Efficient Load Balancing for Cloud Computing by Using Content Analysis

Preecha Somwang

Faculty of Engineering and Architecture, Rajamangala University of Technology Isan, Nakhon Rachasima, Thailand

***Abstract*:** Nowadays, computer networks have grown rapidly due to the demand for information technology management and facilitation of greater functionality. The service provided based on a single machine cannot accommodate large databases. Therefore, single servers must be combined for server group services. The problem in grouping server service is that it is very hard to manage many devices which have different hardware. Cloud computing is an extensive scalable computing infrastructure that shares existing resources. It is a popular option for people and businesses for a number of reasons including cost savings and security. This paper aimed to propose an efficient technique of load balance control by using HA Proxy in cloud computing with the objective of receiving and distributing the workload to the computer server to share the processing resources. The proposed technique applied round-robin scheduling for an efficient resource management of the cloud storage systems that focused on an effective workload balancing and a dynamic replication strategy. The evaluation approach was based on the benchmark data from requests per second and failed requests. The results showed that the proposed technique could improve performance of load balancing by 1,000 request /6.31 sec in cloud computing and generate fewer false alarms

***Keywords*:** Cloud computing, Load balancing, Round-robin.

1. Introduction

Nowadays, normal server is managed by a single server computer that can achieve this easily and quickly. The problem of single server is that the number of visitors exceeds the capacity and performance is not enough to service that, especially at the time [1]. Server computer clustering to optimize the service provided is another possible solution. The development technique of a computer clustering is a virtual cluster consisting of multiple virtual computers running on cloud computing spread out within the network [2].

Cloud computing technology is gaining popularity due to the high speed of the internet which has become quite useful in our daily public life as well. Users can conveniently access various services from their computers or portable devices [3]. Currently, cloud storage systems have many service providers that most forms provide a service storage space for each user [4]. Service providers will develop applications to support services on various platforms on personal computers and on different operating systems [5]. One of the main important features of cloud storage systems is the efficient support of many users [6]. Adding storage to increase resources for user service can efficiently allocate storage resources is a method of load balancing of the storage unit [7].

Workload balancing can classify user access into two types: Centralized system and Distributed system [8]. Centralized system is easier to manage in that all systems are concentrated to a specific leader or site [9]. But the disadvantage is that all the management burden is centralized so the server that is responsible for controlling the workload balancing has to work hard in the case of many users [10]. Distributed system has an advantage in that stability of the system is higher because the workload does not depend on one server but the development technique of the system is more complicated [11, 12]. This paper aimed to propose a workload balancing algorithm suitable for cloud storage systems. The cloud storage system suitable for handling the workload caused by using the services of users more efficiently and use the available resources more efficiently. This research is an implementation of a group of server computers to support the website on Rajamangala University of Technology Isan (RMUTI)

The rest of this paper is organized as follows: Section 2 discusses background and related works of cloud computing. Section 3 describes work load balancing. Section 4 discusses the methodology. Experiment and results are shown in Section 5 and conclusion in Section 6.

2. Background and Related Work

Cloud computing is a virtualization technology that is divided into 3 types of services: Software as a Service, Platform as a Service and Infrastructure as a Service.

2.1 Software as a Service (SaaS)

SaaS is a service based on cloud computing systems for providing services in the form of packet software to recipients [13]. The software does not run on the user's computer or on the host server but on the service provider's computer system. Users access the software online through a web browser and process all the data on the service provider's machine [14].

2.2 Platform as a Service (PaaS)

PaaS is a cloud-based computing service that provides software development services in a scalable environment for software developers [15]. Software developers have easy and flexible access to the database that is similar to the traditional database management system [16].

2.3 Infrastructure as a Service (IaaS)

IaaS is the service of the service provider that provides Information Technology (IT) infrastructure in the resources of virtual computers [17]. Users access resources via the internet to install the desired software t through a virtual computer [18].

Virtual computer or Virtual Machine (VM) technology can create a new system in a short period of time, is flexible and able to increase or decrease resources' size easily and quickly [19]. Virtual computers on a separate cloud must move between computers according to the proper

environment. The software controlling virtual computer migration must decide based on the actual workload [20].

2.4 Related work

Load balancing algorithm plays a vital role in grid computing in the utilization of grid resources that are globally distributed. Grid computing technologies enable controlled resource sharing among users; for execution of the tasks in distributed communities and coordinated use of those shared resources as community members tackle common goals. Grid computing and cloud computing are conceptually similar in that they both share and deliver computing resources such as servers, storage, databases, and software [21]. However, in cloud computing, small amount of data is accessed by a large group of users, while in grid computing, large amounts of data are accessed by small groups of users. Load balancing using enhanced Ant Colony Optimization (ACO) was proposed by [22] in grid computing extended to focuses on load balancing based on resource fitness in addition to providing fault tolerance. The performance metrics used to measure the proposed algorithm, which include execution time, throughput, makespan, latency, load balancing and success rate. The results showed that the proposed method was able to maintain the success rate in addition to maintaining throughput and providing better load balancing, and lower latency. Load balancing in the context of the QoS was proposed by [23] to implement load balancing algorithm that allows obtaining of the shortest or lowest load paths for the transmission and forwarding of packets among the end devices of the network. The implementation of the load balancing algorithm allowed for improved bandwidth, decreased response times and optimal distribution of the load of the links.

3. Workload Balancing

Load balancing is a breakdown of the total number of tasks a computer must perform between two or more computers. It helps to divide the incoming work from the user into a distributed group of servers [24]. Load balancing offers workload distribution services in the hardware, software or integration format [25].

3.1 HA-Proxy load balancer

The hardware of a server computer has the resources or any physical part of a computer system. HA Proxy can build virtual servers on load balancing techniques to increase service efficiency [26]. It is a program that is created for server workload distribution system. It divides the processing workload into the server computers that are registered in the server computer cluster. It has flexibility to increase the number of server computers according to the number of users in the system, load balance to accommodate the growing number of users, and lower the number of access failures [27]. The ability to accommodate a large number of users depends on the number of server PCs available as shown in Figure 1.

This research presents grouping of computers that provide distribution of workloads for processing and sharing existing resources. The proposed technique can handle the amount of users who access the system in the case of one server computer within the group having problems thus interrupting service stop. Load balancer is a workload distribution machine that sends the workload to other server computers until the machine stops serving as normal. Service is processing of the data request from user group of a connection to a load balancer.

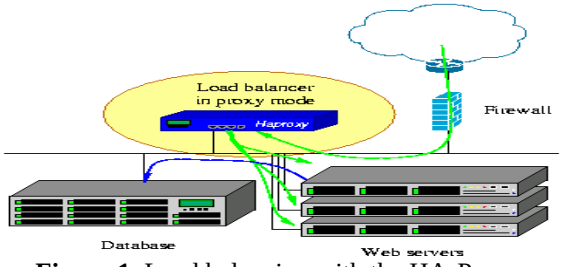


Figure 1. Load balancing with the HA-Proxy.

The service provider selects load balancing for the user, and service and the will contact each other directly. There are three types of load balancing techniques: Sticky, Round-robin and Workload [28].

3.1.1 Sticky

Sticky sends traffic based on the session from the service that the user has access to.

3.1.2 Round-Robin

Round-robin sends traffic to the server within the loop.

3.1.3 Workload

Workload is the transmission of traffic by looking at the importance of the performance of the server within the cluster, if one server has a lot of load then traffic will be sent to another.

Workload distribution is load balancing to share existing resources. The advantage is the ability to send information to the machine quickly that would otherwise take a time-consuming calculator to find a machine in a group of computers to send a lot of information [29].

3.2 Workload Distribution

Workload distribution must take into account user requests to the various editors based on the content analysis of the requests [30]. Similar content must be sent to the same processor as determined by the summary cache. This reduces the amount of information exchanged between overtime and services. It determines against the appellant which server has a lot of workload or less workload. It finds data that is stored on a computer for future use without the need to retrieve data from the source again thus saving time [31]. Caching is intended to speed up data access, not to store data. The data in the cache is likely to disappear at any time. There are 2 processes for distributing workloads in memory cache: Cache hit and Cache summary. Cache hit is able to read data from the cache when the data is not lost from the cache. In addition, cache miss is not in the cache, which makes it necessary to read data from the source which more time-consuming [32]. Cache summary information is exchanged between servers to schedule that server has the requested matching content for duplication to reduce workload, as shown in Figure 2.

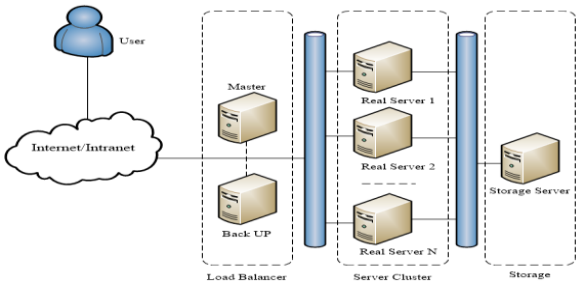


Figure 2. The workload distribution.

Workload distribution is a special criteria that balances the load based on the content in which requests between the same content must be sent to the same server. It is an immediate response to the client from the data in the buffer that all servers need to exchange workloads together as needed. It finds common information between proxies to store data and exchange them in the summary cache format to make the data smaller. It is a reduction process of the amount of observations in exchanging information between the proxy and the status of the workload that indicates whether the workload is high or low. It is a choice function to distribute workload to the proper service [33].

4. The Methodology

HA Proxy is a feature that can distribute workloads in data processing for a computer server group. The structure and functionality were divided into 3 parts: Load Balancer, Web Server, and Database server as shown in Figure 3.

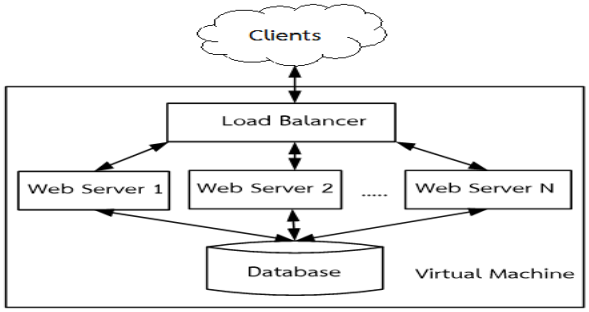


Figure 3. Load balance of propose method.

Website virtualization services on the workload distribution system is a process that provides site space services to departments within the RMUTI University management. Creating a website or virtual host from the reference by the domain name or URL (Name-based), which is a method of using different domain names provided on the same server, as shown in Figure 4.

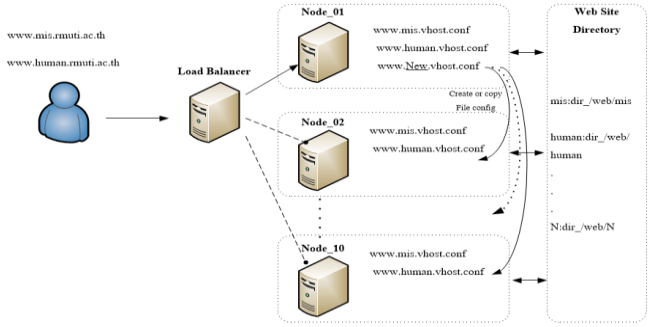


Figure 4. Website virtualization service.

The proposed method is a starting point to checking the status of website services using a PHP program monitoring tool for checking a website's status: HTTP 200 value is normal and HTTP 404 value is abnormal [34]. In addition, the best destination computer was selected by the terminal which has enough memory and is better than the current environment, defined as Equation 1.

$$(mS + mR) < (mD - mR) \quad (1)$$

Given *mS*= memory Source, *mR*= memory Requirement and *mD*= memory Destination [35]. The workload is due to select the destination machine that takes on paid employment rate of the Central Processor Unit (CPU) not exceeding the specified limit in Equation 2.

$$cpu\_average = \frac{\sum_{i=1}^{n} cpu_{usage_i}}{n} + \min \quad (2)$$

It made sure the destination computer has processor load times better than the machine currently being used that is floating point per second, defined as Equation 3.

$$load_d = load_{vm} + \left[load_{rd} x \left(\frac{\cos t_c}{\cos t_d} \right) \right] \quad (3)$$

Given $load_d$ = CPU usage on the destination machine for selected optimal node to assign weights to each node to be used respectively in Equation 4.

$$w = \frac{cpu_{load} + memory_{load}}{200} \quad (4)$$

The weight assignment to the appropriate node is calculated from the sum of the CPU load and the memory load, which is the rate of main memory usage [36]. The amount of CPU usage on each computer is the main choice function for deciding on a virtual computer to move to work load on the optimal computer, as shown in Figure 5.

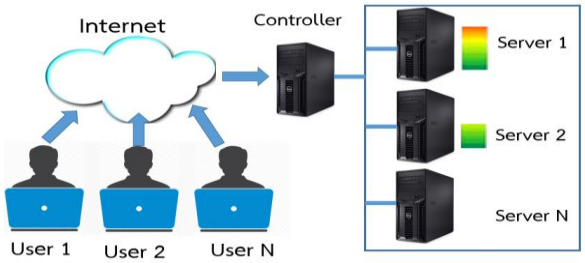


Figure 5. CPU and memory load control.

Network performance and throughput are crucial aspects that need attention from network professionals. Although the overall throughput measures speed, the bandwidth is only indirectly related to speed. The evaluated approach on the throughput refers to how much data can be transferred from one location to another in a given amount of time, and can be calculated as in the Equation 5.

$$Throughput = \frac{Request}{ExecutionTime} \quad (5)$$

The balance sub-module is the controller of the monitoring module and it selects the destination node and moves the workload to a node suitable for the amount of computer needs, as shown in Figure 6.

```
balance
        call initstate;
        call buildinfo;
        loop
                set state= monitoring();
                if state= normal then
                        call findbestnode;
                        call migrate;
                end if
        endloop
end
```

Figure 6. The pseudo code of balance

5. Experiment and Results

The load balancing system designed runs on a Linux system whose operating environment is Debian server 8.10 software [37]. The storage system type of the backup is Intel(R) Xeon(R) 16 Core, 2.4 GHz, 32 GB of RAM, SAS 1 TB of

Hard Disk. This load balancing aimed to improve the efficiency of the Rajamangala University of Technology Isan web server environment and adapt to changing systems. The parameters considered in the evaluation phase were 1,000 to 5,000 requests. The effect of these parameters were evaluated based on the number of requests per second, and failed requests. This was a test to measure the computing performance between traditional computer use and through VM on a separate cloud computing system in which there were 3 computers with hardware capabilities at 3 different levels. The decision making of this automatic balancing system lies in the resource usage of the central processing unit and the main memory usage, as shown in Figure 7.

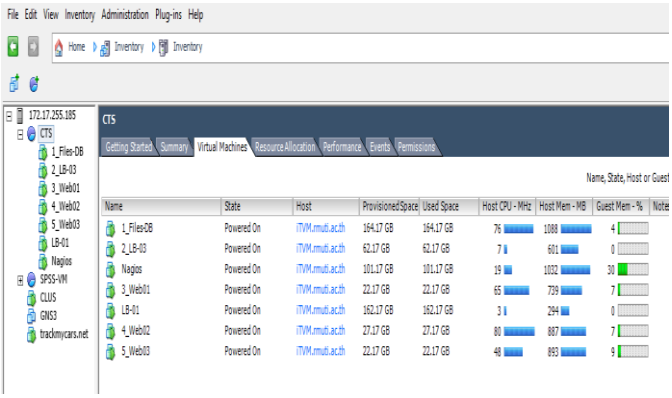
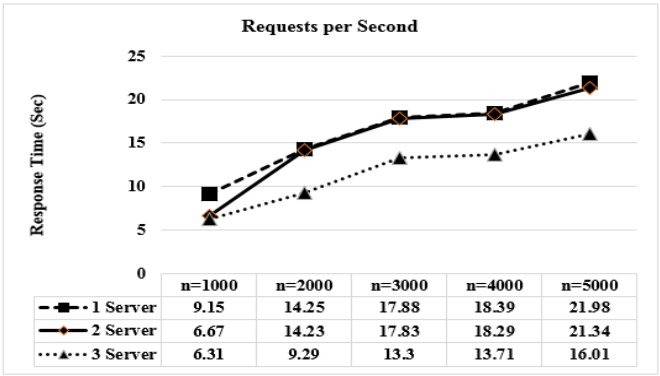


Figure 7. Workload management

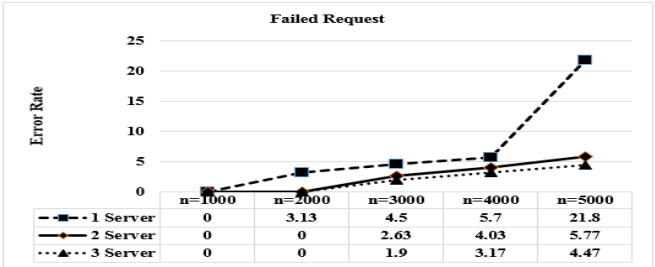
The initial process was the monitoring module sending data from source to the balancing module for the decision to transfer the workload to the computer that had the most capabilities. The test results showed that the proposed single server computer took the most time for user request to the response time between 9.15 - 21.98 seconds, as shown in Figure 8.



	n=1000	n=2000	n=3000	n=4000	n=5000
1 Server	9.15	14.25	17.88	18.39	21.98
2 Server	6.67	14.23	17.83	18.29	21.34
3 Server	6.31	9.29	13.3	13.71	16.01

Figure 8. Request per second

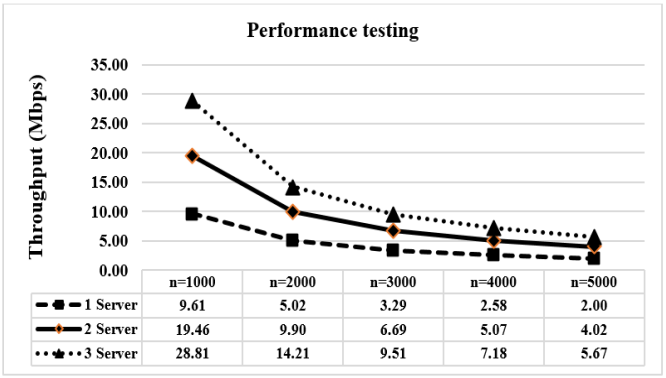
The actual error rate found a single server computer had the highest error rate between 3.13 - 21.80 seconds, meaning that errors started being found between 3,000 - 5,000 requests, as shown in Figure 9.



	n=1000	n=2000	n=3000	n=4000	n=5000
1 Server	0	3.13	4.5	5.7	21.8
2 Server	0	0	2.63	4.03	5.77
3 Server	0	0	1.9	3.17	4.47

Figure 9. The actual error rate

Throughput values are the main concept in testing computer network performance that capacity measured in bits per second.



	n=1000	n=2000	n=3000	n=4000	n=5000
1 Server	9.61	5.02	3.29	2.58	2.00
2 Server	19.46	9.90	6.69	5.07	4.02
3 Server	28.81	14.21	9.51	7.18	5.67

Figure 10. Results of throughput

Switching bandwidth is the actual throughput. Forwarding rate is the amount of packets the device can switch and also packets per second (pps). As a result; workload consolidation through the system performance tests showed that a large number of server increased throughput.

6. Conclusions

The supply distribution of workloads in cloud data was processed by analyzing content of client requests, where the request data were stored in the buffer of each server. The data processing is buffer summary to exchange between servers in order distribute the workload to the service server within the server group. The experiment demonstrated that efficiency of the workload distribution can improve performance of response time for request per second from the client. The workload management is a technique to increase the number of servers serving for faster data processing and reduced errors. The increased number of servers in each service did not interrupt the processing and service.

7. Acknowledgement

This research has been supported by the Office of Academic Resources and Information Technology, Rajamangala University of Technology Isan, Thailand.

References

[1] J. Park, J. Kim, C. Ahn, Y. Woo, H. Choi, "Cluster Management in a Virtualized Server Environment," 2008 10th International Conference on Advanced Communication Technology, Gangwon-Do, pp. 2211-2214, 2008.

[2] H. Halabian, I. Lambadaris, Y. Viniotis, "Optimal server assignment in multi-server queueing systems with random connectivities," Journal of Communications and Networks, vol. 21, no. 4, pp. 405-415, 2019.

[3] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, Y. Xiang, "Block Design-Based Key Agreement for Group Data Sharing in Cloud Computing," IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 6, pp. 996-1010, 2019.

[4] S. S. Tirumala, Abdolhossein Sarrafzadeh, Paul Pang, "A survey on internet usage and cyber security awareness in students," 14th Annual Conference on Privacy, Security and Trust, Auckland, New Zealand, pp. 223 – 228, 2016.

[5] B. Hong, W. Choi, "Optimal Storage Allocation for Wireless Cloud Caching Systems With a Limited Sum Storage Capacity," IEEE Transactions on Wireless Communications, vol. 15, no. 9, pp. 6010-6021, 2016.

[6] J. Cha, S. Kim, "Analysis of I/O Performance for Optimizing Software Defined Storage in Cloud Integration," 2018 IEEE 3rd International Conference on Communication and Information Systems (ICCIS), Singapore, Singapore, pp. 222-226, 2018.

[7] Y. Zhang, Q. Wei, C. Chen, M. Xue, X. Yuan, C. Wang, "Dynamic Scheduling with Service Curve for QoS Guarantee of Large-Scale Cloud Storage," in IEEE Transactions on Computers, vol. 67, no. 4, pp. 457-468, 2018.

[8] M. U. K. Khan, M. Shafique, J. Henkel, "Power-Efficient Workload Balancing for Video Applications," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 6, pp. 2089-2102, 2016.

[9] X. Jin, Z. Liu, Q. Li, Q. Dai, "Depth Assisted Adaptive Workload Balancing for Parallel View Synthesis," IEEE Transactions on Multimedia, vol. 20, no. 11, pp. 2891-2904, 2018.

[10] N. Zarin, A. Agarwal, "A Centralized Approach for Load Balancing in Heterogeneous Wireless Access Network," 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), Quebec City, QC, pp. 1-5, 2018.

[11] Z. Zeng, B. Veeravalli, "On the Design of Distributed Object Placement and Load Balancing Strategies in Large-Scale Networked Multimedia Storage Systems," IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 3, pp. 369-382, 2008.

[12] Y. Jiang, "A Survey of Task Allocation and Load Balancing in Distributed Systems," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 2, pp. 585-599, 2016.

[13] C. Dadi, P. Yi, Z. Fei, H. Lu, "A New Block-Based Data Distribution Mechanism in Cloud Computing," IEEE 3rd International Conference on Cyber Security and Cloud Computing, Beijing, pp. 54 – 59, 2016.

[14] J. Park, D. Yun, U. Kim, K. Yeom, "Approach for Cloud Recommendation and Integration to Construct User-Centric Hybrid Cloud," IEEE International Conference on Smart Cloud, New York, NY, pp. 24-32, 2017.

[15] A. Vig, R. S. Kushwah, S. S. Kushwah, "An Efficient Distributed Approach for Load Balancing in Cloud Computing," International Conference on Computational Intelligence and Communication Networks, Jabalpur, pp. 751 – 755, 2015.

[16] A. B. S., H. M.J., J. P. Martin, S. Cherian, Y. Sastri, "System Performance evaluation of Para virtualization, Container virtualization and Full virtualization using Xen, OpenVZ and XenServer," Fourth International Conference on Advances in Computing and Communications, Cochin, pp. 247 – 250, 2014.

[17] M. Zhang, R. Ranjan, M. Menzel, S. Nepal, P. Strazdins, W. Jie, L. Wang, "An Infrastructure Service Recommendation System for Cloud Applications with Real-time QoS Requirement Constraints," IEEE Systems Journal, vol. 11, no. 4, pp. 2960-2970, 2017.

[18] F. A. Samimi, P. K. Mckinley, S. M. Sadjadi, C. Tang, J. K. Shapiro, Z. Zhou, "Service Clouds: Distributed Infrastructure for Adaptive Communication Services," IEEE Transactions on Network and Service Management, vol. 4, no. 2, pp. 84-95, 2007.

[19] N. T. Hieu, M. D. Francesco, A. Ylä-Jääski, "Virtual Machine Consolidation with Multiple Usage Prediction for Energy-Efficient Cloud Data Centers," IEEE Transactions on Services Computing, vol. 13, no. 1, pp. 186-199, 2020.

[20] G. Sun, D. Liao, D. Zhao, Z. Xu, H. Yu, "Live Migration for Multiple Correlated Virtual Machines in Cloud-Based Data Centers," IEEE Transactions on Services Computing, vol. 11, no. 2, pp. 279-291, 2018.

[21] S. Khan, B. Nazir, I. A. Khan, S. Shamshirband, A. T. Chronopoulos, "Load balancing in grid computing: Taxonomy, trends and opportunities," Journal of Network and Computer Applications, Volume 88, Pages 99-111, 2017.

[22] S. Bukhari, K. R. Ku-Mahamud, H. Morino, "Load Balancing Using Dynamic Ant Colony System Based Fault Tolerance in Grid Computing," International Journal of Communication Networks and Information Security (IJCNIS), Vol. 11, No. 2, pp. 297-303, 2019.

[23] J. Porras, D. Ducuara, G. Puerto, "OpenDaylight vs. Floodlight: Comparative Analysis of a Load Balancing Algorithm for Software Defined Networking," Journal of International Journal of Communication Networks and Information Security (IJCNIS), Vol. 10, No. 2, pp.348- 357, 2018.

[24] H. Son, S. Lee, S. Kim, Y. Shin, "Soft Load Balancing Over Heterogeneous Wireless Networks," IEEE Transactions on Vehicular Technology, vol. 57, no. 4, pp. 2632-2638, 2008.

[25] Z. Zeng, B. Veeravalli, "On the Design of Distributed Object Placement and Load Balancing Strategies in Large-Scale Networked Multimedia Storage Systems," IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 3, pp. 369-382, 2008.

[26] A. B. Prasetijo, E. D. Widianto, E. T. Hidayatullah, "Performance comparisons of web server load balancing algorithms on HAProxy and Heartbeat," 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, pp. 393-396, 2016.

[27] J. E. C. de la Cruz, I. C. A. R. Goyzueta, "Design of a high availability system with HAProxy and domain name service for web services," IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Cusco, pp. 1-4, 2017.

[28] L. H. Pramono, R. C. Buwono, Y. G. Waskito, "Round-robin Algorithm in HAProxy and Nginx Load Balancing Performance Evaluation: a Review," International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, pp. 367-372, 2018.

[29] H. H. Yang, Y. Wang, T. Q. S. Quek, "Delay Analysis of Random Scheduling and Round Robin in Small Cell Networks," IEEE Wireless Communications Letters, vol. 7, no. 6, pp. 978-981, 2018.

[30] A. Khoshkbarforoushha, R. Ranjan, R. Gaire, E. Abbasnejad, L. Wang, A. Y. Zomaya, "Distribution Based Workload Modelling of Continuous Queries in Clouds," IEEE Transactions on Emerging Topics in Computing. Volume: 5, Issue: 1, pp. 120 – 133. 2017.

[31] A. Kiani, N. Ansari, "Toward Low-Cost Workload Distribution for Integrated Green Data Centers," IEEE Communications Letters, Volume: 19, Issue: 1, pp. 26 – 29, 2015.

[32] X. Wang, B. Veeravalli, "Performance Characterization on Handling Large-Scale Partitionable Workloads on Heterogeneous Networked Compute Platforms," IEEE Transactions on Parallel and Distributed Systems, Volume: 28, Issue: 10, pp. 2925 – 2938. 2017.

[33] W. M. A. Ahmed, S. A. Fomenkov, S. V. Gaevoy, "Reducing Approximation Time of Cluster Workload by Using Simplified Hypergamma Distribution," International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Moscow, Russia, pp. 1-5, 2018.

[34] L. Zou, Z. Wang, H. Gao, X. Liu, "State Estimation for Discrete-Time Dynamical Networks with Time-Varying Delays and Stochastic Disturbances Under the Round-Robin Protocol," IEEE Transactions on Neural Networks and Learning Systems, Volume: 28, Issue: 5, pp. 1139 – 1151, 2017.

[35] K. A. Torkura, M. I. H. Sukmana, F. Cheng, C. Meinel, "Leveraging Cloud Native Design Patterns for Security-as-a-Service Applications," IEEE International Conference on Smart Cloud (SmartCloud), New York, NY, pp. 90 – 97, 2017.

[36] J. Wang, X. Chen, X. Huang, I. You, Y. Xiang, "Verifiable Auditing for Outsourced Database in Cloud Computing,"

IEEE Transactions on Computers, vol. 64, no. 11, pp. 3293-3303, 2015.

[37] V. Boisselle, B. Adams, “The impact of cross-distribution bug duplicates, empirical study on Debian and Ubuntu,” IEEE 15th International Working Conference on Source Code Analysis and Manipulation, pp. 131 – 140, 2015.