https://doi.org/10.58864/mrijet.2023.10.1.4

Load Balancing Algorithms with the Application of Machine Learning: A Review

Divyansh Singh
Department of CSE, FET,
Manav Rachna International Institute of Research and Studies, Faridabad, Haryana, India
E-mail: singhdivyansh1407@gmail.com

Vandit Bhalla
Department of CSE, FET,
Manav Rachna International Institute of Research and Studies, Faridabad, Haryana, India
E-mail: vanditbhalla3@gmail.com

Neha Garg
Assistant Professor,
Department of CSE, FET,
Manav Rachna International Institute of Research and Studies, Faridabad, Haryana, India
E-mail: nehagarg.fet@mriu.edu.in

Abstract: *Cloud computing is the provision of computing services over the web. Cloud Computing's load-balancing algorithms are applied in static, dynamic, and centralized environments. The article compares and summarizes several load-balancing strategies in the cloud computing environment and discusses the pros and cons of different load-balancing algorithms. The paper also mainly focuses on exploring machine learning models used in LB techniques. The most popular algorithms in the articles reviewed include Statistical Regression, Random Forest Classifier Artificial Neural Networks (RF), Convolutional Neural Networks(CNNs), and Regenerating Memory Neural Networks. long-term (LSTM-RNN). LB specifications have been defined through performance metrics such as throughput, latency, traveltime, fault tolerance, and power savings.*

Keywords: *cloud computing, load balancing, static load balancing, dynamic load balancing, machine learning.*

I. INTRODUCTION

Cloud computing provides end users with a pool of resources that can be quickly reallocated to other purposes with minimal latency. Cloud Computing Power can be an IT service rooted in computing power that provides end-users with infrastructure, applications, and more, wherever they are. Cloud computing combines hardware, software, networks, storage, services, and interfaces to create the computing-as-a-service aspect. There are many reasons to use cloud computing listed as follows:

- No hardware or software required
- OS independent
- Dynamic allocation
- Program movement
- Scalability
- Pay-as-you-go
- No commitments
- Many, Web-abstract infrastructures around the world

Cloud computing uses on-demand hardware as the foundation. Hardware is usually replaced at any time without affecting the cloud. It uses a product-based software container system. For example, a service must be available to move from one cloud provider to another without affecting the service.

Cloud computing requires virtualization, which is an abstraction layer of hardware and software. The services provided by cloud computing are:

- *Platform as a Service (PaaS)* - PaaS provides software developers with the necessary resources required for running, developing, and managing applications without having to deal with the maintenance of the resources.
- *Infrastructure as a Service (IaaS)* - IaaS provides the required infrastructure, such as physical and virtual servers, networks, and storage, which can be upscaled if the requirement increases.
- *Software as a Service (SaaS)* - It is also known as cloud-based software where the user has to pay a recurring subscription fee or pay as per usage, or as stated in the service agreement.

Cloud computing is divided into three types based

on the method of deployment used. The design and setup of these cloud environments can vary based on the technologies used.

- *Public Cloud* - A public cloud is an open setup model where a third partyis the cloud service provider and the resources are available to users over the public internet, anyone can access them for a subscription fee.
- *Private Cloud* - As the name suggestsin this type of cloud environment all the resources are dedicated to and accessible by one customer only.
- *Hybrid Cloud* - It is a combination of both public cloud and private cloud so as to overcome thedisadvantages of both of them, as it helps the organization to choose which one is suitable for which task.

II. LOAD BALANCING

When running workloads on the cloud, it is essential to distribute the load across a group of servers or a distributed system in such a way that the task efficiently utilizes the resources, while not hindering other processes. This process of distribution of workload toavoid system overload and crashes is called Load Balancing. Cloud computing load balancing is required to handle networked or decentralized systems so tasks are assigned to all or any processors for efficient use of resources. To facilitate global load balancing in the cloud, you need the right scheduling algorithms. Load balancing is the distribution of load order among multiple nodes in your network. It is the responsibility of the Algorithm Load Balancer to select the nexttask in a way that minimizes execution time and resource usage in the data center. Many load-balancing algorithms are used in cloud computing for load-balancing networks.

A load balancing algorithm aims at

- Maximizing the output of the network
- Minimizing the system overhead
- Ensuring the reliability of the system
- Scalability of the system
- Efficient utilization of resources
- Maximizing fault tolerance

Cloud Computing Load Balancing can be categorized into two categories as follows:

Static load balancing - It uses and provides advanced information about all task properties, computing resources, communication networks, and memory. A static load balancing algorithm may be a non-pre-emptive type.

Dynamic load balancing - It has no prior permission information. Dynamic load balancing moves roles from overloaded node to idle node. Dynamic load balancing algorithms are centralized or distributed, depending on whether you are responsible for global dynamic scheduling tasks that can be physically deployed on a single processor, or whether the work is associated involved in deciding whether to be physically distributed between processors. Dynamic load balancing algorithms can be distributed and undistributed. The CloudSim toolkit is used to simulate the load balancing algorithm and give power.

III. MACHINE LEARNING

It is a branch of artificial intelligence (AI) and computer science that focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. It focuses on training systems to perform new tasks without being explicitly programmed. It is further divided into two categories namely, Supervised Learning and Unsupervised Learning.

Supervised Learning: This algorithm consists of a target/outcome variable (dependent variable) that must be predicted from a certain set of predictors (independent variable). Usingthis set of variables, we create a function that maps the inputs to the desired outputs. The training process continues until the model reaches the specified level of accuracy on the training data. Regression and classification problems are key. Supervised learning techniques are linear regression, support vector machines, neural networks, decision trees, Naive Bayes, and nearest neighbors. It ismainly used in predictive modeling.

Linear regression technique: Simple regression performs the task of predicting a value of a variable quantity (Y) supported by a given experimental variable (X). This regression technique finds a linear relationship between the given parameters of the variable X (input) and Y (output). In the figure above, X (input) is work experience and Y (output) isan

individual's salary. The curve is the best fit for our model.

Linear Regression: Y = a1 + a2X Where X = input training data and Y = knowledge labels

When training the model, it matches the most efficient line to predict the value of Y for a certain value of X. The model obtains the simplest regression fit line by finding the simplest values a1 and a2. where a1 = Intercept coefficient and a2 = Factor of X.When we find the simplest a1 and a2 values, we get the most efficiently fitted line. So finally after using our model to predict it will predict the value of Y given the input value of X.

Unsupervised Learning: In this algorithm, we have no objective or outcome variable to predict/estimate. it is used to cluster a population into multiple groups, widely used to segment clients into different groups for specific interventions. it is used for clustering (clustering) problems, and anomaly detection (in banks for anomalous transactions) where it is necessary to determine the relationships between the information provided. Unsupervised learning techniques are k-meansclustering, K Medoids Fuzzy C-Means, Neural Networks, and Hierarchical. It is mainly used in descriptive modeling.

IV. IMPORTANCE OF LOAD BALANCING

It is one of the essential components of a cloud computing environment that sustains server provisioning and workload distribution.Let's take a look at the benefits of load balancing in an overhyped cloud environment:shut down, then load balancer up and move your application to another node. It allows server maintenance without service interruption.

Scalability: We all know cloud computing is more scalable. An increase in traffic can disrupt equipment and affect performance if not handled properly. Load balancing allows you to provision one or more virtual servers on demand without service interruption. Whenever there is high traffic on a website, a brand new server comes online and so the load balancer adapts effectively to this new server without affecting the performance of theapplication. This whole process usuallyrequires some downtime.

Redundancy: When you use load balancing to support an application, it comes with built-in redundancy. this shows that if your server goes down, the load balancer will redirect all traffic to another server without affecting the users.

Improved security: Load balancing barelyhandles the traffic, but it also helps to neutralize any DDOS attacks. Whenever there is malicious activity on the application, theload balancer not only detects it but also switches to another server to neutralize any attacks.

V. EXAMPLES OF LOAD BALANCERS

Direct Request Method: This method of sending a request is similar to the method implemented by IBM NetDispatcher. Physical servers and load balancers share a common virtual IP address. The load balancer has an interface built with a virtual IP address that accepts request packets and forwards the packets directly to the selected host.

A Load Balancer Server Based On Dispatcher: Administrators configure where TCP/IP requests are sent by performing intelligent load balancing using server availability, workload, capacity, and other user-defined parameters. The load balancer dispatcher can distribute HTTP requests to other nodes in the cluster. The dispatcher distributes the load among multiple servers ina cluster so that services from different nodes act as a virtual service on a single IP address; Consumers connect to each other as if it werea single server, with no knowledge of the back-end infrastructure.

Linux Weight Balancer: This is an open-source advanced load balancing solution used to build highly scalable and highly available network services such as HTTP, POP3, FTP, SMTP, methods convenience and caching, and Voice over Internet Protocol (VoIP) succeeded. It is a simple and powerful product designed for load balancing and failover. The load balancer itself is the main entry point of the server cluster system. It can run Internet Protocol Virtual Servers (IPVS), which implements transport-layer load balancing in the Linux kernel, also known as layer 4 switching.

VI. LOAD BALANCING ALGORITHMS

Load balancing is an important and challenging topic in cloud computing. Cloud computing load balancing helps efficiently useresources, reduces response time, distributesthe load more evenly, and reduces power consumption. A service to achieve full resource utilization. Several algorithms have been proposed for load balancing in cloud computing few of them have been mentioned below:

A. Round Robin Algorithm

Round robin is the simplest algorithm that works in a complete way. Using this type of algorithm rule, computer hardware allocates a quantum or time slice of time to perform a task on each node. When a VM is assigned atask, the VM moves to the bottom of the list. Round-robin offers higher performance than FCFS. If the time slice is too large, the round-robin behaves like FCFS, and if the time slice is too short, there will be more context changes in the round-robin algorithm. At any given time, some nodes may remain idle, and others may be overloaded.

B. Weighted Round-Robin

A modified version of Round Robin. Tasks are assigned according to the capacity of the VM, and the higher the VM, the more tasks you get. You can assign weights to the server, which are integer values that represent the processing power of the VM.

C. Dynamic Round-Robin

This algorithm works to reduce power consumption in the following steps:

If the virtual machine finishes running and there are more VMs on the physical machine, the physical machine will not accept any more VMs. Such physical machines are said to be in a "sleep" state, i.e. When the remaining virtual machines finish running, their physical machines shut down.

If a physical machine is idle for an extended period, you can migrate to another physical machine instead of waiting for all running virtual machines. This wait time threshold iscalled the "quiet threshold".

D. Throttled Load Balancer

The Throttled Load Balancer (TLB) maintains the state of the VM in a table of records. In this case, each virtual machine can be busy/idle. As soon as the request arrives, it searches the table and accepts the request if it finds a match in terms of machine size and availability. Otherwise, therequest is returned, and the request is also queued. The current utilization of the VM is not considered in the allocation.

E. Central Load Balancer

The algorithm's central load balancerdistributes the load between virtual machines with different hardware. A central load balancer maintains a table of VM IDs and states (busy/inactive). This is essentially an updated version of the throttling algorithm program. Maintains a table of each VM's status and its priority, similar to the throttling VMs. Priority is calculated by taking into account processor speed and memory capacity. Therefore, the VM allocation policy is similar to throttled,except that the VM with the highest priority gets the primary preference during thisalgorithm program. If so, the next highest priority VM is checked and the procedure continues until one is found or the entiretable is searched. However, this burdens the algorithm program with efficiency tuning in highly non-uniform settings. The algorithm is hampered by the possibility of all requestsreturning to the central load balancer. Also, the algorithm program is based on VM priority. This is calculated in a very static process and is not updated during contract awards.

F. Active Monitoring Load Balancing (AMLB)

AMLB maintains a table of information about each VM and the various requests currently assigned to that VM. When a request arrives, it looks for the least loaded virtual machine. When you receive an invitation to assign a VM replacement, the least stressed VM is identified. In that case, the known primary page is chosen. The load balancer returns the VMID to InfoCenter Controller. It forwards requests to the VM known by that ID and notifies the active VMload balancer of the new quota across VM quotas. Only the current utilization of theVM matters, not the computing power of theVM. Therefore, some jobs may have high latency and violate QoS requirements.

G. VM Assigned Load Balancer

This algorithm is a modification of the Active Monitoring load balancing algorithm.When a new

VM request comes in, checkthe VM table for new VM requests. The VMis made available, assigned a VM ID, and returned to the data center. Shridhar G. Domanal et. Al., this algorithm uses all available VMs and can use all VMs perfectly and correctly. This is different from the previous algorithm where only a few VMs could have multiple requestscoming in and the rest could stay below that.However, the paper does not explicitlymention how this is done. The algorithm does not use a VM if it was already allocated in the last cycle. But there is no logic in this as it should be the lightest VM with intelligent processing speed. Therefore,many tasks are assigned to her. By looking for evenly loaded VMs, only if there are multiple VMs with evenly loaded loads compared to the previous VM, or multiple VMs with the lowest subsequent loads and the highest processing speeds, will the tasks be evenly distributed. can be dispersed. However, the algorithm only considers load,and if the VMs are evenly loaded, in the last iteration the task will be sent to one of the VMs, regardless of whether that VM wasused.

H. Load Balancing Min-Min

The Min-Min algorithm, which selects thetask with the shortest execution time, canalso be a simple, fast expression that canimprove performance. Min-Min minimizes execution schedules, schedules optimal tasks, and improves overall production margins. Therefore, small tasks should stop first, large tasks should remain in thewaiting phase, and eventually, machineutilization will decrease. Min-Min shows the minimum job completion time, notconsidering the previous load on the machine.

I. Load Balancing Improves Min-Min (LBIMM)

The first step starts with the min-minalgorithm rule. The second step is to select the task with the smallest size from the mostloaded resource and calculate the completion time of that task across all different resources. The minimum execution time of this task is then compared to the mined makespan. In the case of makespan, the task is reassigned to the resource that created it, and each resource's preparation time is also updated. This process repeats until other resources use resources that are more expensive than makespan to reduce the processing time of the smallest task.

This way, overcommitted resources are released and idle and unused resources are used frequently. Therefore, LBIMM provides schedules that improve loadbalancing and reduce execution time.However, employment priorities in the plan have not yet been addressed.

J. Max-Min Algorithm

In the max-min algorithm with the largest completion time of all the available tasks is selected and executed in the node (VM) which produces a minimum execution time for the selected task. The same procedure is repeated for the left-over task.

K. User Priority Awarded LoadBalance Improved Min-Min

User priority used by H. Chen et. al. areincluded with the LBIMM rule to develop PA-LBIMM. The algorithm works in two teams. All tasks are divided into two groups,G1 and G2. Group G1 is for high-priority tasks. Group G2 is for regular tasks. High-priority tasks run as the first mining algorithm. The task is then scheduled with normal priority. Finally, the load-balancing algorithm creates a make-span for the running task.

L. Opportunistic LoadBalancing (OLB)

Opportunistic load balancing (OLB) is a static load balancing algorithm. OLB keeps all nodes busy, so don't think about previousloads. However, OLB does not consider the execution time of the task on this node. This can slow down the processing of the task, increase the total completion time (makespan), and incur overhead as therequest may appear incomplete. Forward to the node to be released.

M. Honeybee Foraging

The honeybee is a distributed load-balancing algorithm inspired by nature. Achieves load balancing across local servers. Foragingbees look for food and advertise the quality of the nectar and the distance of the food source from the nest by wobbling when found. The harvester’s bees then chase after the forager’s bees to the feeding area and harvest them.

Honeybee Foraging divides the server into virtual servers. Virtual servers have their own queues for managing server requestsand calculating processing time.

- Calculate the profit for a particularrequest. The gain can be adjusted as needed. In general, processing requests consumes latency and CPU time.
- It makes use of response time as a parameter.
- The server will only stay on if it is profitable and will continue to search to indicate if the state is loaded, overloaded, or underloaded.

N. Exponential Smoothing Based Weighted Least Connection

This algorithm program ESBWLC makes the choices supported by the node's electronics experience, performance, memory, various connections, and therefore the amount of memory currently in use. Recent knowledge has a greater impact on prophetic value than long-standing knowledge, as it uses all historical knowledge and distinguishes it by smoothing it. ESBWLC then predicts that the node should be selected with a carefully selected and assisted exponential smoothing method.

O. Weighted Active Monitoring Load Balancing

Jasmine James set. al predicted thistechnique. This could be a combination of a weighted round-robin and a load-leveling algorithm program with active monitoring.

VMs are assigned different weights based on the available processing power of the VM. The next VM assignment is done by choosing from the smallest loaded VMs and assigning the task to the best performance according to its weight.

This method eliminates the shortcomings of the ActiveMonitoring Load Balancing algorithm program by taking into account not only the load but also the processing power of the available VMs.

VII. MACHINE LEARNING-BASED LOAD-BALANCING ALGORITHMS

Load balancing techniques are classified differently into stages with different functions. They play an important role in server resource usage. Load balancers are built around certain aspects of the cloud environment, such as server CPU and memory resources, servicelevel agreements (SLAs), network congestion prediction, quality of service, etc. service, etc. service (QoS), estimated service latency, and cloud storage requirements.

Machine learning can be a part of AI that focuses on training systems to perform new tasks without being explicitly programmed. Historical data and statistical techniques are combined through a process known as trainingto create models that will predict new unseen values. Deep learning can be a subset of machine learning that uses variations of neuralnetworks with deeper networks and huge data sets. Deep learning combines featurediscovery and prediction in a deep network within hidden layers. It achieves better performance than traditional machine learning problems. The following smart models have been tested:

A. Deep Learning-based Regression Techniques:

Deep learning-based regression is used to predict the continuoustime of timed tasks and compute values. by Kaur et para. The deep learning network is designed to have 3 hidden layers consisting of a complex neural network, a composite layer, and thus an activation layer generated from the ReLU function. Training data includes time and value parameter data from the larger workflow.

B. Fully Connected Network (FCN):

A deep learning-based load balancing mechanism built from fully connected aggregation layers. The model was developed by Zhu et alto modify hash functions commonly used for task scheduling. Historical cluster access data is used to train the model. The FCN model is designed as a hierarchical model consisting of submodules that provide their output as input to the next hierarchical step. The hierarchy is the product of the 4 steps of input, scatter, map, and join. Eachsubmodule has 3 fully connected layers. The models used a deterministic approach to map workloads to servers.

C. Support Vector Machine (SVM) And K-Suggest Techniques:

Lilhore et al. proposed a load-balancing solution that supports multiple machine learning algorithmssuch as the SVM clustering engine and K-suggest. Clustering is used to identify groups of virtual machines that are derived from CPU and main

memory (RAM) usage. This system has assets shared with different groups and thus other virtual machines. It thenuses dynamic support mapping to specify the appropriate hundreds of virtual machine groups based on their size, i.e.: regular, idle, underload, and overloaded VMs. Resource mapping involves mapping grouped tasks to an accepted pool of virtual machines. This approach has improved service levels and reduced wait or decline times.

D. Bayesian Networks with Reinforcement Learning:

Liang et al. proposed a load balancer to manage traffic in the software-defined network controller component of data centers. Bayesian networksare used to predict the amount of traffic to load and combine with reinforcement learning for optimal reasoning for action and for a self-correcting parameter. The software-defined network is the brain of the network, separating the communication layer from the control layer. Bayesian network predicted the load traffic on the SDN controller while the predictions were used by reinforcement learning to determine the simplest causal action. The strategies adopted involve distributed network load and device handling and control. Improve network stability, load balancing speed, and controller performance.

E. Regression-Based Engineering, Random Forest and AdaBoost:

A machine learning-based load distributionmodel that includes many models, namely Multiple Regression (MLR), Random Forest (RF) and AdaBoost (Ada) used to determine the location of each processing request. Both CPU and GPU uptime are supported. This approach addresses architectural heterogeneity by taking into account differences in processing units and their associated operational characteristics. Its main purpose is to distribute transactions from distributedmanagement systems.

F. ANN and Self-adaptive Differential Evolution (SaDE):

This method was developed by Kumar et al. topredict workloads in the cloud data center. This approach combines alternative neural networks (ANN) and self-adapting differential evolution (SaDE). User requests are grouped into time units used due to historical data. The ANN part is trained with specific workloads as well as historical data. The resulting model is used to predict the upcoming addition of the information center. The model was trained on datasets from NASA and Saskatchewan servers long Memory Repetitive Neural Network Approach (LSTM-RNN) LSTM can be a particularly reasonable RNN to preserve the weights of past operations, generating a suitable deep learning algorithm to be used in time series forecasting. The LSTM algorithm has a built-in forget gate that allows it to leak long-term dependencies up to a point, and it retains only the necessary functionality. In their paper, Kumar and arrangements deal with the main problems of load balancing, namely: power consumption and dynamicresource scaling.

G. LSTM-RNN Load Balancing Technique:

It is developed by analyzing the history of the information center through cloud traffic logs taking into account the time factor. Information from historical data is usedto predict long-term workloads. Continuous training data over time. Expected workload data is used to stretch resources and discard unused resources to save energy. The LSTM-RNN model is trained on the HTTPtraces of the NASA dataset, the Saskatchewan server, and the Calgary server.

H. Back-Propagation Artificial Neural Network (BPANN):

The approach was used on an agent-based dynamic load balancer proposed by Prakash and Lakshmi over a software-defined network (SDN). SDN can be a globally visible component of a cloud architecture. As part of the offloading process,they are responsible for moving virtual machines into the Information Center. The BPANN algorithm has been trained on moving and loading virtual machine data. The resulting model is used to predict the load of the virtual machine. The projected load is thenused to determine the migration of the virtual machine. Efficient virtual machine migration improves network efficiency, as well as information migration speed and processing speed, which are significantly reduced because the heavy load suitable for the virtual machine has not been fully utilized.

I. Quantum Neural Network (QNN):

Approach QNN can be a variant of the neural network powered by the principles of quantum computing.

Quantum circuits work likeartificial neural networks. QNNs have some computational advantages as they include only necessary/required parameters to fit specificdata. This feature makes them more efficient than their conventional counterparts. The QNN model is used to predict the workload that will be generated by cloudlets. Singh et al encoded the workload data into qubits, and the model was used to very accurately estimate the workload and resources required. Their model used an uncontrolled gate (CNOT) because the activation function in the hidden layer and the output layer adjusted the weightsof the qubit lattice.

Further network weight optimization is performed using a self-balancing differential algorithm. Smart load balancing corresponds to conventional load balancing methods. uses machine learning and deep learning algorithms to develop load-balancing models that improve time intervals, resource elasticity,and energy savings. for example, Google has applied neural networks in its data centers to manage central cooling, reducing devices used for cooling by 40%. This shows the potential of computers to solve complex problems.

VIII. PERFORMANCE METRICS OF LOAD BALANCING TECHNIQUES

The performance parameters of the LB can be measured through a number of its quantifiable characteristics. Metrics can help determine thebest approach to load balancing. Some measurable attributes are directly measured while others depend on the variables involved.The following metrics have proven effective in evaluating the load distribution component:

Throughput: Throughput describes the measurement of the number of tasks/elements that go through a process in each time period. In load, load balancing throughput can be thought of as the number of operations the LB component can handle in a particular time period. For example, the LB component has high throughput if it responds to queries because it will process more than one task with delayed responses. How the LB component forwards requests and the time it takes to decide which cluster to assign workloads will affect throughput. The number of tasks completed in a specified time period also measures throughput.

Travel Time: Transfer time is the time it takes for the LB component to move processesfrom overloaded equipment to underutilized equipment. In load balancing, migration timeis measured by the time it takes to move a virtual machine from one physical machine to another. Migration is initiated when a task requires execution through multiple virtual machines or when a task is interrupted. Higher numbers of migrations lead to longer migration times. An efficient load-sharing technique minimizes virtual machine migration.

Response Time: This metric measures the time taken by the LB module to respond to a cloud task/request. The response time is calculated by adding the transmission time,wait time and service time. A good LB technique maintains a very minimal response time because performance is inversely proportional to response time. This metric is very common.

Fault Tolerance: Fault tolerance describes the ability of a system to withstand failures. In load balancing, it provides the ability to perform uninterrupted service evenif some of its parts fail. An efficient load balancer continues to operate even if someservers, VMs, and PMs fail. Logical error resolution ensures fault tolerance. This performance parameter is measured by having one or more points of failure. Redundant LB components are recommended to ensure that the LB component does not fail. For example; address fault tolerance by decentralizing control over the network.

Power Consumption: This metric determines the amount of power/power consumed by the virtual machines after performing load balancing.

IX. ADVANTAGES AND DISADVANTAGES OF LB ALGORITHMS

Table 1 shows a brief description of the algorithms along with their advantages and disadvantages.

Table 2 is a summary table showing the reviewed intelligent load balancing techniques.

Table 1: Advantages and disadvantages of the load balancing algorithm

S. No.	Algorithm	Description	Advantage	Disadvantage
1.	Round Robin	The request is allocated for a fixed period	Equal distribution of workload	The process is not known in advance. For a larger task context switching increases.
2.	Weighted Round Robin	According to the processing capacity of VM weight is assigned	Optimal resource utilization	Processing time is not taken into consideration.
3.	Dynamic Round Robin	It maintains VM retiring state and VM threshold state.	Cost of power consumption gets reduced	Does not scale up for large data center
4.	Throttled LB Algorithm	Maintain a state of VM busy or idle	Evenly distribution of load	Does not consider the current state of the VM
5.	Central Load Balancer	Maintains a list of all available VM and their state	Load is balanced centrally	Fixed priority
6.	Active Monitoring Load Balancing	The least loaded VM is allocated with the request	The existing load is taken into consideration	VM processing power is not considered
7.	VM Assigned LB Algorithm	VM is allocated as and when available	Proper VM utilization	NA
8.	Weighted Active Monitoring LB Algorithm	Weights are assigned to the VM according to their processing power	Consider the weight and processing power of the VM	Complexity increases
9.	Min-Min Algorithm	The select task with the least execution time	Simple to execute	Does not consider existing load
10.	Load Balancing Improved Min-Min LBIMM	Similar to Min- Min algorithm. From the all available task, the task with the smallest completion time from the most heavily-loaded resource is calculated	Overall completion time is reduced	Does not consider priority
11.	Max Min Algorithm	A job with a higher execution time is executed first.	Shorter makespan as compared to Min-Min	Shorter jobs have to wait
12.	User Priority Awarded Improved Min-Min	Divides the task into two groups according to user priority VIP and ordinary.	Consider priority and makespan	No deadline
13.	Opportunistic Load Balancing (OLB)	Uses static load balancing algorithm attempt to allocate selected job available VM	Keeps all available VM busy	Does take into account the previous load
14.	Honey Bee Foraging	Distributed load balancing for self-organization	Well suited for heterogeneous environment	Well suited for heterogeneous environment

15.	Weighted Least Connection	Assign the task to the node having the least number of connection	Balances load efficiently	Processing speed is not considered
16.	Exponential Smoothing Forecast based WLC (ESBWLC)	The task is assigned according to the processing power and memory of the node	Each is examined	Complex calculations

Table 2: Summary table showing reviewed intelligent load balancing techniques

S. No.	Publication	Underlying Machine/Deep Learning Model	Data Used	LB Problem Addressed
1.	Deep Learning Regression	CNN	Tasks workflow data	Quality of Service (QoS) resource utilization and throughput
2.	Deep Learning-Based Load Balancer	Hierarchical sub-models of FCN	Historical cluster access logs	Solves data skew problem in classical LB
3.	Lilhore machine Learning-based LB	SVM, K-Means Clustering	RAM & CPU usage data	VMs Resource utilization & execution time reduction
4.	Reinforcement based SDN controller	Bayesian Network & Reinforcement Learning	Network traffic data	SDN Controller LB, Network Stability, Security
5.	Distribute d database query load distribution	multiple linear regression (MLR), random forest (RF), and AdaBoost (Ada)	Database queries data	Cloud Heterogeneity of CPU & GPU
6.	Workload prediction	Artificial Neural Network and self-adaptive differential evolution (SaDE)	Client requests amassed to time units	Distribution of workloads
7.	Temporal aware LB	LSTMRNN	Cloud workload with a time factor	Resource elasticity & power saving
8.	Dynamic agent LB	Backpropagation Artificial Neural Network BPANN	Network Traffic Logs	VM migration, data migration
9.	Quantum based LB	Evolutional Quantum Neural Network EQNN	Cloudlets workload logs	Dynamic resource scaling

X. LOAD-BALANCING ENVIRONMENTS FOR DIFFERENT ALGORITHMS

A load balancer can be a hardware or software device that efficiently distributes traffic across healthy servers to prevent any server from being overloaded. There are two basic approaches to load balancing: static load balancing and dynamic load balancing.

A. Difference between static and dynamic load balancing:

Static load balancing methods distribute traffic without adapting to the current state of the system or server. Some static algorithms send an equal amount of traffic, either in an explicitly specified order or in all directions, to each server in a group. Dynamic load balancing algorithms look at the current state of the system and each server and base traffic distribution across these factors.

A static load-balancing algorithm does not take into account the state of the system because it distributes the tasks. Instead, the distribution is formed by assumptions and knowledge of the general system

made before the classification begins. This includes known things like CPU count, communication speed, and power, as well as assumptions like resource requests, response times, and arrival times for incoming tasks. Static load balancing algorithms in distributed systems minimize the operation of functions by matching a known set of tasks to available processors. These types of load-balancing strategies typically focus on a router that optimizes operational functionality and load distribution. The good thing about static load balancing in distributed systems is its ease of use and quick and simple implementation, although there are some situations that don't seem to be better served by this type of algorithm.

Dynamic algorithms take this load into account for each node or compute unit in the system, achieving faster processing by moving tasks from overloaded nodes to overloaded nodes. Dynamic algorithms are much morecomplicated to style, but especially when the execution times for different tasks vary widely, they produce superior results. Also,because there is no need to dedicate specific nodes to distributing digits, dynamicload-balancing architectures are often more modular.

Both dynamic and static load balancing techniques are shaped by other factors as well:

Task type: The task type has a large impact onthe efficiency of the load balancing algorithm, so maximizing access to task information when the algorithm's advanced cognitive processes are taking place offers optimization potential increase.

Task Size: It is very rare to know exactly how long a task will run, but it allows for optimal load balancing. There are several ways to estimate various execution times. For similarly sized tasks, the average execution time can be used very well. However, if execution times are highly irregular, more advanced techniques are needed. For example,tasks can be tagged with metadata, andinferences about future tasks can be drawn based on statistics based on previous execution times of similar metadata.

Dependencies: Tasks can be dependent on each other and some tasks cannot start until other tasks have been completed. We show such dependencies in a directed acyclic graph and optimize the order of tasks to minimize overall execution time. Some algorithms can use metaheuristic techniques to compute the optimal task distribution.

Task Separation: This refers to the power of tasks to be undermined into subtasks during execution. This specificity is important for the design of load-balancing algorithms.

Hardware architecture between parallel units.

Heterogeneity: Units of different computing power often contain parallel computing infrastructure, and load balancing must takethis variation into account. For example, less powerful units should receive requests that require less processing power than larger units, or requests of the same kind or unknownsize.

Storage: Parallel devices often fall into thecategories of shared storage and distributedstorage. The shared memory unit follows the PRAM model, where all sharing, reading, and writing are done in parallel on shared memory.Distributed storage units follow the distributed storage model, each unit exchanges information via messages and has its own storage.

Both types have their advantages, but few systems fall perfectly into either category. In general, load-balancing algorithms must be specially adapted to parallel architectures so as not to reduce the efficiency of parallel problem-solving.

Hierarchy: The two main forms of load-balancing algorithms are controller agents and distributed control. Within the controller-agent model, the controller assigns tasks to agents that execute the tasks and notify the controller of progress. Controllers can assign and reassign tasks only for dynamicalgorithms. When control is distributed across nodes, the nodes share responsibility for assigning tasks, and load-balancing algorithms are also run on each node. Additionally, an intermediate strategy is possible where he putsall the control nodes of the sub-clusters under one global control. In fact, various multi-level strategies and orchestrations possible usingelements of both distributed control and control agent strategies.

Scalability: Computer architectures evolve, but it's better to avoid designing replacement algorithms every time the system changes. Therefore, the scalability of algorithms, or the ability to adapt to

scalable hardware architectures, can be a critical parameter. An algorithm is scalable with respect to its input parameters if the scaling of the parameters, i.e. the performance of the algorithm remains relatively independent.

Fault Tolerance: Failure of one component during execution does not cause the entire parallel algorithm to fail, especially on large computational clusters. Fault-tolerant algorithms identify problems while allowing recovery.

11. DISCUSSION AND CONCLUSION

Load balancing in cloud computing is of great importance. Load balancing improves system performance. This article summarizes the different cloud computing algorithms and theirstrengths and weaknesses. The common load balancing techniques described here mainly focus on reducing the time involved, improving throughput, reducing production time, and improving performance byconsidering other factors such as processor,memory, and disk. The main goal of this article is to investigate the latest trends in cloud load-balancing research by finding the most used machine learning algorithms in load-balancing components. Traditional deep learning and machine learning models haveproven to be an adaptation in the big data age. Traditional machine learning algorithms were found to include multiple linear regression (MLR) and random forest classifier (RF); SVM clustering and K-Means. thanks to the complexity of load balancing and hence the huge data associated with their training, such as CPU logs, network traffic data, and storage logs.

Traditional machine learning algorithms are being replaced by deep learning models. Deep learning models implemented in this area include BPANN, CNN, FCN, ANN, and LSTMRNN. The deep learning model shows better performance in terms of prediction accuracy. These models handle big data well without affecting the standardization of the model. These models represent an important trend from spatially oriented models such as ANN and CNN to space-time models such as LSTM and CNN-LSTM. This trend suggests that score is an important factor to consider during load balancing. Other deep learning models that stand out from the rest include a load distribution component based on deep reinforcement learning and, therefore, a load balancer based on a quantum neural network.

Table 3: Load Balancing Environments for Different Algorithms

Algorithm	Static Load Balancing	Dynamic Load Balancing	Centralized Load Balancing	Distributed Load Balancing
Round-Robin	True	False	True	False
Min-Min	True	False	True	False
Max-Min	True	False	True	False
CLB	True	False	True	False
LBMM	False	True	False	True
Active Clustering	False	True	False	True
OLB	True	False	True	False
PA-LBIMM	True	False	True	False
WLC	False	False	True	False
ESWLC	False	False	True	False
Honey Bee Foraging	False	False	False	True

REFERENCES

[1] Juliet Gathoni Muchori, Peter Maina Mwangi, "Machine Learning Load Balancing Techniques in Cloud Computing: A Review", International Journal of Computer Applications Technology and Research Volume 11–Issue 06, 179-186, 2022, ISSN:-2319–8656 DOI:10.7753/IJCATR1106.1002

[2] Harish Sharma, Pradeep Semwal, SGRR University, Dehradun, India, "A REVIEW OF LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING", 2021 IJCRT, Volume 9, Issue 3 March 2021, ISSN: 2320-2882

[3] Boden, M. A. (2004). The Creative Mind: Myths and Mechanisms. London: Routledge. Bringsjord, S., Bello, P., & Ferrucci, D. (2001). Creativity, the Turing Test, and the (better) Lovelace Test. Minds and Machines, 11, 3-27.

[4] Menabrea, L. F. (1843). Lovelace, A., trans. Sketch of the Analytical Engine invented by Charles Babbage. Scientific Memoirs 3.

[5] Rehling, J., & Hofstadter, D. R. (2004). Letter Spirit: A Model of Visual Creativity. Sixth International Conference on Cognitive Modeling (pp. 249-254). Mahwah, NJ: Lawrence Erlbaum.

[6] Retrieved from http://arxiv.org/abs/1410.6142

[7] Wang, Z., Yang, J., Jin, H., Shechtman, E., Agarwala, A., Brandt, J., et al. (2015). DeepFont: Identify Your Font from An Image. MM '15 Proceedings of the 23rd ACM international conference on Multimedia Pages, (pp. 451-459). Brisbane.

[8] (2016, November). Retrieved June 2017, from https://www.indiehackers.com/businesses/logo joy

[9] Retrieved July 2022, from https://avinetworks.com/glossary/static-load-balancing/#:~:text=Static%20load%20balancing%20algorithms%20in%20distributed%20systems%20minimize%20specific%20performance,performance%20function%20and%20distributes%20loads.

[10] IBM Cloud Education, "What is Machine Learning?," IBM Cloud, 15 July 2020. [Online]. Available: https://www.ibm.com/cloud/learn/machinelearning. [Accessed 17 February 2022].

[11] IBM Cloud Education, "What is deep learning? " IBM Cloud, 1 May 2020. [Online]. Available: https://www.ibm.com/cloud/learn/deeplearning. [Accessed 17 February 2022].

[12] Colton, S., Charnley, J., & Pease, A. (2011). Computational Creativity Theory: The FACE and IDEA Descriptive Models. Proceedings of the Second International Conference on Computational Creativity, (pp. 90-95). Mexico City.

[13] Higgins, M., & J., M. (2000). The Role of Creativity in Planning: The 'Creative Practitioner'. Planning Practice & Research, 117-127.

[14] Jefferson, G. (1949). The Mind of Mechanical Man. British Medical Journal, 1105-1110.

[15] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521, 436-444. Luan, F., Paris, S. S., & Bala, K. (2017). Deep Photo Style Transfer.

[16] Luger, G. F. (2009). Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Boston: Pearson. Kaur, B. Kaur, P. Singh, M. S. Devgan, and H.K. Toor, "Load Balancing Optimization Based on Deep Learning Approach in Cloud Environment," I.J. Information Technology and Computer Science, vol. 3, no. I, pp. 8-18, 2020.

[17] Zhu, Q. Zhang, T. Cheng, L. Liu, WeiZhou and J. He, "DLB: Deep Learning Based Load Balancing," CoRR, vol. 1910, no. 08494V4, 2021.

[18] U. K. Lilhore, S. Simaiya, K. Guleria, and D. Prasad, "An Efficient Load Balancing Method by Using Machine Learning-Based VM Distribution and Dynamic Resource Mapping," Journal of Computational and Theoretical Nanoscience, vol. 17, no. 7, pp. 2545-2551, 2020.

[19] S. Liang, W. Jiang, F. Zhao, and F. Zhao, "Load Balancing Algorithm of Controller Based on SDN Architecture Under Machine Learning," Journal

of Systems Science and Information, vol. 8, no. 6, pp. 578-588, 2021.

[20] Abdennebi, A. Elakas, F. Taşyaran, E. Öztürk, K. Kaya and S. Yıldırım, "Machine learning - based load distribution and balancing in heterogeneous database management systems," Concurrency and Computation, vol. 34, no. 4, 2021.

[21] J. Kumar and A. K. Singh, "Workload prediction in the cloud using artificial neural network and adaptive differential evolution," Future Generation Computer Systems, vol. 81, no. C, pp. 41-52, 2019.

[22] J. Kumar, R. Goomer and A. K. Singh, "Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters," Procedia Computer Science, vol. 125, pp. 676-682, 2018.

[23] S. WilsonPrakash and P. Deepalakshmi, "Artificial Neural Network Based Load Balancing On Software Defined Networking," in IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 2019.

[24] Abbas, D. Sutter, and S. Wörner, "The power of quantum neural networks," IBM, 2 July 2021. [Online]. Available: https://research.ibm.com/blog/quantumneural-network-power. [Accessed 16 Feb 2022].

[25] K. Singh, D. Saxena, J. Kumar and V. Gupta, "A Quantum Approach Towards the Adaptive Prediction of Cloud Workloads," IEEE Transactions on Parallel and Distributed Systems, vol. 32, pp. 2893-2905, 2021