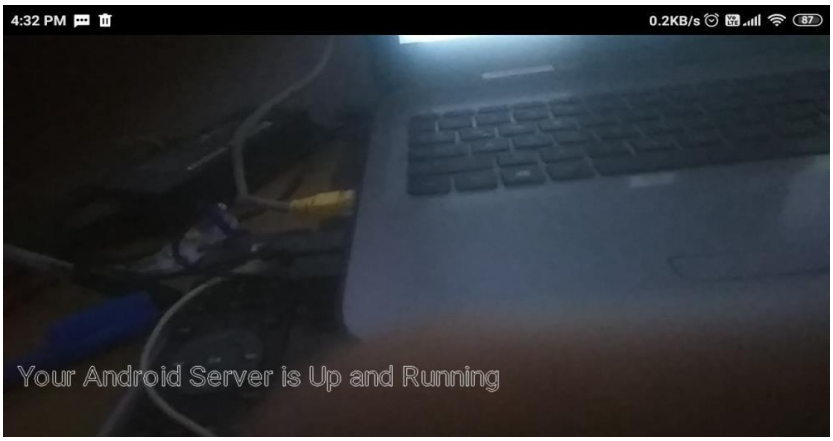

Figure 7. Custom Server (Web Server) over the Cloud Resource

Status	Connected
Signal strength	Excellent
Link speed	65Mbps
IP address	fe80::da63:75ff:fe41:b914 10.0.0.3
Subnet mask	255.255.255.0
Gateway	10.0.0.1

Figure 8. Gateway Status with Link Speed

The above figure 8 is depicting the subnet mask with Gateway IP to connect to Balancerover the cloud resource.
The following is an example of the results of the above test with HTPperf.

```
httperf --hog --timeout=5 --client=0/1 --
server=10.129.0.134 --port=80 --uri=/ --rate=50
--send-buffer=4096 --recv-buffer=16384 --num-
conns=100 --num-calls=1
Maximum connect burst length: 316
Total: connections 5151 requests 5122 replies 4390
test-duration 15.468 s
Connection rate: 333.0 conn/s (3.0 ms/conn, <=1022
concurrent connections)
Connection time [ms]: min 19.1 avg 1637.0 max
7369.7 median 1365.5 stddev 943.4
Connection time [ms]: connect 603.0
Connection length [replies/conn]: 1.000
Request rate: 331.1 req/s (3.0 ms/req)
Request size [B]: 66.0
Reply rate [replies/s]: min 133.0 avg 292.5 max
451.4 stddev 159.2 (3 samples)
Reply time [ms]: response 1038.7
transfer 0.0
Reply size [B]: header 281.0 content 375.0 footer
0.0 (total 656.0)
Reply status: 1xx=0 2xx=0 3xx=4390 4xx=0 5xx=0
CPU time [s]: user 0.33 system 14.08 (user 2.1%
system 91.0% total 93.2%)
Net I/O: 203.2 KB/s (1.7*10^6 bps)
Errors: total 15610 client-timo 761 socket-timo 0
connrefused 0 connreset 0
Errors: fd-unavail 14849 addrunavail 0 ftab-full 0
other 0
```

Figure 8. Gateway Status with Link Speed

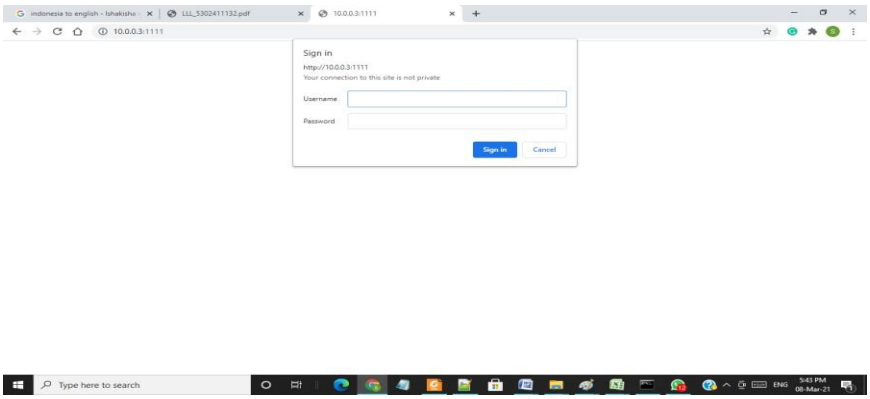

Figure 9. Request from Client to Access the Resource Over the Cloud Server

The above figure 9 using the web browser sends the request to access the resource over the cloud resource where the server is running on port vide port number 1111. Subsequently the server is prompting for security credentials to establish the trust relation.
In the second process in Figure 4.5, the load balancer is the core of the system, where the algorithm can be changed from one algorithm to another. In this study, the algorithm used is the Round robin algorithm by comparing the least connection algorithm. The following is an example of the configuration of the Least connection algorithm with 3 virtual servers:
ipvsadm -a -t 10.0.0.134:1111 -s lc
ipvsadm -a -t 10.0.0.134:1111 -r 10.0.0.10:1111 -m ipvsadm -a -t 10.0.0.134:1111 -r 10.0.0.1280 -m ipvsadm -a -t 10.0.0.134:1111 -r 10.0.0.30:80 –m
After completion, the next step is to test the test value the same as in process 1 according tothe request value. Then after the test value has been obtained, the next process is to change the algorithm as a research source, namely the Round robin algorithm by removing the IPVS Leastconnection first.

delete ipvsadm –Z LVS packet data# delete IPVS table ipvsadm –C
using the new Round robin algorithm with 3 virtual servers. ipvsadm -A -t 10.0.0.134:1111 -s rr
ipvsadm -a -t 10.0.0.1111:80 -r 10.0.0.10:1111 -m ipvsadm -a -t 10.0.0.134:1111 -r 10.0.0.20:1111 -m ipvsadm -a -t 10.0.0.134:1111 -r 10.0.0.30:1111 –m
After finishing configuring with the Round robin algorithm, the next process is to test withHTPperf like process 1.
From the several sections, it can be found the parameters needed for testing.

Throughput
Throughput based on the HTTF manual can be taken from NET I / O which is the average network throughput which has units of KB (kilobytes) per second and Mb (megabits) per second.

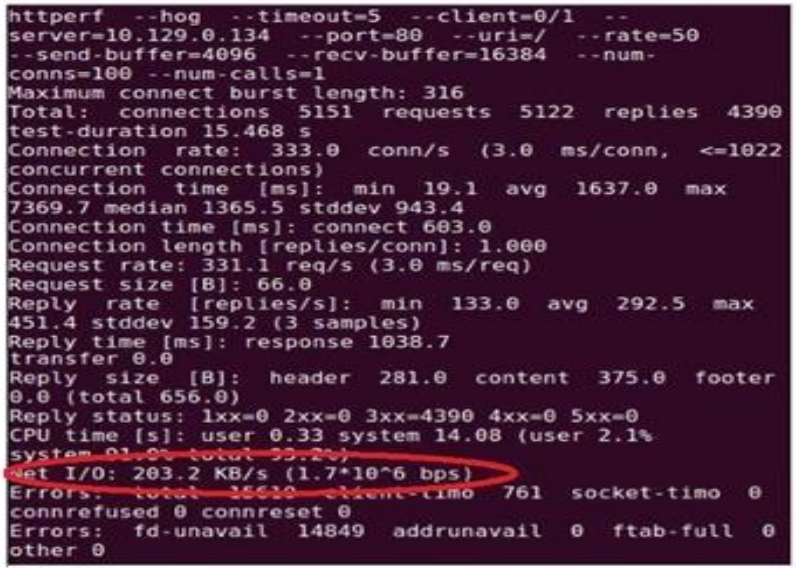

Figure 10. Throughput Parameters

Response Time
Response time obtained from the Reply Section

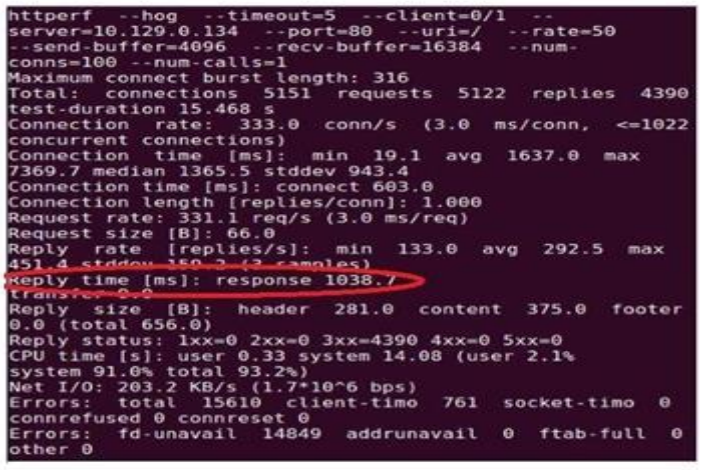

Figure 11. Parameter Response Time

Error
Error obtained from the Error Section on conn-refused and conn-rest.

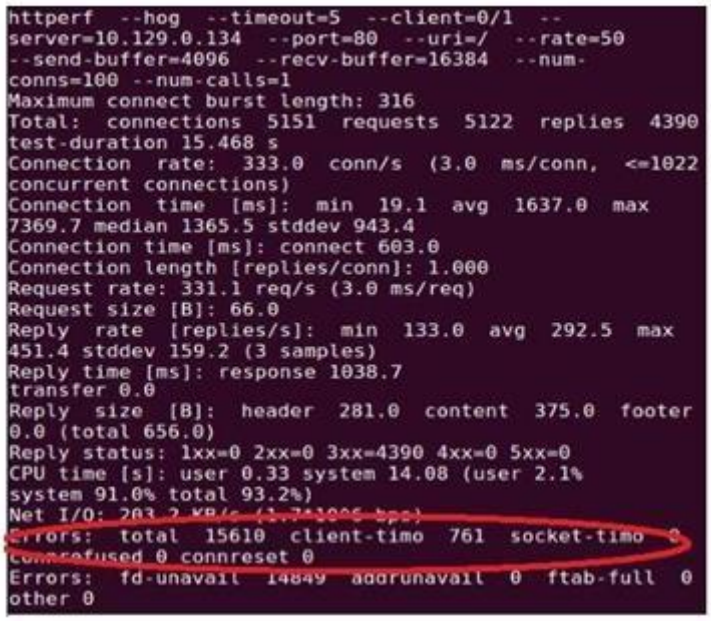

Figure 12. Parameter Error

CPU Utilization

CPU Utilization obtained from the CPU time of the user.

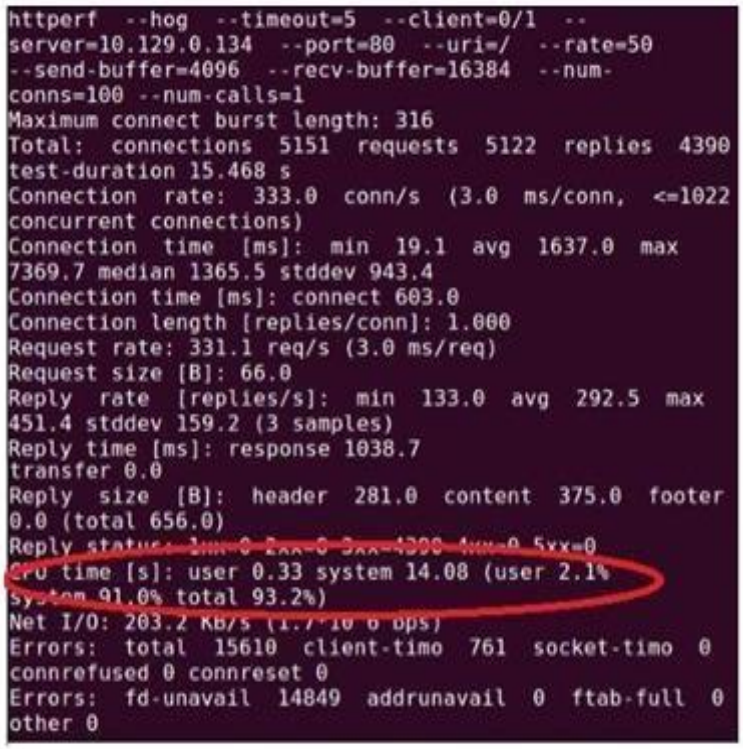

Figure 13. CPU Utilization Parameters

Throughput Parameter Analysis

The throughput parameter in this study represents the number of requests that can be respondedto 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. Data from theresearch results are attached and presented in tabular form, as in table 4.

Table 4. Throughput Parameter Test Results

REQUEST/SEC	*THROUGHPUT* (KB/SECOND)					
	2 RR	2 LC	3 RR	3 LC	4 RR	4 LC
50	5,215	5,172	5,213	5,214	5,212	5,216
100	2,364	2,203	2,383	2,299	2,363	2,428
150	1,605	1,547	1,636	1,622	1,581	1,605
200	1,213	1,208	1,156	1,071	1,215	1,218
250	0.868	0.552	0.986	0.811	0.934	0.890
300	0.666	0.686	0.671	0.658	0.693	0.676
350	0.571	0.562	0.560	0.538	0.575	0.491
400	0.482	0.482	0.464	0.484	0.495	0.472
450	0.425	0.421	0.418	0.417	0.426	0.415
500	0.369	0.361	0.316	0.365	0.377	0.373
Average	**1,377**	**1,319**	**1,380**	**1,347**	**1,387**	**1,378**

Table 4 shows the comparison of the throughput produced in this study. Implementation ofserver load balancing with the Direct Routing method and using the Round robin algorithm has a better average throughput than the least connection algorithm in all load balancing implementations with 2 servers, 3 servers, and 4 servers. Throughput value in the research results above shows that it is directly proportional to the number of servers used, this proves that increasing the number of servers can increase the throughput value of server load balancingimplementation.

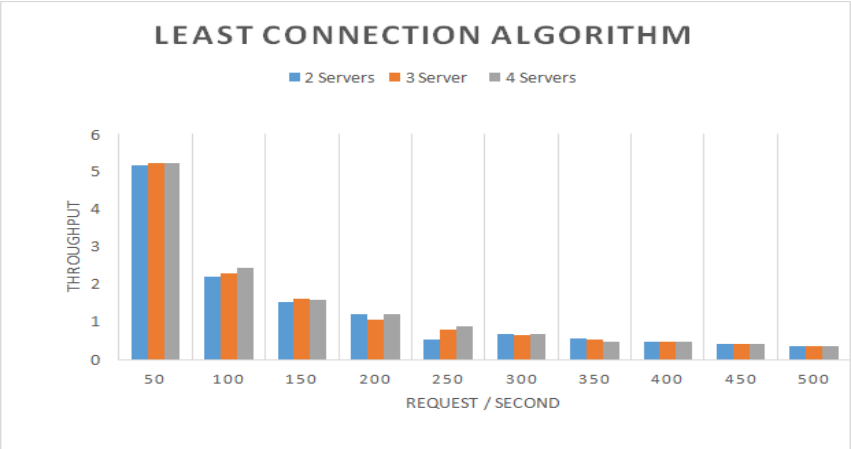

Figure 14. Throughput Graph of the Least Connection Algorithm

Figure 14 shows the amount of throughput using the Least connection algorithm. Changesin the amount of throughput produced are not very significant. Same is the case with the graphicchanges 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.

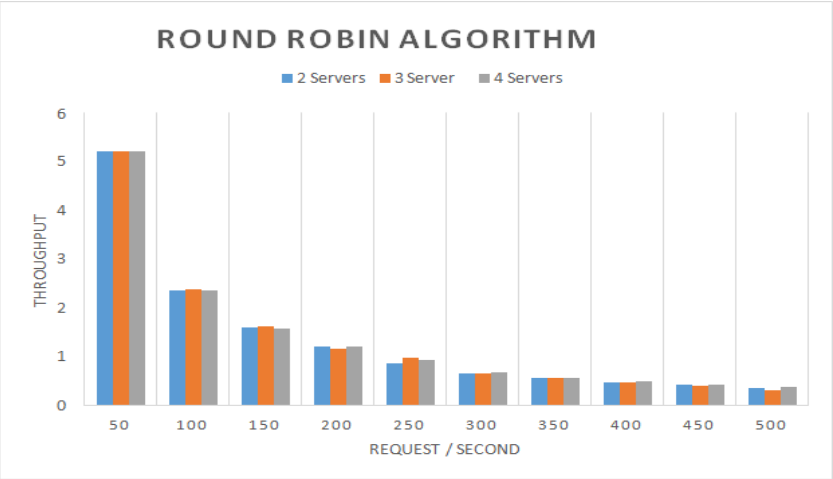

Figure 15. Throughput Graph of the Round Robin Algorithm

Figure 15 shows the throughput of the study using the Round robin algorithm. The resultingchange in the amount of throughput is not very significant. This can be seen from the throughput comparison graph in Figure 15, 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.

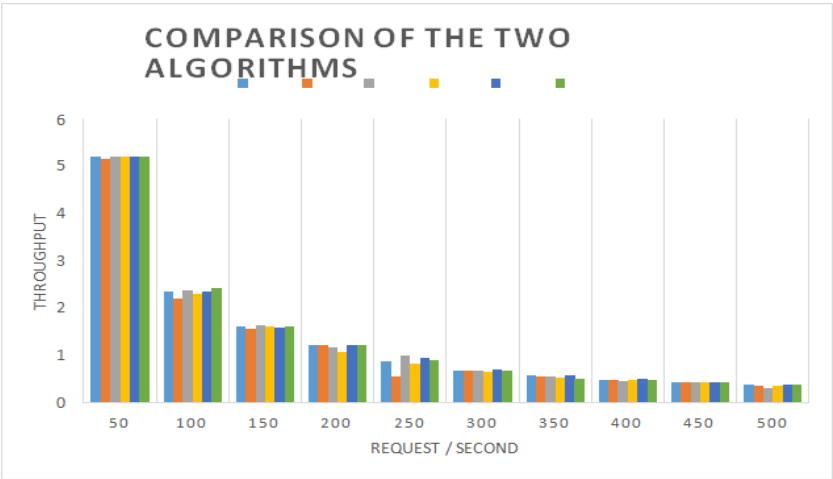

Figure 16. Two Algorithm Response Time Graph

Figure 16 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. In addition, the appearance of errors also affects the response time to be faster due to the reduced throughput value that the web server can respond to.

The Round robin algorithm has an average error that is less than the algorithm that has been implemented previously, namely the Least connection algorithm. The error value

on the Least connection algorithm with 2servers, 3 servers and 4 servers shows that the comparison of the average value is not too significant with theRound Robin algorithm.

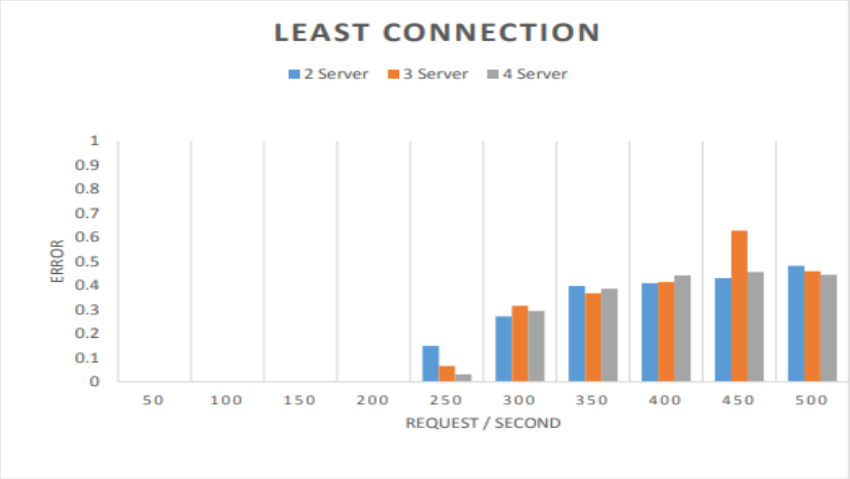

Figure 17. Least Connection Algorithm Error Graph

Figure 17 shows the number of errors in the Least connection algorithm. Just like the Roundrobin algorithm on requests for 50, 100, 150 and 200 requests / second, there were no errors. And similar to the Round robin algorithm in the Least connection algorithm, errors began to appear on 250 requests / second. However, in this experiment the Round robin algorithm showsthe average error value is smaller than the Least Connection algorithm.

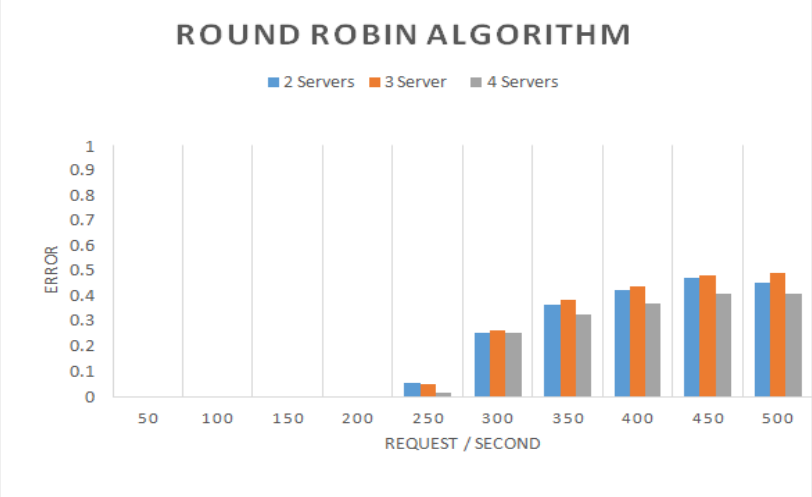

Figure 17. Graph of Round Robin Algorithm Error

Figure 17 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.

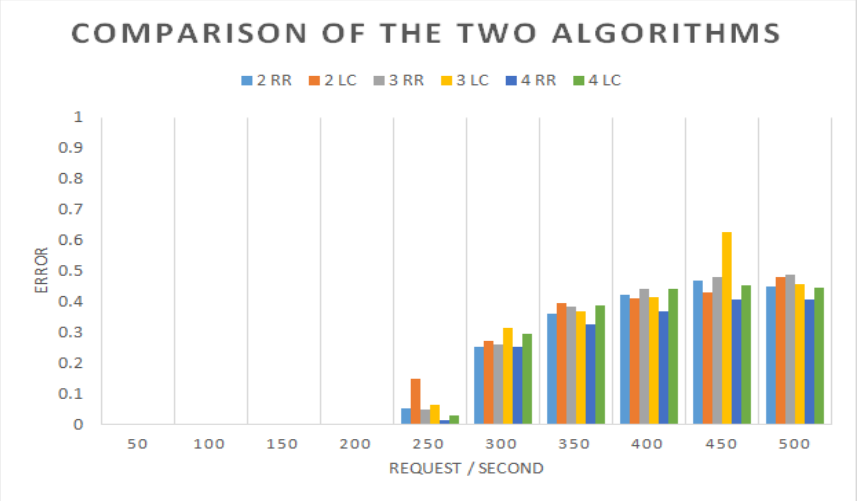

Figure 18. Error Graphs of Two Algorithms

Figure 18 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). Unlike the Round robin algorithm, in implementation 4 servers with 250 requests / second to 500 requests / second show the number of errors that are below other chart movements. So it can be concluded that using the Round robin algorithm is considered the best in reducing the number of errors.

CONCLUSION AND FUTURE SCOPE

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 fromthe 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 robinalgorithm, whereas the previous algorithm did not increase significantly.

Future Work

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 Apacheas 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.
5. SpringerPlus 5, 1991 (2016). https://doi.org/10.1186/s40064-016-3669-0
6. 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
7. Garg, Atul. (2014). A comparative study of static and dynamic Load Balancing Algorithms. IJARCSMS. Volume 2. Page 386-392.
8. 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.
9. 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.
10. 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.
11. Qi, Lianyong & Khosravi, Mohammad & Xu, Xiaolong & Zhang, Yiwen & Menon, Varun. (2021). Cloud Computing. 10.1007/978-3-030-69992-5.
12. Daigrepont, Jeffery & Morrow, James. (2020). Cloud Computing. 10.4324/9780429356674-2.
13. Beigrezaei, Mahsa. (2020). cloud computing.
14. Ali, Omar. (2020). Cloud Computing.
15. Beri, Rydhm & Singh, Jaspreet. (2020). Cloud computing.
16. P, Krishna Sankar & N P, Shangaranarayanee & Saravanan, K.. (2020). CLOUD COMPUTING.
17. Manvi, Sunilkumar & Shyam, Gopal. (2021). Cloud Computing: Concepts and Technologies. 10.1201/9781003093671.
18. Mammadova, Narmin & Akkuş, Banu & Ppp, Ppp. (2020). GRID VS CLUSTER COMPUTING. 10.13140/RG.2.2.29134.08009.
19. Buyaa(Ed, R.. (1999). High Performance Cluster Computing: Architectures and Systems.
20. Pinho, Eduardo & de Carvalho-Junior, Francisco. (2010). A Language for Object-OrientedParallel Programming Targeted at Cluster Computing Architectures.
21. Gupta, Sohan. (2020). Computer Architecture and organizational.
22. Dinquel, Jerry. (2021). NETWORK ARCHITECTURES FOR CLUSTER COMPUTING.
23. Kaushik, Shweta & Gandhi, Charu. (2021). Fog vs. Cloud Computing Architecture.10.4018/978-1-7998-5339-8.ch020.
24. Ağalarov, Mehran & Elizade, Cebrayil & Ppp, Ppp. (2020). Cluster Computing vs.

CloudComputing.
25. Rodionov, Alexey. (2019). On Evaluating a Network Throughput. 10.1007/978-3-030-19063-7_3.
26. Weinberg, Frankie. (2018). A Process Model of Network Throughput. Academy ofManagement Proceedings. 2018. 12877. 10.5465/AMBPP.2018.12877abstract.
27. Persico, Valerio & Marchetta, Pietro & Botta, Alessio & Pescapè, Antonio. (2015). On Network Throughput Variability in Microsoft Azure Cloud. 1-6. 10.1109/GLOCOM.2015.7416997.
28. Jechlitschek, Christoph. (2021). A Survey Paper on Processor Workloads.
29. Adessa, Frank. (2013). System and method for processor workload metering. [29] Martin, Philippe. (2021). The Workloads. 10.1007/978-1-4842-6494-2_5.
30. Kaushik, Shweta & Gandhi, Charu. (2021). Cloud Computing Technologies. 10.4018/978-1-7998-2764-1.ch011.
31. Labhade, Chetan. (2018). Cloud Computing Technologies: an Overview. Journal of Advances and Scholarly Researches in Allied Education. 15. 43-48. 10.29070/15/57787.
32. 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.
33. Ashraf, Adnan & Hartikainen, Mikko & Hassan, Usman & Heljanko, Keijo & Lilius, Johan & Mikkonen, Tommi & Porres, Ivan & Syeed, Mahbubul & Tarkoma, Sasu. (2013). Introductionto Cloud Computing Technologies. 10.13140/2.1.1747.8082.
34. Ma, Huan & Shen, Gaofeng & Chen, Ming & Zhang, Jianwei. (2015). Technologies based on Cloud Computing Technology. 1-5. 10.14257/astl.2015.82.01.
35. ZHOU, Xue-Song & MI, Jia-Wei & MA, You-Jie & GAO, Zhi-Qiang. (2017). Cloud Computing Technology in Smart Grid. DEStech Transactions on Engineering and TechnologyResearch. 10.12783/dtetr/icmeca2017/11958.
36. AL-Harthy, Khoula & Shimaa, & Khadjiab, & a, Duaa. (2014). Cloud Computing Technology:SWOT Analysis.
37. Srujana, R. & Dr.Y, Mohana & Mohan, M.. (2019). Sorted Round Robin Algorithm. 968-971. 10.1109/ICOEI.2019.8862609.
38. A Stephen & Shanthan, Hubert & Ravindran, Daks. (2018). Enhanced Round Robin Algorithm for Cloud Computing. International Journal of Scientific Research & Management Studies. 7.
39. 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.
40. 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.
41. Zhu, Liangshuai & Cui, Jianming & Xiong, Gaofeng. (2018). Improved dynamic load balancing algorithm based on Least-Connection Scheduling. 1858-1862. 10.1109/ITOEC.2018.8740642.