iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35873

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

Understanding Load Balancing Algorithms and Their Need in Cloud Computing

Mridul Sadhu[1], Nandish B J[2], Pranav Shastri[3], Veena Gadad[4], Suma B[5], Mamatha G S[6]
[1, 2, 4, 5]Computer Science and Engineering, [3 6]Information Science and Engineering, RV College of Engineering

***Abstract:** With each passing day, the number of users on the internet increases and thus does the data associated with them. This implies an exponential increase in the data storage and data processing units of practically all the servers on the net. Considering that the underlying idea and original purpose of the internet was to share data and resources, it becomes an increasingly difficult task to manage gigantic amounts of data associated with the ever-increasing number of users. This is why Cloud Computing has been able to bring a revolution in the world of technology and become the indispensable part of the internet that it is today. In this paper, we review the topic of load balancing and the various load balancing algorithms upon different measures. We also try to understand the most challenging problems that the cloud faces and some new load balancing techniques inspired from the natural world.*

***Keywords:** Load balancers; load balancing algorithms; cloud computing; natural phenomena based load balancing*

I. INTRODUCTION

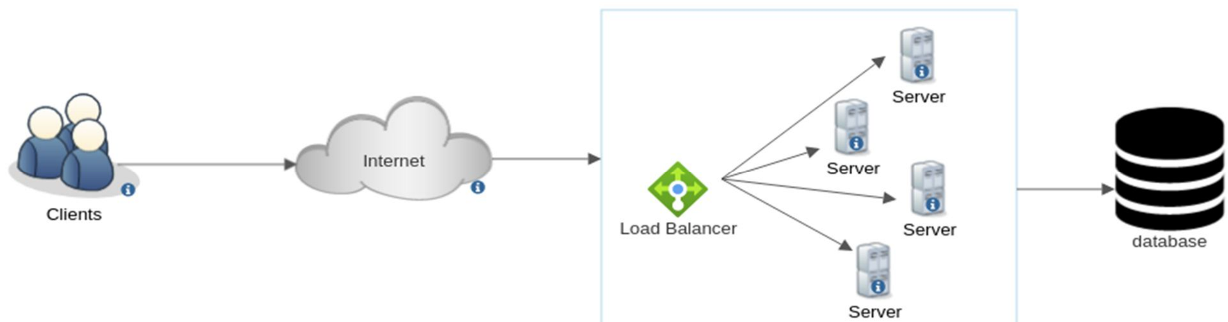


Fig. 1 General structure of cloud load balancer[17].

The distribution of storage and computing resources through the cloud is known as cloud computing. It is made up of two concepts in the area of technology. Cloud is the first concept and computing is the second. A cloud is a collection of resources, whereas computing is based on certain criterias[1]. Cloud Computing is mainly the distribution of computing resources like storage, servers, networking, databases, software, analytics, etc to end-users over the internet. These services can be dynamically allocated. It provides three levels of services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

Load balancing is a process of allocating the workload across all the available servers in the cloud. The workload is the total time taken to execute all the tasks, balancing the workload means managing the servers on the cloud. It Ensures that no cloud servers are idle or just partially loaded when others are fully loaded[2]. Some measurement parameters that can be used to measure load balancing techniques include throughput, efficiency, scalability, response time, resource usage, and fault tolerance[3]. Load balancing strives to distribute the workload on the cloud among different servers in a way that no server should be overloaded while some other server is idle. It helps to maximize throughput by minimizing the response time, load balancing as a service refers to load balancing in the cloud (LBaaS) [4]. Load balancing does the resource allocation and task scheduling to all the servers in the cloud. Load balancing is divided into two types: Static and Dynamic.

In Static Load Balancing, prior knowledge is required about servers' memory, processing power, capacity, etc. At runtime, these requirements are not subject to modification. On the contrary dynamic Load Balancing operates in a real-time environment, collecting continuous data about the server's load. It can't depend on prior knowledge, but it does consider run-time statistics. In heterogeneous and complex environments, dynamic load balancing produces better performance and it is more flexible[1].

II. LOAD ON THE CLOUD AND THE NEED FOR LOAD BALANCING

It can happen on cloud that while some servers are in idle state or partially loaded, other servers are heavily loaded. This problem leads to reduced performance ratio, decreasing resource utilization and maximizing response time. To overcome this problem, load balancing is used. It allocates every single task to servers and monitors the load status[2].

Need for load balancing are as follows:

A. Flexibility

A load balancer's primary goal is to save or shield a website from unexpected mishaps. When a workload is allocated over a number of servers, the load may be transferred to another server if one server is down.

B. Better Performance

Apart from their on-premise counterparts, cloud load balancing methods are less costly and easier to implement. Organizations will work on their clients' applications much more efficiently and with better results at lower costs.

C. Ability to handle sudden traffic spikes

Load balancers have the ability to manage any unexpected traffic that comes in at a specific time. For example, when concert tickets go on sale and the site completely goes down due to the fact that too many requests arrive at the same time. They won't have to worry about traffic spikes if they use cloud load balancers. However large or small the size of the request is, it can be wisely spread among different servers to yield the best results in the shortest amount of time.

D. High-availability

When several servers are used together, availability is increased. For example: if one server is down, the load is distributed to other servers, which ensures that services are not interrupted.

III. METRICS USED TO ASSESS LOAD BALANCING ALGORITHMS

How well a load balancing algorithm performs is reflected by it's score in a number of parameters or metrics which are used to judge the impact of the load balancer. The authors in [5], [6], [7], [8] and [9] have considered the following parameters: Resource Utilization, Throughput, Response Time, Performance, Fault Tolerance, Makespan, etc. The following topics explain these parameters in detail.

A. Resource Utilization

To take an analogy, we can consider the different resources as essential ingredients and the required final output to be a meal prepared from those ingredients. Obviously, the meal will be prepared best when all the ingredients are used in their optimum quantity. Similarly, a load balancing algorithm must utilize all the resources to their optimum best, not only for better performance but also to reduce the overall cost of the system.

B. Throughput

Throughput, even outside cloud computing environments, it generally means the amount of work or the number of tasks completed per unit of time. Even in the case of load balancers in cloud computing, it must be as high as possible for better results.

C. Response Time

Since this is the time taken by the entity in question, such as a load balancing algorithm here, to respond to any given request, it must be as low as possible.

D. Performance

This metric is much more general than the other ones. Put simply, we can just compare how the system fares when using two different algorithms to compare their performance in that particular environment. If using one particular algorithm gives results that are more desirable than the results achieved by using the other load balancing algorithm, then the performance of the first algorithm is better than that of the second one.

E. Fault Tolerance

This parameter represents how well a technique is able to execute itself when one node in the system fails. Optimally, the going down of one node should affect the performance of the algorithm as little as possible.

F. Makespan

The primary task of any load balancing algorithm is to allocate resources among various nodes in a system. The time taken to complete this allocation of resources by a technique is called its' 'Makespan'. It should be as low as possible.

IV. DISTRIBUTED LOAD BALANCING

In [10], the authors state that cloud computing was intended to share resources spread far wide from each other, rather than for collecting them on a local system. Naturally then, the load on the different nodes connected to such a cloud environment will be unevenly distributed and unequal, especially as the system size increases. As mentioned in [11], this is where load balancers are used to ensure that some nodes don't become a bottleneck while other nodes have very little or no work at all. The major difference between load balancing and distributed load balancing is the presence of central load balancers in traditional load balancing systems while they are not used in systems that use distributed load balancing. Nodes in distributed load balancing systems use reverse proxies as client side load balancers. The author in [12] basically explains a reverse proxy as a proxy server that intercepts requests from clients towards a web-page and then distributes these requests among various servers, all of which work for the same page. This not only balances the load on these servers but also protects them from malicious attacks since all the requests have to go through the reverse proxy and the address of the origin server is not directly available to the attacker. Distributed load balancing can also be achieved through allocation of Virtual Machines (VM) to all the Physical Machines (PM) in a system and then subsequently sharing of resources via sharing of VMs from one PM to another. In [13], the authors state that when in one such system, a node is identified which is overburdened, it faces the choice that which VM should it migrate and to which PM should it shift the selected VM to. Also in [13], the authors say that the method to find which VM to shift from the problematic PM is to first sort the VMs in the PM in decreasing order of their utilization and then migrate the VM with the highest utilization first. Distributed load balancing, thus, becomes essential for building an architecture with multiple resources that function together because of its innate ability to connect different resources with each service.

V. STATIC AND DYNAMIC LOAD BALANCING ALGORITHMS

Load balancing algorithms are classified based on the type of load balancers these algorithms incorporate:

A. Static Load Balancing

In this type of load balancing, it generally requires prior information about the system. The server's performance is determined at the compilation time[14]. As a result, the decision to move the load is independent of the current state of the device. Based on the performance, the remote server assigns the workload. All the other servers calculate the allotted work and submit their result to the remote server[15]. Static load balancing algorithms are non-preemptive. However, static load balancing algorithms have a drawback in that once task allocation is made to servers, it cannot be changed at execution time. The main goal of static load balancing algorithms is minimizing the execution time and reducing contact bottlenecks[16]. There are four types of static load balancing algorithms: round robin, randomized algorithm, central manager and threshold algorithm which will be described below.

1) *Round robin algorithm*: In this algorithm the load is distributed uniformly over all servers.it allocates tasks to all servers in a round robin order. Round robin algorithm selects each server in the list and assigns tasks, once it reaches the last server it again starts with the first server. Round robin is simple, starvation free and easy to implement, this algorithm gives better performance for special purpose applications and it does not require inter process communication[15].
2) *Randomized algorithm*: Randomized algorithms choose slave processors using random numbers. These random numbers are generated based on a statistical distribution[16]. Randomized algorithms give best performance compared to other load balancing algorithms for particular special purpose applications.
3) *Central Manager Algorithm*: The Central Manager Algorithm uses the central server as a master. This master server is coordinator and distributes the workload among the slave processors. The slave processor is chosen according to the rule of assigning the job to the processor with the least load[17].
4) *Threshold algorithm*: When the processes are created in this algorithm, they are automatically allocated to the server nodes. Without sending remote messages, nodes are chosen locally. Each node maintains its own copy of the system's load[15]. The load is characterized is in three levels: underload, medium and overloaded. Under loaded is where the load is less than tunder, medium load is where tunder is less than then load which in turn is less than tupper. And lastly we have overloaded load where the overall load is greater than tupper Initially, all nodes are considered to be under-loaded. When a node's load state reaches a load level limit, it sends messages to all remote nodes informing them of the new load state, keeping them up to date on the current load state[17].

B. Dynamic Load balancing

In this type of load balancing, workload is distributed at run time.The current state of the system is used to make load management decisions. Remote server is able to collect information about other servers and based on the information collected it allocates tasks to servers[16].There are two forms of interaction among servers to achieve load balancing:cooperative and non-cooperative. In cooperative the servers work together to achieve a common goal. For example:to improve response time of all servers. In non-cooperative, Each server operates on its own to achieve a goal that is specific to it., for example, to improve a local task's response time[15].

1) *Central Queue Algorithm*: On the main host, it keeps track of new events and unfulfilled requests in a cyclic First In First Out (FIFO) queue. Each new activity that the queue management receives is added to the queue. When a requester sends an activity request to the queue manager, the queue management takes the first activity out of the queue and gives it to the requester. The request has been queued if there are no activities in the queue. When a new job arrives at the queue manager while there are still unanswered requests in the queue, the first one is removed from the queue and replaced with the new activity.[15].
2) *Local Queue Algorithm:* The Local Queue algorithm's major feature is its support for dynamic process migration. When the host load falls below a certain threshold, the local queue algorithm uses process migration to handle all new processes locally. The load manager tries to provide a minimum number of ready processes on each processor with this parameter. All underloaded hosts receive new processes created on the main host in the first step. The number of parallel activities established by the main host's first parallel construct is usually sufficient to allocate all remote hosts.[16].

VI. POLICIES OR STRATEGIES IN DYNAMIC LOAD BALANCING

In the case of dynamic load balancers there are several policies that has to be understood and accessed before deciding on stuff:

A. Location Policy

The location policy is utilized to choose an appropriate transfer partner(destination node)[18]. This policy selects destination nodes based on local information[19]. Location strategy tries to find lightly loaded nodes so that heavily loaded nodes can share some tasks to lightly loaded nodes [20].

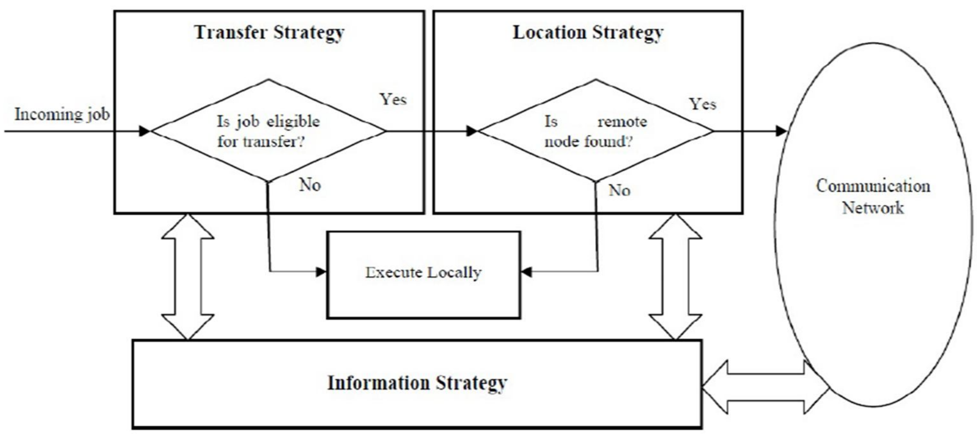


Fig. 2 Interaction between different components of a dynamic load balancing algorithm[20].

B. Transfer Policy

Transfer policy or Transfer strategy refers to the part of the dynamic load balancing algorithm that selects a job for transferring from a local node to a remote node[21].

C. Selection Policy

This policy is used to select jobs for transfer. In sender-initiated schemes, heavily-loaded nodes choose which tasks to transfer. Lightly loaded nodes tell senders of the kind of tasks they are willing to accept in receiver-initiated schemes. So the amount of load and how many tasks to transfer is determined by this policy.

D. Information Policy

The element of the dynamic load balancing algorithm that collects information about the nodes in the system is known as information policy.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

E. Load Estimation Policy
This defines how to estimate the workload of a specific system node[18].

F. Process Transfer Policy
This policy is used to identify the execution of a job that is to be done remotely or locally[21].

G. Priority Assignment Policy
This policy determines the order in which local and remote processes are executed at a given node.[20].

H. Migration Limiting Policy
It establishes a limit for migration and determines the total number of migrations from one node to another for a process[18].

VII. CURRENT CHALLENGES FACED BY CLOUD BASED LOAD BALANCING
Whenever a technological advancement grows at a rate as much as cloud computing is today, it is certain to come across various challenges. The primary reason for this is the enormous increase in the number of users each year which leads to a rapid increase in different quality of service metrics such as storage space, processing speed, etc. The same is true for cloud computing. The boom in cloud computing has led to an incessant demand for load balancers and load balancing algorithms which are more efficient than the ones currently in use. Although this has led to significant development in the field of load balancing, it has also posed many challenges. In [22], the authors list the following as the major challenges faced in load balancing for cloud computing: storage management, load balancer scalability, single point of failure, virtual machine migration and heterogeneous nodes. Here we review these challenges in some detail.

A. Storage Management
The authors in [23] say that the cloud allows heterogeneous access of data without any considerable problems in accessing it. It also solves most problems associated with the traditional storage systems such as high hardware costs. The cloud has two major replication policies: full data replication and partial replication. The problem with these strategies, respectively, are reduced efficiency on replication points and availability of the dataset. Partial replication also tends to make the load balancing algorithm complex.

B. Load Balancer Scalability
One of the most prominent features of the cloud, which has allowed it to manage huge workloads and become an almost ubiquitous part of the internet today is its' scalability and availability. Hence, it comes as no surprise that it is a constant challenge for industry researchers to look for algorithms that are more and more scalable than before in terms of the topology of the system, power of computing and storage in order for it to able to meet the ever increasing demands, as mentioned in [24].

C. Single Point of Failure
As we have discussed before in this paper, there are many scenarios in which some decisions related to load balancing have to be taken in a cloud environment, such as which VMs to transport from an overloaded PM. Due to various reasons, many load balancing algorithms have a single central node which takes all these decisions, especially in dynamic algorithms. This implies that if that central node fails, the entire algorithm might collapse. Thus it is a challenge to create an algorithm in which the failing of any one particular node doesn't cause the whole algorithm to break down.

D. Virtual Machine Migration
In cloud environments, it is quite common for a single PM to have multiple separately configured VMs, all of which operate independently of each other. Many of such VMs have to be transferred to different PMs if the current one gets overloaded. It is constantly being studied how to make these transfers faster and more efficient.

E. Heterogeneous Nodes
Traditionally, cloud computing has focused on homogeneous environments. However, as the cloud grows bigger and more devices and systems are connected to it, there is a growing need for load balancing algorithms to be developed for heterogeneous nodes. This is especially true for environments having a large number of VMs on a relatively small number of PMs, since the hypervisor will be able to manage different VMs much more efficiently.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI June 2021- Available at www.ijraset.com

VIII. NEWER TECHNIQUES FOR LOAD BALANCING INSPIRED BY NATURAL PHENOMENA

In [25], the authors reviewed various load balancing strategies that are inspired by nature or natural phenomena. These techniques include Cuckoo Optimization Algorithm (COA), Genetic Algorithm (GA), Ant Colony Optimization and HoneyBee Based Load balancing (HBB-LB). Let's try to understand these in an elaborate manner:

A. Cuckoo Optimization Algorithm

This method was proposed in [26]. As the load on the cloud increases, it becomes more and more crucial with each passing day for a load balancing algorithm to be energy efficient. COA was designed keeping this purpose in mind. The name of the algorithm is derived from the bird Cuckoo as this algorithm tries to emulate Cuckoos when they have to lay eggs. For this, unlike most other birds, Cuckoos don't build their own nest but rather search for nests of other birds whose eggs are most similar to the Cuckoo eggs. Then they sneakily put their egg/eggs in such a nest when the bird who built that nest is not present there. The load balancing algorithm proposed here works in 3 steps. The first step is simply the detection of over-utilized host nodes. Secondly, VMs from such an over-utilized host are transported to other hosts. For this, all hosts except the over-utilized one are considered under-utilized and it is attempted to transport all VMs of an under-utilized host to other hosts and then switch that host to sleep. Should this process fail, the under-utilized hosts are not put to sleep. The final step consists of selecting VMs which are to be transported. This is done by following the Minimum Migration Time (MMT) policy. A big advantage of this algorithm is that it is energy efficient but it comes with an inherent disadvantage, it can lead to a SLA breach.

B. Genetic Algorithm

Proposed in [27], this algorithm, as the name suggests, tries to function as a load balancer while at the same time it reduces the time taken to finish any given task set to as low as possible. The genetics of any living organism are such that they evolve over generations. For Example, in [28], the authors hypothesize that the ZIKA Virus has developed considerable changes in both its' protein and nucleotide sequences during the past 50 years. As a result, it has increased its' geographical range and severity of impact upon contraction. Trying to simulate this feature of the genetic algorithm in living beings, this load balancing algorithm produces several possible solutions to a given problem and then waits for them to 'evolve' over time to generate more efficient and beneficial solutions.

C. Ant Colony Optimization

The authors in [29] study a novel load balancing technique which is based on ants' behaviour, especially when roaming around searching for food. The ants display a very particular behaviour pattern in such a state: they continuously reassign weights which different ants in the group are carrying by moving forward to search and backwards to reassign. The technique here is based on this particular behaviour. First, a node is chosen which is surrounded by the maximum number of nodes. Then, moving forward through the nodes, a node is chosen which is overloaded. Some load is then taken off from this node and assigned to some other node which is under-utilized by moving backwards through the node cluster. A significant advantage of this method is that it is quick and efficient in searching for over and under-utilized nodes.

D. HoneyBee Based Load Balancing

The authors in [30] propose a load balancing algorithm that is inspired by the lifecycle and work ethic of bees. First, let us understand how do bees search for food. Whenever a bee or a group of bees find a food source, they return to their hive and through a special form of movement, inform the other bees in the hive about the food source, its' quality and quantity. Then the most physically fit and active bees are selected to go to the location and bring as much food as possible. The load balancing technique mentioned above works in a similar fashion and it is designed to reduce the waiting time of high priority tasks in the queue. Whenever a VM is overloaded, other nodes help in deciding which tasks are to be shifted and to which VM. Higher priority tasks are shifted first if the original node can't execute them at the moment. In this way, response time and waiting time of tasks, especially high priority ones, are reduced.

IX. CONCLUSION

Load balancing helps to maximize the performance and use of resources. In this paper,we discussed different load balancing algorithms and also compared static and dynamic load balancing techniques. Static load balancing is used when the system requires stability.This algorithm has performance issues and fails to account for the diverse nature of cloud. Dynamic load balancing is used when the system requires higher performance.This algorithm is best suited for the heterogeneous nature of cloud. We reviewed different metrics upon which a load balancing algorithm is judged.

The cloud, despite its' rampant usage and popularity, faces some stern challenges today. Those were discussed in the paper along with some newer load balancing techniques inspired by nature and natural phenomena.

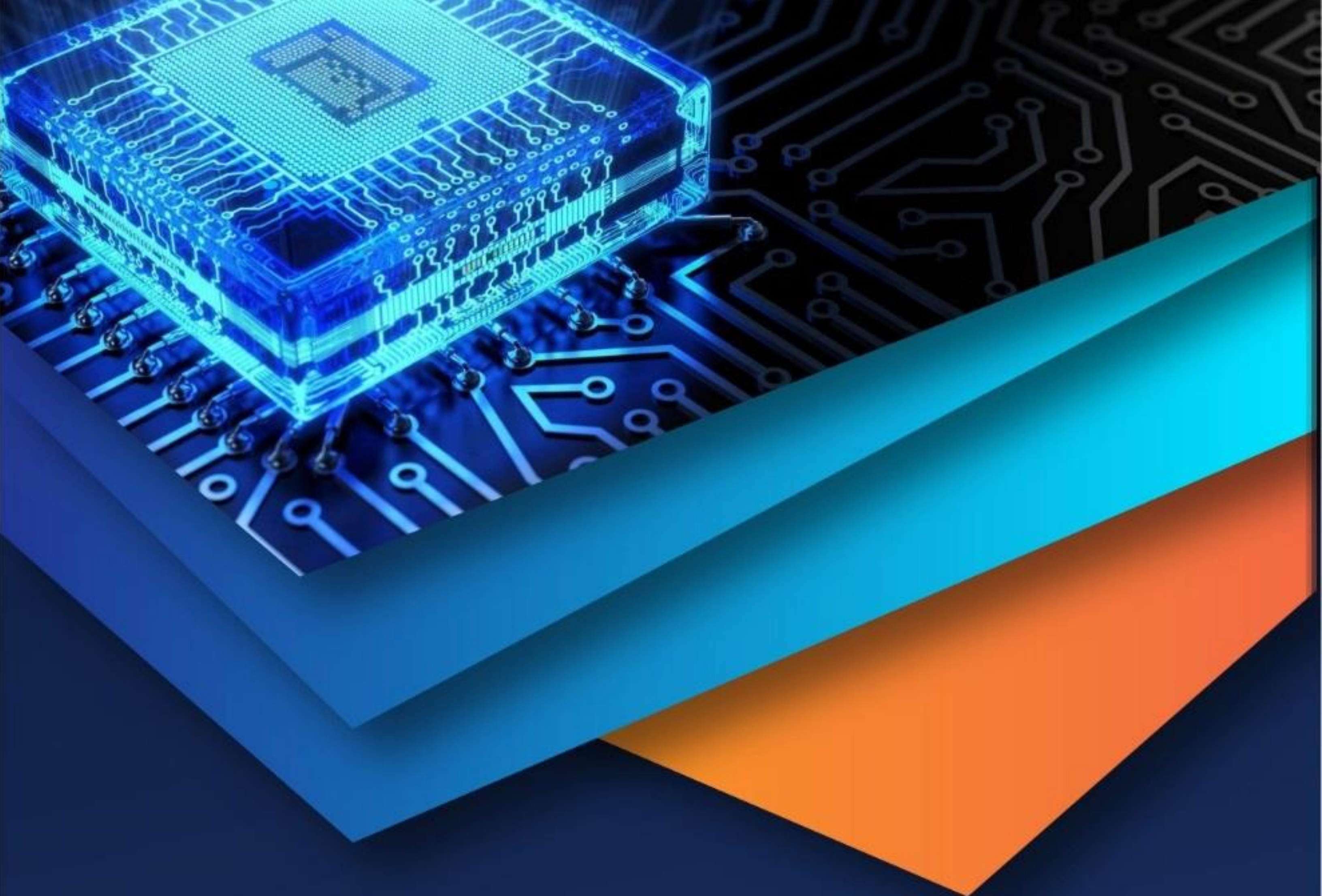
X. ACKNOWLEDGEMENT

We sincerely thank RV College of Engineering for their support in the presentation of this topic. We also like to thank our friends and family for supporting us throughout the work.

REFERENCES

[1] Katyal M, Mishra A. A comparative study of load balancing algorithms in cloud computing environment. arXiv preprint arXiv:1403.6918. 2014 Mar 27.
[2] Kherani FF. Prof. Jignesh Vania,"Load Balancing in cloud computing". International Journal of Engineering Development and Research. 2014;2(1).
[3] Sajjan RS, Yashwantrao BR. Load balancing and its algorithms in cloud computing: A survey. International Journal of Computer Sciences and Engineering. 2017 Jan;5(1):95-100.
[4] Ankita J, Aisha F, "Load Balancing Algorithms For Cloud Computing Environment ", International Journal of Emerging Technologies and Innovative Research (www.jetir.org | UGC and issn Approved), ISSN:2349-5162, Vol.7, Issue 5, page no. pp59-67, May-2020, Available at : http://www.jetir.org/papers/JETIR2005010.pdf
[5] Deepa T, Cheelu D. A comparative study of static and dynamic load balancing algorithms in cloud computing. In2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) 2017 Aug 1 (pp. 3375-3378). IEEE.
[6] Daraghmi EY, Yuan SM. A small world based overlay network for improving dynamic load-balancing. Journal of Systems and Software. 2015 Sep 1;107:187-203.
[7] Chen SL, Chen YY, Kuo SH. CLB: A novel load balancing architecture and algorithm for cloud services. Computers & Electrical Engineering. 2017 Feb 1;58:154-60.
[8] ABDULHAMID SI, Abd Latiff MS, Bashir MB. SCHEDULING TECHNIQUES IN ON-DEMAND GRID AS A SERVICE CLOUD: A REVIEW. Journal of Theoretical & Applied Information Technology. 2014 May 10;63(1).
[9] Ramezani F, Lu J, Hussain FK. Task-based system load balancing in cloud computing using particle swarm optimization. International journal of parallel programming. 2014 Oct;42(5):739-54.
[10] Randles M, Lamb D, Taleb-Bendiab A. A comparative study into distributed load balancing algorithms for cloud computing. In2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops 2010 Apr 20 (pp. 551-556). IEEE.
[11] Kansal NJ, Chana I. Cloud load balancing techniques: A step towards green computing. IJCSI International Journal of Computer Science Issues. 2012 Jan;9(1):238-46.
[12] Sommerlad P. Reverse Proxy Patterns. InEuroPLoP 2003 Jun 25 (pp. 431-458).
[13] Ma F, Liu F, Liu Z. Distributed load balancing allocation of virtual machine in cloud data center. In2012 IEEE International Conference on Computer Science and Automation Engineering 2012 Jun 22 (pp. 20-23). IEEE.
[14] Shah N, Farik M. Static load balancing algorithms in cloud computing: challenges & solutions. International Journal Of Scientific & Technology Research. 2015 Oct;4(10):365-7.
[15] Rajguru AA, Apte SS. A comparative performance analysis of load balancing algorithms in distributed system using qualitative parameters. International Journal of Recent Technology and Engineering. 2012 Aug;1(3):175-9.
[16] Elngomi ZM, Khanfar K. A Comparative Study of Load Balancing Algorithms: A Review Paper. International Journal of Computer Science and Mobile Computing. 2016 Jun;5(6):448-58.
[17] Beniwal P, Garg A. A comparative study of static and dynamic load balancing algorithms. International journal of advanced research in computer science and management studies. 2014;2(12):1-7.
[18] Kumar YR, Priya MM, Chatrapati KS. Effective distributed dynamic load balancing for the clouds. International Journal of Engineering Research & Technology (IJERT). 2013 Feb;2(2):1-6.
[19] Deepika DW, Kumar N. Performance analysis of load balancing algorithms in distributed system. Advance in Electronic and Electric Engineering. 2014;4(1):59-66.
[20] Alakeel AM. A guide to dynamic load balancing in distributed computer systems. International Journal of Computer Science and Information Security. 2010 Jun;10(6):153-60.
[21] Haroon M, Husain M. Different Policies For Dynamic Load Balancing. International Journal of Engineering Research And Technology. 2012;1(10).
[22] Kumar P, Kumar R. Issues and challenges of load balancing techniques in cloud computing: A survey. ACM Computing Surveys (CSUR). 2019 Feb 4;51(6):1-35.
[23] Wu TY, Lee WT, Lin YS, Lin YS, Chan HL, Huang JS. Dynamic load balancing mechanism based on cloud storage. In2012 Computing, Communications and Applications Conference 2012 Jan 11 (pp. 102-106). IEEE.
[24] Ray S, De Sarkar A. Execution analysis of load balancing algorithms in cloud computing environment. International Journal on Cloud Computing: Services and Architecture (IJCCSA). 2012 Oct;2(5):1-3.
[25] Ghomi EJ, Rahmani AM, Qader NN. Load-balancing algorithms in cloud computing: A survey. Journal of Network and Computer Applications. 2017 Jun 15;88:50-71.
[26] Yakhchi M, Ghafari SM, Yakhchi S, Fazeli M, Patooghi A. Proposing a load balancing method based on Cuckoo Optimization Algorithm for energy management in cloud computing infrastructures. In2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO) 2015 May 27 (pp. 1-5). IEEE.
[27] Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S. A genetic algorithm (ga) based load balancing strategy for cloud computing. Procedia Technology. 2013 Jan 1;10:340-7.

[28] Wang L, Valderramos SG, Wu A, Ouyang S, Li C, Brasil P, Bonaldo M, Coates T, Nielsen-Saines K, Jiang T, Aliyari R. From mosquitos to humans: genetic evolution of Zika virus. Cell host & microbe. 2016 May 11;19(5):561-5.
[29] Nishant K, Sharma P, Krishna V, Gupta C, Singh KP, Rastogi R. Load balancing of nodes in cloud using ant colony optimization. In2012 UKSim 14th international conference on computer modelling and simulation 2012 Mar 28 (pp. 3-8). IEEE.
[30] LD DB, Krishna PV. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Applied soft computing. 2013 May 1;13(5):2292-303.



10.22214/IJRASET

INDEX COPERNICUS 45.98

ISRA JIF IMPACT FACTOR: 7.129

SJIFactor TOGETHER WE REACH THE GOAL IMPACT FACTOR: 7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY