

# Experimental Model for Load Balancing in Cloud Computing Using Equally Spread Current Execution Load Algorithm

Ivan Noviandrie Falisha<sup>1</sup>, Tito Waluyo Purboyo<sup>2</sup> and Roswan Latuconsina<sup>3</sup>

Research Scholar<sup>1</sup>, Lecturer<sup>2</sup>

<sup>1,2,3</sup> Computer Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia.

ORCID: <sup>1</sup>0000-0002-9595-6827, <sup>2</sup>0000-0001-9817-3185, <sup>3</sup>0000-0002-8753-6187

## Abstract

Cloud Computing is the new generation networking paradigm that appears to provide various services over the internet. Load balancing is a key aspect of cloud computing and avoids situations where some nodes become overloaded while others are idle or busy to do. There are algorithms are advisable to provide a good mechanism and more efficient algorithm to assign multiple client requests in existing cloud nodes and for results from this experiment conducted to determine the level of effectiveness of load balancing algorithm that exists in cloud computing. in this research, we will know the output based on the overall response time, request servicing, data center loading and cost details in virtual machines. whether the algorithm used can work well to equalize the burden on cloud computing.

**Keywords:** Cloud Computing, Load Balancing, Virtual Machine.

## INTRODUCTION

The Cloud Computing system provides a relatively easiest way to store and retrieve data and files. Most importantly, if you want to create a large dataset and a large file size is available to prepare the number of users scattered throughout the world [3]. The number of user requests that access the server periodically, will impact the server has a large enough payload. So we need a solution to the problem of excessive load on the server so that it can be distributed evenly across multiple servers [1].

Distributed technology in cloud computing systems has so many benefits in terms of giving users needs and applications. the most important thing that clouds computing is to share in terms of software resources, software resources, sharing of information resources through the Internet network, reducing cost-effectiveness, to provide quality of service (QoS) metrics. Including response time, cost, throughput, performance and resource utilization [5].

Therefore, there are several approaches to load balancer methods in terms of cloud computing. In this experiment, we will discuss load balancing algorithms that are often used in cloud computing. The algorithm used in load balancing will determine the result of throughput, response time and cost. After that, we will experiment using one load balancer technique on a virtual cloud computing machine and will use

simulations using a simulator. CloudAnalyst is one of the simulator tools developed at the University of Melbourne whose aim is to support and simplify the evaluation of social networking tools based on the geographic distribution of users and data centers. In this tool, user communities and data centers that support social networks are characterized and, by location [4].

## METHODS

They are many methods are often used to equalize the queue load on available cloud computing. some of which are presented in this paper, which are:

### A. Round Robin

The Round Robin algorithm is commonly used algorithms in cloud computing for load balancer and is the simplest and easiest to implement. as for several variants of the available round robin algorithm focusing on the measurement of different parameters, and here is the terminology that is familiar with this algorithm [2]:

- Burst Time: BT is the required completion time of the request.
- Time Quantum: TQ is time calculated on request that is allowed to access the VM.

### B. Throttled

This algorithm uses the system available in the VM status list (BUSY / AVAILABLE.) at the time of the request from the user comes on the load balancer. its checks to find the available VM, after listing the available VM and then allocates to the VM that is available. so these algorithms do not have much of a high degree of complexity in their implementation and their performance is significant [2].

### C. ESCE (Equally Spread Current Execution Load)

This algorithm works continuously based on the queue and passes it to a different virtual machine randomly and distributes it by checking the size of the load coming, then distributes it to a virtual machine that has light work. [4]. In this experimental paper, the method to be used is from one of

the above algorithms (ESCE) which will be used in the CloudAnalyst simulator.

### EXPERIMENT SETUP

In this research as for things related to the implementation of an experiment, is software and any hardware used by researchers. For our software using eclipse OXYGEN.1 with Java programming language, and for hardware used in the form of Intel® Core™ i3 Processor with 4GB RAM. We assume the use of the large-scale internet that is on the application, for example, Facebook, Instagram, Twitter, etc. in this case, we assume the use of more than 300 million users worldwide who access the application and only executed in 1 Data Center.

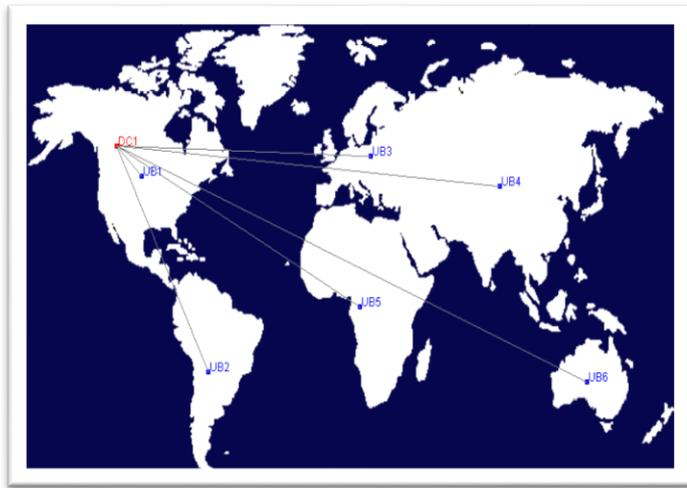


Figure 1: Scenario Cloud Computing [6].

Figure 1 shows the Platform execution results in CloudAnalyst in the form of a GUI with a map of the world map adjusted to the geographic scale between continents in cloud computing that actually happens with the Java programming language in eclipse OXYGEN. 1 And shows the

network scenarios to be performed based on 1 data center used in this experiment and requests from many User Bases. the algorithm to be used in this experiment is Equally Spread Current Execution Load (ESCE) on the load balancer.

Table 1: Assumptions of the scale of application usage.

Region	CloudAnalyst Region Id	Users
Africa	4	20
Oceania	5	30
Asia	3	80
North America	0	70
South America	1	50
Europe	2	50

Table 1 shows the assumptions of multiple users based on the user's geographical location and above this initials should be inputted to the cloud computing simulator is CloudAnalyst in order to run this simulation. why only some of the Region is displayed because it is already representative of each continent. for the Region of Id here to explain to the region code program in the Region, for the number of users only assumes use based on actual events.

#### A. Simulation Configuration

In this scenario, we assume that at the time users is accessing for 2 hours. users use the app at night and it is also assumed that every 5 minutes the user loads a new request. this is related to the number of user requests from many regions and continuously it will affect the Data Center in cloud computing system. for the configuration to be used in the software will be explained in many tables that represent the various parameters that must be used for the simulation to run properly and correctly. The parameters used are based on User Base (UB), Cost, Specification of Data Center, and parameters contained in the Internet network.

Table 2: Parameter on the user base.

User Base	Region	Time Zone	Peak Hours (GMT)	Peak Hours (Local Time)	Online Users Simultaneously During Peak Hour	Online Users Simultaneously During Off-Peak Hours
UB1	0	GMT - 6.00	14:00-16:00	8.00-10.00 pm	500,000	50,000
UB2	1	GMT - 4.00	16:00-18:00	8.00-10.00 pm	200,000	20,000
UB3	2	GMT + 1.00	21:00-23:00	8.00-10.00 pm	400,000	40,000
UB4	3	GMT + 6.00	02:00-04:00	8.00-10.00 pm	250,000	25,000
UB5	4	GMT + 2.00	22:00-24:00	8.00-10.00 pm	150,000	15,000
UB6	5	GMT + 10.00	10:00-12:00	8.00-10.00 pm	180,000	18,000

Table 2 shows the data on each base derived from which users and when active users access the cloud server simultaneously of each User Base (UB) have different access times from each other and based on the local time of each region in Greenwich Mean Time (GMT). For other parameters, we also used online users simultaneously during peak hour offline and online users simultaneously during peak hour online.

**Table 3:** Assume of Cost.

<b>Cost per 1 Gb of Data Transfer (from/to Internet)</b>	\$0.11
<b>Cost VM per hour (1024Mb and 100MIPS)</b>	\$0.11
<b>Memory Cost</b>	\$0.05
<b>Storage Cost</b>	\$0.1

In the world of cloud computing as for things that should take precedence in terms of the specifications that belong to the virtual machine on the cloud is the cost that must be met from the organizers of the cloud itself. It can affect the performance that will be generated by the virtual machine and the level of satisfaction of the consumer. Table III. shows In this experiment, we assume an approach to the costs of cloud service providers such as Amazon EC2. Because in the case of this simulation we adjust to the actual price specification.

**Table 4:** Physical parameter of Data Center.

<b>Parameter of Data Center</b>	<b>The Value Used in The Simulation</b>
<b>VM Image Size</b>	100,000
<b>VM Bandwith</b>	1,000
<b>VM Memory</b>	1024
<b>Architecture of Data Center</b>	X86
<b>OS of Data Center</b>	Linux
<b>VMM of Data Center</b>	Xen
<b>Number of Machines of Data Center</b>	20
<b>Memory per Machine of Data Center</b>	4096
<b>Storage per Machine of Data Center</b>	100,000
<b>Available per BW of Data Center</b>	10,000
<b>Number of Processors per Machine of Data Center</b>	4
<b>Processors Speed of Data Center</b>	100
<b>VM Policy of Data Center</b>	1,000
<b>Grouping Factors on Request</b>	100
<b>Executable Instruction Length</b>	250

Table 4 shows the assumptions on the level of specification of the virtual machine and the physical machine to be used in this experiment and will be inputted to the cloud computing simulator engine. this can greatly affect the experiment this

time because these parameters are located in the Data Center, the better the virtual machine specifications contained in the Data Center the better the performance of the cloud itself but will certainly be more costly.

For the value used in the simulation, of course, we try to adjust to the circumstances actually, because then we can see a comparison that is not so far from the results that actually exist. Of course, this simulation aims to be able to handle user requests in huge number. Data Center with good quality specifications that can support it.

**Table 5:** Assumption value of matrix latency ( milliseconds ).

<b>Region/Region</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>0</b>	35	110	160	260	260	110
<b>1</b>	110	35	260	510	360	210
<b>2</b>	160	260	35	160	160	210
<b>3</b>	260	510	160	35	510	510
<b>4</b>	260	360	160	510	35	510
<b>5</b>	110	210	210	510	510	35

Table 5 shows 1 of 2 categories in the matrix parameters of the existing Internet network is the latency of the internet. and will influence the simulation because it is adjusted to the actual state.

**Table 6:** Assumption value of matrix latency ( Mbps ).

<b>Region/Region</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>0</b>	3000	1500	1500	1000	1000	1500
<b>1</b>	1500	700	1000	1000	1500	1000
<b>2</b>	1500	1500	2500	1000	1500	1000
<b>3</b>	1000	1000	1500	2000	1000	1500
<b>4</b>	1500	1000	1000	1500	600	1000
<b>5</b>	1000	1500	1000	1000	1500	2500

Table 6 shows 1 of 2 categories of matrix parameters on the existing Internet network is the bandwidth parameter. And in this case, will affect the simulation because it is adapted to the actual situation. With geographical conditions, there will be a lot of delays in the process of sending user requests to the Data Center.

## RESULTS

From the assumptions we have made before, what have expected results from the ESCE (Equally Spread Execution Load) algorithm on load balancing in cloud computing on the simulator. How the algorithm used in this simulation works to

equalize the load request from the user whether the result is good enough or indeed there is overload in the simulation. Of course, the results obtained in many tables below, what parameters are measured in the simulation this time is the overall response time summary, response time by region, Data Center servicing time, and overall cost of the virtual machine.

**Table 7:** Overall response time summary.

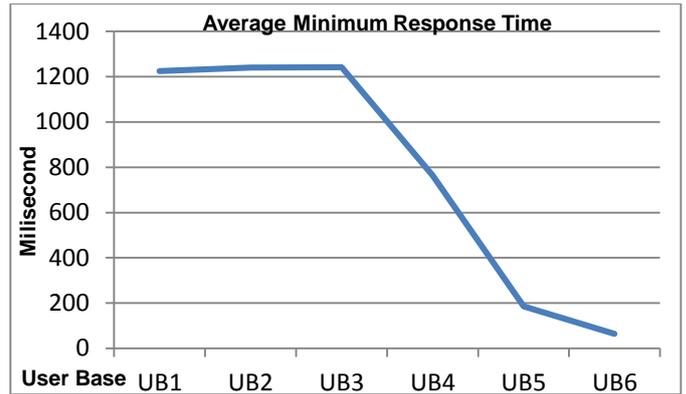
	Avg (ms)	Min (ms)	Max (ms)
<b>Overall response time</b>	963452.59	6513.63	1868402.51
<b>Data Center processing time</b>	960399.45	6290.43	1868321.56

Table 7 shows a summary of the average response time and data processing time of the simulated output used in this experiment. So, it can be seen from the simulation level efficiency of the algorithm that we use in load balancer to equalize incoming requests from users to the Data Center [6] [7] [8] [9].

**Table 8:** Response time based on region.

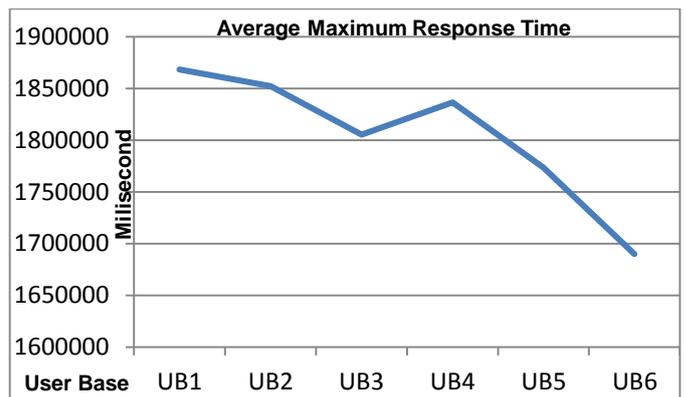
Userbase	Avg (ms)	Min (ms)	Max (ms)
<b>UB1</b>	982262.26	122442.12	1868402.51
<b>UB2</b>	969778.58	123975.79	1852376.29
<b>UB3</b>	962658.52	124222.40	1805495.26
<b>UB4</b>	930623.97	76397.68	1836549.64
<b>UB5</b>	887299.41	18594.52	1773514.95
<b>UB6</b>	485736.14	6513.63	1689898.69

Table 8 shows a summary of the average response time based on each region of the simulated output used in this experiment. We can also see the simulation results of the efficiency level of the algorithm we use in load balancer to equalize incoming requests from users to the Data Center. There are several parameters that are generated by the load balancer where the response time process occurs at minimum response time, maximum response time, and average response time. Can be seen also in some tables that have been provided below, in the graphical representation very clearly how the process performed by load balancer in the Data Center.



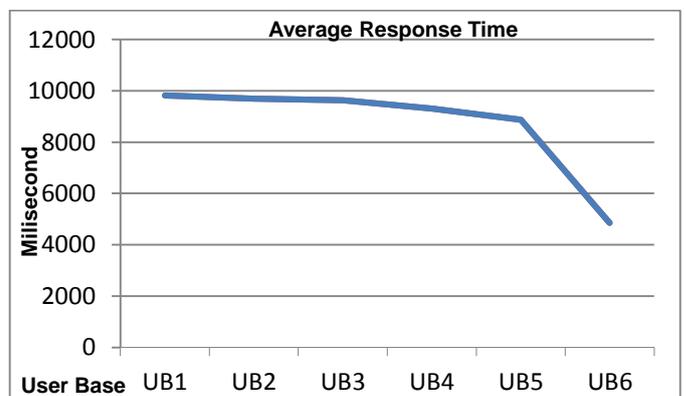
**Figure 2:** Average response time minimum chart by region.

Figure 2 shows the average result of minimum response time by region, it can be said that the minimum time taken by the load balancing algorithm used before reaching the load stage equally in the Data Center. From the graph above can be seen also in each User Base distance based on geographical location affect the response time level.



**Figure 3:** Average response time maximum chart by region.

In this figure 3. shows the average result of the maximum response time by region, it is said that the maximum time taken by the load balancing algorithm is used before it reaches the load stage equally in the data center.



**Figure 4:** Average response time chart by region.

Figure 4 shows the average result of the overall response time by region, it can be said that the maximum time taken by the load balancing algorithm is used before reaching the same load stage in the data center.

**Table 9:** Data center request servicing times.

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	960399.45	6290.43	1868321.56

In table 9 can be seen how the data center to overcome request from users based on the average service time. Also seen minimum service time span until maximum service time is calculated average. From this result can be known very clearly the quality of a Data Center service to all incoming user requests.

**Table 10:** The Overall cost of the virtual machine (\$).

Data Center	VM Cost \$	Data Transfer Cost \$	Total \$
DC1	0.88	41.05	41.93

Table 10 shows the total cost incurred by 1 Data Center, related to the total amount of costs that is based on virtual machine cost & data transfer cost [10].

## CONCLUSION

In this research, it can be seen that load balancing is a very important aspect of the performance of existing cloud computing. Because the demand load from users is very much and varied. From this research, it can be concluded that the ESCE algorithm used in load balancer in cloud computing simulations using CloudAnalyst is the average response rate that is still in the average range between UB1 and the others. However, in terms of average minimum response and maximum response is too long. And for future work, it is known that load balancing greatly affects cloud computing. In this study only use one data center, but there is still another approach that can be used in cloud circles with distributed Data Center.

## REFERENCES

- [1] Cahyadi, R. (2016). Load Balancing Performance Analysis On SDN Using Partitioning and Transitioning Algorithm.
- [2] Maheta, S. D. (2014). Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Environment. *International Journal of Computer Application* (0975-8887), 94.
- [3] Klaithem Al Nuaimi, N. M.-J. (2012). A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms.
- [4] Harmandeep Singh Brar, V. T. (n.d.) (2009). A Survey of Load Balancing Algorithms in Cloud Computing. *International Journal of Computer Science Trends and Technology (IJCTST)*, 2 (3).
- [5] Younis, H. J. (2015). Efficient Load Balancing Algorithm in Cloud Computing.
- [6] Wickremasinghe, B. (2009). *CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environment*. Melbourne: Melbourne University.
- [7] Rajkumar Buyya, R. R. (2009). Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. *Proceedings of the 7th High Performance Computing and Simulation*. New York, USA, Leipzig, Germany: IEEE Press.
- [8] R. Buyya, and M. Murshed (2002), "GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing," *Concurrency and Computation: Practice and Experience*, vol. 14, no. 13-15, pp. 1175-1220.
- [9] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic (2009), *Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility*, *Future Generation Computer Systems*, Volume 25, Number 6, Pages: 599-616, ISSN: 0167-739X, Elsevier Science, Amsterdam, Netherlands.
- [10] Kaur, N. S. (2013). Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing . *International Journal of Engineering Science Invention*, ISSN (Online): 2319-6734.