International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Efficient Hybrid Load Balancer for Software Defined Networks using OpenFlow Accuracy Prediction

Ananth B.

Submitted: 04/02/2024 **Revised**: 11/03/2024 **Accepted**: 17/03/2024

Abstract: Cloud computing is a global vision for real-world IT offerings where data and resources are integrated by web-based cloud management organizations using hardware and structured, primarily web-based packages. people at a reasonable cost. Sharing resources can cause problems with access to those resources, leading to a crash. The strategy for distributing network traffic across multiple connecting node or servers is called as load balancing. It is referred that no server is overloaded. Load control builds user responsiveness by distributing shares evenly. It also makes projects and sites more accessible to customers. The reason for this archive is to understand the billing control. It has associated structures of communication organizations over the Internet. Load balancing is an important part of a distributed computer to stay away from work overload and provide equally important support. Different statistics are used to determine system complexity

Keywords: *Cloud Computing, Load Balancer, Service Offering, Virtualization, Scheduling.*

1. Introduction

Cloud Computing is optimizing the uploads that refers to the spread of customer requirements across all multi-platform server applications running in the cloud. In addition, Cloud Load Tuning with a variety of load scales helps us maximize application performance and static quality [1]. Benefits of dispersing resources from common sources nearby (usually) to reduce costs and improve or reduce interest rates to meet demand [2].Typical load balancing agreements rely on machine information and require sophisticated IT staff to access, modify, and maintain. Only major partnerships with organizations with high IT spending plans can achieve better results and better quality, immovable: they are not cloud-based load configurations, as cloud-based primary providers often do not allow their knowledge to be personalized [3]. Fortunately, setting a load balance can provide team leaders with consistency while incorporating a small portion of the cost. Working with essential equipment, they also have access to private businesses.They are also good for improving cloud load as they can have cloud control just like any other app. A distributed computer can be a permanent and flexible framework for the entire asset unit [4] .All the assets of the framework must work together to meet customer needs. A distributed computer is the distribution of various offerings such as capacity, servers, organizations, scheduling, information and online research to transfer faster progress, streamline multiple resources and measure website savings. In this paper we addresses the following challenges and contributions to design our proposed work.i There are various fuzzy strategies for virtual machine migration to reduce power consumption. Cloud Sim is currently used as a common framework for validating models. We want to match these models with other kinds of applications to avoid an uncertain operating environment. Intensive analysis is done for virtual machine migration wherever the entire unit of the model domain is applied in a single cloud. That is, the migration of the virtual machine takes place among the DCs of the selected cloud, we want to redesign the algorithmic rule that can be applied in the inter-cloud to solve the migration problems [5][6].ii. Completely different unit area models designed to overcome the uncertainty and quality between cloud service provider and cloud service users regarding service quality. Different types of unit area attributes are designed for different models [7]. Extensive analysis is required to overcome the dual challenges that will arise between the cloud service provider and the user, and a knowledgeable technical unit is required to measure the complete quality model [8].iii. Much of the victimization math logic analysis is done in the area of resource improvement, scheduling, and repair reliability in cloud computing, but they need the overlooked area of knowledge storage and security in cloud computing using math logic. Deep analysis is necessary to overcome the problems of information storage and security [9]. Funds Open to all Countless Customers can have a website or one at any time Visit a web-based application The ability of a web application to address these customer needs per second becomes one of the most effective programs to reduce PC usage through a number of events [10]. This paper organize as follows, section 2 discuss above various related works deals about literature of different load balancing techniques and unbalancing issues. Section 3 explains load balancing strategies and their open flow stack for processing cloud. Section 4 describes the load balancer operations and carried out each stage process based on OpenFlow conditions. Section 5 gives experimental setup and simulations. Section 6 explains about conclusion and further implementation of research works.

*1*Department of Computer Science and Engineering, Puducherry Technological University, Puducherry-605014, India*
ORCID ID : 0000-0002-6274-3112
** Corresponding Author Email: ananth89bala@pec.edu*

2. Related Works

With the bulk rating, you can split and distribute the bond between at least two servers in the cloud. This allows you to adjust your base to extend the movement, review asset selections and achieve a less responsive response time [11]. is recommended in all cases, whether you want at least one friend: By using stack balance under all circumstances [12][13].We can deal with at least parallel needs, guaranteed management compliance, high traffic management and scheduling Emergency application requests are required in configuration mode and information, for example two unambiguous cloud servers [14] [15]. Cloud Load Tuning is the process by which process resources and obligations are distributed to at least one server. Posting ensures the most extreme execution in a short lead time. At least two servers, hard drives, corporate communications or other PC assets are isolated to enable the optimal use of assets and the expected response time of system operations [16].

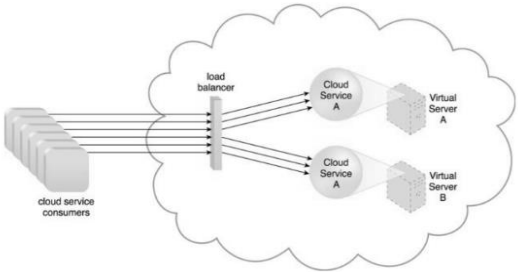

Fig 1: Various Cloud Services offering process with Load Balancer

Utilizing this kind of setup, the heap balancer acknowledges network demands and sorts them between at least two servers in the cloud displayed in fig.1. Cloud load adjusting is an interaction that circulates jobs and ascertains assets in a distributed computing climate. With load adjusting, associations can deal with the application or required responsibility by dispensing assets to various PCs, organizations, or servers [17].Virtualization is about increasing the utilization of resources and minimizing the cost of operation in DC clouds [18]. a completely different kind of input parameters area units used to realize the above mentioned goal, however we want to think about a complete application running on top of the VM to choose an acceptable physical machine to host these VMs. This provides an opportunity to think about the communication dependencies between VMs in multi-tier enterprise usage, as well as the elementary DC network topology and most limitations of the physical machine in DC [19]. The various literatures are discussed and below table 1 shows that comparative analysis of load balancer with results.

Parameters	Objectives	Methods	Merits	Drawback	Contribution with year
Throughput	Selected based on request feed	Heterogeneous	Scalability	Handling large volume dataset	Wong et al, 2019
Bandwidth	Handled multi object resource provisioning	Statistical Analysis	Global State access	Server Limitation	Zung et al, 2020
Response time	Zero down time access	Open VM	Physical resource provisioning	Live VM migration	Russra and Hugne et al, 2021
Agent Interaction	Replica based assessment	Open ML	Large scale data access	More time consuming and Global access issue	Manikandan et al, 2022
Dynamic Load	Greatest resource allocation	Dynamic VM	Better response time	Cloud data center accessing issues	Xia and Xunge et al, 2022
Turn around time	Minimum execution time	Deep Learning	Optimal resource provisioning	Balancing not possible to recommend	Ringa and Winge et al, 2022

Table 1: Various related results with comparison

Based on above comparison of various related work we are taken analytics study. In this case major challenges are considered such as load balancing, data center limitation, global network access and live migration. We considered above issues to our proposed load balancing strategies in software defined network.

3. Load Balancing Strategies in Software Defined Network

Load adjustment is the movement of organizational traffic through server collection. Stack balance is the server that does this. Load difficult tests in response to team and program performance. It is more affordable and easier to use. Organizations can deliver their customer requests quickly and efficiently through cloud services. Configuration helps keep site traffic light and responsive. It can integrate extended customer traffic with productive uploads from balances and steals it from multiple servers. or orchestration instruments. This is especially important for online sites that handle large numbers of online visitors per second. These productive load balances are essential for the delivery of work in contracts or other special management processes. Adaptation to traffic peaks - With any statement of results, a functioning university environment may fail completely due to the potential for serious queries. You do not need to load too much when using Cloud Load Balances. Reduce response time to customer needs or drives in the organization. Table 2 gives the load balancing reference stack for selecting and optimizing resources.

Measuring load by using following reference stack: • *Managing the traffic over the height on server and controlling the loads* • *Reduce the response times to users' requests* • *Increase utilization of resources* • *Significant improvements in performance* • *Maintain system consistency* • *Increase system flexibility in response to changes.* • *Working time and waiting time in line are reduced* • *Improve services at user level*

Table 2: Load Balancing – Reference Stack

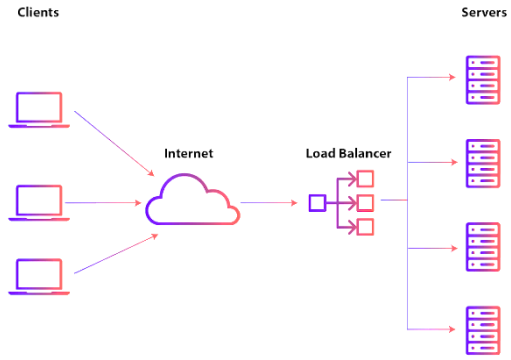

Fig 2: SDN Load Balancer in Hybrid Cloud

A cloud organization can be described as different server farms that have different CSPs connected to each other and that share their assets for efficient administration. In a unified cloud, assets are typically distributed and controlled among CSPs regarding various elements such as asset utilization, local workloads, and legal issues. In this unique circumstance, transferring the benefits of the cloud, which are necessary to transfer the application to different structures within the alliance in light of their security prerequisites, can achieve the ideal level of security required for the safe operation of the application shown in fig.2. Improving the security of an application mainly depends on three points of view:
i Limiting security spending due to security overhead, moreover, the unhappiness that comes from security frustrations,
ii. Providing an application with the appropriate level of fulfillment security, as providing lower or larger sums may prompt specialized or later again money-related misfortunes,
iii. Submitting an application with a slight chance of danger. For example, the frustration of information security or, again, availability due to a lower than required level of security can damage the reliability of the application and lead to a drop in the number of customers; the application may also experience the ill effects of Quality of Service (QoS) degradation due to excessive security, which may be redundant at times, leading to Service Level Agreement (SLA) violations.

4. Hybird Load Balancer

Load adjustment is also a basic piece of sky lift. The base of the clouds needs to stretch freely to handle both upward and downstream traffic. When the cloud "scales" it usually uses multiple visible the server and used for different variety of applications. The part that spreads traffic between these new looks is the battery rating.

Without a load gauge, virtual servers may not be able to detect incoming traffic in a planned manner or think they do not think so. It can also detect inaccessible servers and continuous traffic for those who still need it. Depending on the stack tuning statistics, load loaders can even test whether a given server (or group of servers) is more likely to be managed faster and router traffic to other hubs than these potential for your cloud management capabilities.
Performing overhead adjustments may also be important to achieve a distributed green computer. The explanations for this are: Limited power consumption - Tune loading can reduce power consumption by limiting unnecessary debt to key harbors or virtual machines. The different openflow implementation steps are shown in below table 3.

Software defined network	Load Balancing at Server	Virtualization	Load Balancer Service
Function the network using different layout modules and operation level service	Network services and delivery the contents using optimization	Manage Real Chargers on virtual machine	Meet out the traffic loads and find the each service offering values
Performance	Website Traffic	Traffic Effect	Flexibility
Finding the smaller value of each content and measure the performance of each node	Managing traffic, find the loads and offer the service based on customer level dependencies	Balance the traffic and calculate each bundle for loads	Demonstrates flexibility, flexibility and ability to handle traffic.

Table 3: Proposed Hybrid Load Balancer – Conditions

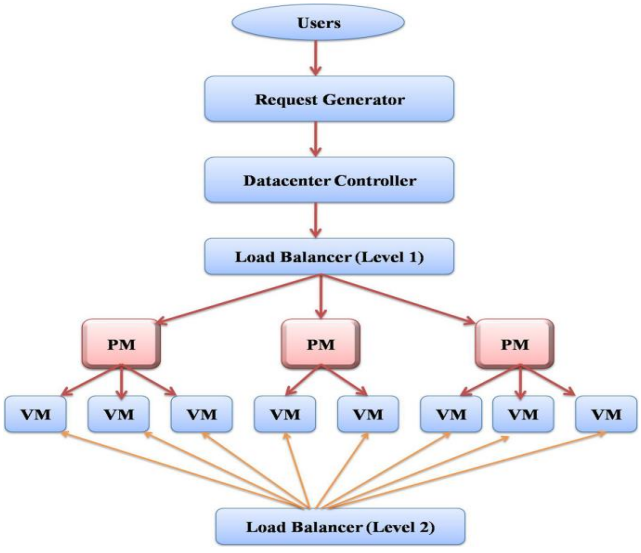

Fig 3: Load Balancer operations in VM level

5. Operational Level Service as Load Balancer

The load balancing is categorized according to the current state of the system and it has types:

a. Static Load Balancing:
In the case of stack adjustments, the stacking method is "static" which is thought to ignore the frame structure. Framework incorporates sports as the standard for a wide range of different processors (and in a few cases as well as floods). For all the considerations, a comprehensive draft framework is made ahead of time, for example, the appearance and proximity to the requirements of the asset. The numbers, power and connectivity of the CPUs are well known.Solid load adjustment is for this reason with the intention of registering the integration of functions with hand processors so that the graphics of the exhibition are limited. Solid load balances are often focused on the switch, or expert, who balances and upgrades the bulk. These small numbers may recall almost all the tasks to be dissolved and the normal performance time. The beauty of standing figures is that without a doubt the endless promise of inclusion and amazing ability.

b. Dynamic Load Balancing:
Flexible statistics take into account the current mass of each IT unit (also known as harps) instead of standard tuning methods. Organizations can then move from overcrowded-to-overcrowded areas for better management. they are very difficult to build altogether, they can produce amazing results, especially if the kill times are very different from message to message. Since you do not need a separate work evaluation center, a strong load balancing setting can make it more flexible. It is a unique activity where businesses with a processor are provided according to their circumstances. Flexible function, on the other hand, refers to the ability to rearrange shares in a continuous structure as indicated by the framework, and apparently involves the calculation of the equilibrium of the masses which requires the greatest interaction to reach its conclusion, risk. deviation from the common purpose of the problem. The below fig.4 is shows that the AWS cloud optimizer with openflow policies.

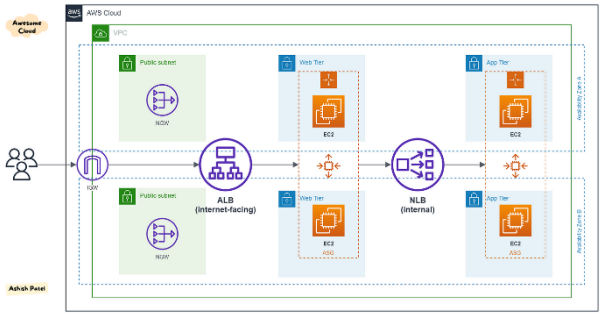

Fig 4: AWS Cloud Optimizer with Load Balancer Input and Parameters – Dataset Optimization

Simulation of Load Balancer based on Static and Dynamic

This is the default load balancing method in AWS Code Ninja and has no directive values such as weight, max_fails, fail_timeout, down, backup, slow_start, max_conns

```
upstream app {
   server s1.application.com slow_start=30s;
   server s2.application.com weight=3;
   server s3.application.com max_fails=5;
   server s4.application.com fail_timeout=10s;
   server s5.application.com backup;
}
upstream app {
   server s1.application.com;
   server s2.application.com weight=3;
}
upstream app {
least_conn;
   server s1.application.com;
   server s2.application.com weight=3;
}
upstream app {
ip_hash;
   server s1.application.com;
   server s2.application.com weight=3;
}
upstream app {
   hash request_uri;
   server s1.application.com;
   server s2.application.com weight=3;
}
```

weight=3 means s2 will be selected 3 times as much as the other servers. The default weight is 1
slow_start=30s specifies the time (30seconds) a server which was down is given to recover its operations before being overburdened with connections, which may cause it to go down again.
max_fails=5 means there should be 5 timeout connections within the period specified by fail_timeout directive, before the load balancer marks a server as inoperative

Table 4: Simulation setting load balancer parameters

A finite number of virtual machines is selected and the lowest probability values are set. If any error occurs, the delay means that a genetic algorithm is used to select the virtual machine from the cluster groups,
ϕ is a coefficient to measure the completed transaction.
$\Phi = Q * [1(Tp+d-n)]$
This case resolves the minimal failure and the transaction status is recorded. The objective weight function is calculated using the completed transaction and burst time of each process. Each recorded transaction and execution time are calculated as follows:
$Tj(Vj) = Q \text{ X } Mr(Qj) \sum mj = Tp \text{ X } (Qj))$
Based on the above results, CPU processing time and job execution time are recorded. From the above table. 3 shows that the result of burst time, execution time and accuracy factors. In this case, different virtual machines are classified and the capacity level is fixed in 4 levels 5, 10, 15, 20 respectively. A virtualized genetic algorithm is applied and the accuracy is calculated. Our proposed average system accuracy is 96%, which is achieved on different VMs and different capacity values with a load of 100 with Table 5. Based on the Table. 5, the accuracy of our proposed system is higher than 90% compared to other existing methods.Visual flow: The concept is to make the machine into a document or record. An important part of the information can be used in conjunction with its postponement of OK. A powerful load gauge should consider changing rapidly the mathematical requirements, memory, gadget

layout, and so on.

Number of VMs	Load Balancer	Weight	Hop Count	Accuracy (%)
5	2,5,10,20	50,50,50,50	10,10,10,10	87,88,89,88
10	2,5,10,20	50,50,50,50	10,10,10,10	91,92,90,92
20	2,5,10,20	50,50,50,50	10,10,10,10	92,92,91,93
50	2,5,10,20	50,50,50,50	10,10,10,10	89,88,90,91
100	2,5,10,20	50,50,50,50	10,10,10,10	92,92,91,92

Table 5: Test bed values and results

Experimental results prove that the proposed ranking model performs better in terms of average response time compared to the non-ranking model in federated architecture. The simulation results are shown in Table 6. The average response time is defined as the time between when the user requests the service and when the service is actually accessed.

Number of users	Selection based on Ranking Model (ms) with Load Balancer	Selection Without Ranking Model (ms) Without Load Balancer
1000	1.68	3.25
5000	1.78	4.21
10000	1.92	5.31

Table 6: Average Response time each ranking model and users

Step 1: Initial stage
In the genetic algorithm, a population of chromosomes Si (i=1, 2, 3...N) is first created randomly. N represents the population size.
Step 2: Assess suitability
Evaluate the fitness function of each chromosome and select the highest fitness value as the best one.
eligibility >= amount if value, ratio value, line and loop coverage
Step 3: Go to the best solution
One or more parental chromosomes are selected and perform a single point crossover
Step 4: Mutation
In the process of mutation, the values of the chromosomes differ depending on the possibility of the subsequent formation of a new chromosome.
Step 5: Update
At this stage, the current chromosome is replaced by a new one
Step 6: Discover the fitness function
If the fitness value of the new chromosome is greater than the current chromosome. Choose a new chromosome - a better chromosome.

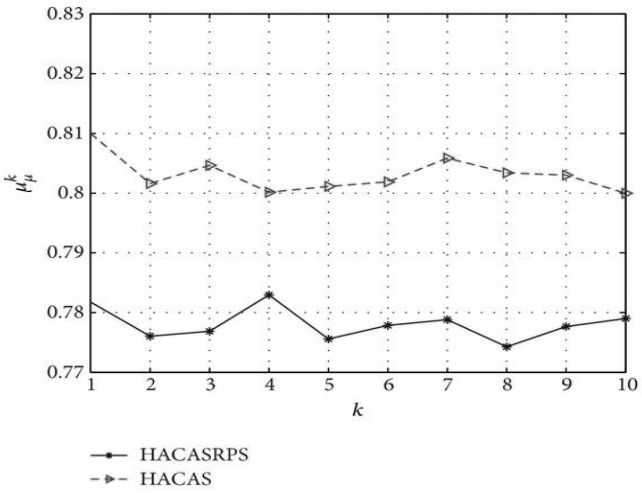

Fig 5: No. of VMs and response time results of with and without hybrid cloud balancer result

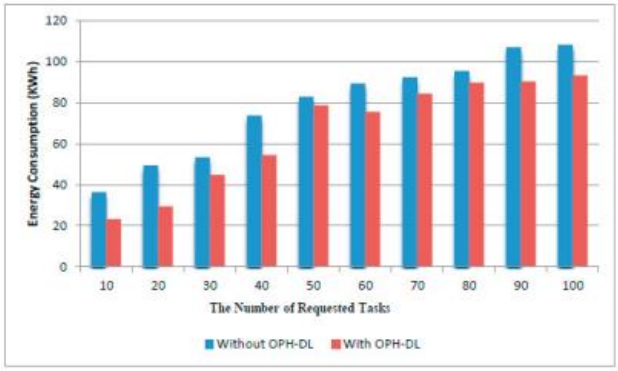

Fig 6: Response time of each VMs based on Load Balancer result with and without ranking results

Cloud Provider	Number of Task	Clustering Accuracy	Time
CP1	20,40,60,80,100	1.25	0.35
CP2	20,40,60,80,100	1.78	0.38
CP3	20,40,60,80,100	1.98	0.34
CP3	20,40,60,80,100	2.52	0.32
CP4	20,40,60,80,100	2.98	0.38
CP5	20,40,60,80,100	3.13	0.41

Table 7: Cloud Service Provider results based on load balancing conditions and openflow routing.

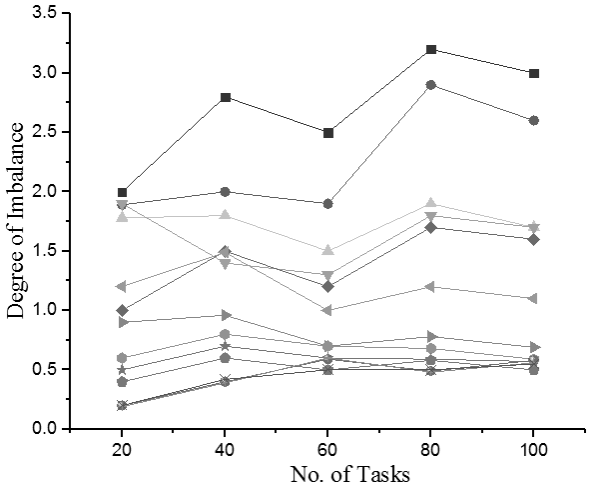

Fig 7. No. of tasks and imbalance conditions (openflow) based on iterations

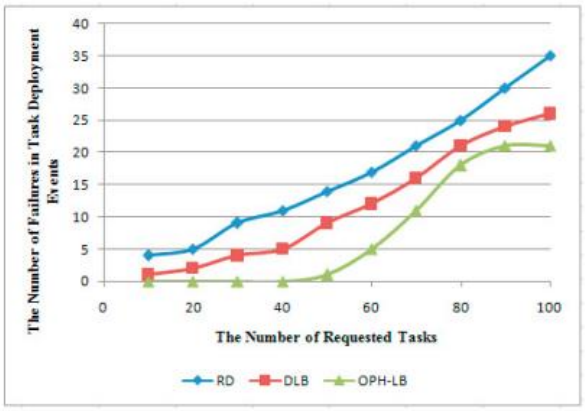

Fig 8: Simulation results of AWS Code Ninja comparison status with OpenFlow conditions.

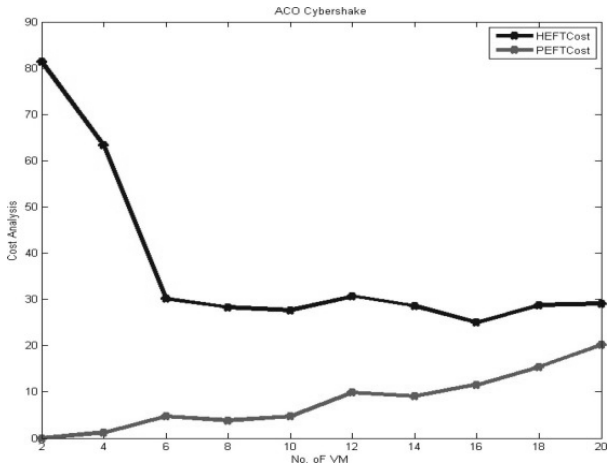

Fig 9: Average simulation results based on OpenFlow condition and VMs

The above simulations our proposed system compared with existing methods using openflow conditions. The results are compared with existing methods such as support vector machine cloudsim optimizer, open ML and virtualized load balancer. The below table 8 shows that comparison results

Methods	Number of Task	Avg. Clustering Accuracy	Avg. Response Time	Avg. Turn Around time
Support vector machine	20,40,60,80,100	7.18	2.21	2.78
Cloudsim optimizer	20,40,60,80,100	8.92	1.12	3.13
Open ML	20,40,60,80,100	10.12	2.21	2.23
Virtualized Load balancer	20,40,60,80,100	7.39	3.12	1.31
Hybrid Load balancer	20,40,60,80,100	3.13	0.41	0.78

Table 8: Comparison of proposed system with existing methods

Cloud computing is one of the major IT developments for storing and retrieving resources over the Internet. Most industries use online resource sharing for storage, infrastructure, and application development. It's a discounted price and a way to pay for usage. Load balancing is the main important factor for implementing and efficient cloud services. At the same time, access to more resources means that load balancing is essential. This paper proposes a virtualized hybrid load balancer implemented Our experiments are performed using a cloud simulator and achieve good accuracy rates. Our proposed system compares with existing methods and our system gives good accuracy compared to others.

2.1. Spending on Research, and Development (RDE)

The relationship between research and development (R&D) spending and its impact is a complex terrain marked by varying findings. While the absence of a positive effect of R&D expenditure on exports, other studies emphasize the value brought by innovation and underscore R&D as a conduit to fostering innovation. Intriguingly, the benefits of investing in employee training have been positioned to surpass those of R&D spending [36]. The upward trajectory of R&D expenditure, particularly in conjunction with Information and Communication Technology (ICT), has been noted [37]. However, studies probing the correlation between sustainable development and R&D remain limited. While [38] establish a positive correlation between socioeconomic well-being and R&D expenditure—a cornerstone of sustainable development—[39] introduces the intricate influence of ICT, R&D, and technical know-how on Total Factor Productivity (TFP). Amidst these divergent perspectives, this study's focus on R&D expenditure and its relationship with the Digital Adoption Index (DAI) assumes prominence. Through this exploration, the study seeks to disentangle R&D's role in technological integration, offering insights into its multifaceted implications within the realm of sustainable development.

5. Conclusion

Cloud computing is a strategy where different customers can access different services on the web depending on their needs. However, there are major obstacles to computer distribution. Load

control is a major obstacle to distributed computer. Many static and variable statistics are tested in this dose. The cloud is known as heterogeneous in nature. Sturdy statistics make it easy to visualize and explore the weather, but do not re-create the mixed concept of clouds. Variable load adjustment figures are difficult to demonstrate, but they all fit the different view of the cloud conditions. A distributed computer fixes costs associated with head reduction, continuous improvement of framework performance, framework enhancement, management time and governance.

References

[1] Khan Z, Singh R, Alam J, Saxena S (2020) Classification of Load Balancing Conditions for parallel and distributed systems. International Journal of Computer Science Issue 8: 411- 419.

[2] Gulati A, Chopra RK (2019) Dynamic Round Robin for Load Balancing in a Cloud Computing, International Journal of Computer Science and Mobile Computing: 274-278.

[3] Shahbaz Afzal and G. Kavitha, "Load balancing in cloud computing – A hierarchical taxonomical classification", Journal of Cloud Computing, volume 8, pp.981-991, 2019.

[4] Wong and William, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment using Cloud Analyst", IJESC, Volume 7 Issue No.3, 2019

[5] Ms. Shalini Joshi ,Dr. Uma Kumari "Load Balancing in Cloud Computing:Challenges& Issues" , Conference: 2021 2nd International Conference on Contemporary Computing and Informatics, Singapore, 2021

[6] S.Manikandan,M.Chinnadurai, D.Maria Manuel Vianny and D.Sivabalaselvamani, "Real Time Traffic Flow Prediction and Intelligent Traffic Control from Remote Location for Large-Scale Heterogeneous Networking using TensorFlow", International Journal of Future Generation Communication and Networking, ISSN: 2233-7857, Vol.13, No.1, (2020), pp.1006-1012.

[7] Manikandan, S, Chinnadurai, M, "Effective Energy Adaptive and Consumption in Wireless Sensor Network Using Distributed Source Coding and Sampling Techniques",.*Wireless Personal Communication* (2021), 118, 1393–1404 (2021).

[8] Hongsuk Yi, HeeJin Jung and Sanghoon Bae, "Deep Neural Networks for Traffic Flow Prediciton", IEEE Conference on Big Data and Computations, 2019, 971-5090-3015-6/17, pp.328-331

[9] 9. Ring and Winge, "TensorFlow:Large-Scale Machine Learning on Heterogeneous Distributed Systems", IEEE Access, vol.21, pp.267-281, 2021.

[10] N.Kato, "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," IEEE Wireless Communication, vol. 24, no. 3, pp. 146–153, Jun. 2020.

[11] S. Manikandan, P. Dhanalakshmi, K. C. Rajeswari and A. Delphin Carolina Rani, "Deep sentiment learning for measuring similarity recommendations in twitter data," Intelligent Automation & Soft Computing, vol. 34, no.1, pp. 183–192, 2022.

[12] Manikandan, S., Chinnadurai, M. (2022), "Virtualized Load Balancer for Hybrid Cloud Using Genetic Algorithm", Intelligent Automation & Soft Computing, 32(3), 1459–1466, 2022

[13] S. Sahu and P. Manish, "Efficient load balancing algorithm analysis in cloud computing," in Proc. of the Fourth Int. Conf. on Communication and Electronics Systems (ICCES 2019China, vol. 15, no. 6, pp. 177–184, 2019.

[14] Karan and Bhalodia, "An efficient dynamic load balancing algorithm for virtual machine in cloud computing," in Proc. of the Int. Conf. on Intelligent Computing and Control Systems (ICICCS 2019India, vol. 11, no. 4, pp. 215–224, 2019.

[15] K. Sheetal, "Enhanced improved genetic algorithm for load balancing in cloud computing," Journal of Research in Computer Science and Engineering, vol. 4, no. 1, pp. 275–286, 2020.

[16] S. Kripa and K. Kosal, "SIQ algorithm for efficient load balancing in cloud," IEEE Int. Conf. on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), Singapore, vol. 21, no. 13, pp. 1756–1766, 2021.

[17] C. Priyanka and S. Sankari. "Comparative analysis on virtual machine assignment algorithm," IEEE Int. Conf. on Computing and Communication Technology, Tokyo, vol. 17, no. 3, pp. 526–533, 2019.

[18] D. Menno Dobber, M. Rob van dermei and K. Gerkoole "Dynamic load balancing and job replication in a global-scale grid environment: A comparison," IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 2, pp. 207, 2019.

[19] C. Zenon, M. Venkatesh, S. Shahrzad and M. Christopher, "Availability and load balancing in cloud computing," in 2011 Int. Conf. on Computer and Software Modeling (IPCSITIndia, vol. 14, no. 3, pp. 219–224, 2011.