LOAD BALANCING ALGORITHMS FOR CLOUD COMPUTING ENVIRONMENT

[1]Ankita Jaiswal, [2] Dr. Aisha Fernandes

[1]PG student, [2]Assistant Professor

[1, 2] Department of Information Technology

[1, 2] Goa College of Engineering, Farmagudi, Goa, India.

Abstract: The Internet of things (IoT) is a new emerging technology from the last few years and growing day by day. IoT devices are the physical devices that are embedded with sensors, softwares, computing devices and electronics with the capabilities of sending and receiving data. IoT devices collect the data and send it to the cloud to get useful information. Cloud computing provides services like storage, infrastructure and softwares in IoT applications. As the number of customers increases day by day, the data generated by IoT applications also increases, which creates load on the cloud server. For efficient use of the cloud resources, it is very important to balance load on servers using different types of load balancing algorithms. This paper presents a detailed study on the different load balancing algorithms for cloud computing proposed in the literature.

***IndexTerms* - Load Balancing, Cloud Computing, Internet of Things, simulation tools.**

I. INTRODUCTION

The Internet of Things (IoT) is a form of distributed computing environment which helps in sharpening our future world in a better way. The key concept of IoT is device-to-device communication, which helps the physical object to communicate with each other over a network. Physical objects can be anything such as vehicles, animals, electronic objects, humans or any object having a unique Identifier which helps them to connect to the network. These objects are embedded with the small sensors which collect the data from the surrounding and share with the other devices to analyze data. Based on the analysis of the data, these devices make independent decisions, which makes them 'smart' objects [1]. It is expected that the number objects connected to the internet will be much more than the number of human beings and the major part of the data traffic will be produced by these objects [2].

IoT is considered to be a big step towards technology and improving our lifestyle in a better way. It provides us to live a luxurious life, save our time by making decisions and gives useful information. Some of the IoT applications are already available in the market. For example, a smart home system provides many facilities to the homeowner such as security, comfort, convenience and energy efficiency. The homeowner can give instructions to their IoT appliances from anywhere at any time, such as through the mobile application, the owner can control the temperature and lighting inside the house, monitor the home security by controlling the entrance gate and allow known persons to enter the house. IoT can be applied in Healthcare to monitor the health of independent living elderly people and to provide constant medical supervision for chronic patients living in isolated or underserved locations. There have been several research groups working towards development of e-Agriculture application, to boost agricultural production.

1.1Cloud Computing

Cloud computing is the backbone of IoT, through which all the data collected by the IoT devices are analysed. The Cloud could be described as a system of interconnected, distributed, and parallel servers hosted by numerous data centres which are located in different geographical regions. Cloud computing services are divided into three levels: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

Infrastructure as a Service (IaaS): - IaaS provides the on demand provisioning of servers which runs multiple operating systems and a customized software stack. It is considered as the bottom most layer of cloud computing architecture. IaaS services are mainly provided by the web services [7][8]. Amazon Web Services (AWS) IoT provides the IaaS services for the IoT devices. It provides broad and deep functionality which helps to build IoT solutions. Some of the AWS IoT services provided are IoT Core which helps to connect devices easily and securely to the cloud server, IoT Cloud Management it register, organize, monitor, and remotely manage IoT devices , IoT Analytics collects, pre-processes, enriches, stores, and analyzes IoT device, IoT Greengrass is software that lets you run local compute, messaging, management, sync, and ML inference capabilities on connected devices in a secure way, etc [9].

Platform as a service(PaaS): - PaaS is the middle layer of cloud computing architecture. PaaS offers the environment to the developers to create and deploy applications. Here developers do not need to know how many processors and how much storage that application will be using. Microsoft Azure provides the platform as a service [10]. The Azure Internet of Things (IoT) is a collection of Microsoft-managed cloud services like connect, monitor, and control billions of IoT devices. Platform services provide the building blocks for customized and flexible IoT applications [11].

Software as a service (SaaS): - SaaS is the top most layer of cloud computing. Any user can access these services through a web portal. Therefore, consumers are now starting to use more online software services than the installed services. Where online software also provides us the same functionality as the installed ones. Word processing and spreadsheet are the traditional desktop applications which can now be accessed as a service on the web [12][13]. Oracle provides SaaS applications for IoT devices for production monitoring, Asset monitoring, fleet monitoring and many more [14].

The main characteristic of the Cloud model is remote and on-demand access to computing resources based on service level agreement. The second key aspect is elasticity [15]. Elasticity is the property of cloud computing where the user can change the configuration as per need. Cloud provides services where we have to pay the service provider as we use the resources. It has the distributed architecture where the servers are located geographically. Only a few servers are used to handle the load in all over the world. There are various types of loads in the cloud network such as memory load, Computation (CPU) load, network load, etc [16]. As the IoT devices are increasing the data collected by the devices are also becoming huge in number, so it is very important that all the servers should be balanced, no server should be overloaded nor under loaded. To balance the load between the servers load balancing algorithms are used. These algorithms help to distribute the load in servers as their capabilities so that all the resources are fully used.

The rest of the paper is organized as follows. Section 2 presents the Load balancing in cloud computing and the related work done in section 3. Simulation tools which could be used to simulate the IoT environment are described in Section 4.

II. LOAD BALANCING IN CLOUD COMPUTING

In cloud computing, cloud servers should always be balanced, to use the resources with their full capacity. Sometimes it happens that some servers are heavily loaded while the other servers are under loaded or in idle state. To overcome this problem load balancing algorithms are used. These algorithms help in allocating every single task by monitoring load on each server. According to the Deepak Mahapatra [16] the balancing algorithm is defined as "The load balancing in clouds may be among physical hosts or VMs. This balancing mechanism distributes the dynamic workload evenly among all the nodes (hosts or VMs). The load balancing in the cloud is also referred to as load balancing as a service (LBaaS)".

There are different types of load balancing algorithms which are used for cloud computing; they are categorised in two categories namely static load balancing and dynamic load balancing.

Static load balancing algorithms allocate tasks to the servers before the compilation where all the requirements of the resources are known to the algorithm. The allocation of tasks are based on those requirements. Static Load Balanced Algorithm is suitable for small distributed environments with high internet speed and ignorable communication delays It works properly when the systems or nodes have the ignorable differences in the load, therefore the algorithms which come under the static are usually not suitable for cloud computing. Because in the cloud we have n number of users due to which load highly varies [17].

Dynamic algorithms work in the real time situation, where it takes continuous information about the load on the server. With respect to that it takes the decision of distributing the tasks amongst the servers. Accordingly we can allocate, reallocate or remove any task from the server based on the priority. Dynamic Load Balanced Algorithm focuses on reducing communication delays and execution time for large distributed environments. These techniques or the algorithms are highly successful for load balancing the cloud environment on their nodes among different types of resources [17].

In the past few years there are many static and dynamic load balancing algorithms that have been proposed for the cloud computing environment. A detailed comparison is done in [18]. In this section some of the existing algorithms proposed by the researchers are discussed.

The static load balancing algorithm works in small distributed environments so they are less complex compared to dynamic which have a highly distributed environment. Advanced information is needed in static algorithms such as length and number of tasks. The scheduling decisions are taken at runtime by dynamic algorithms and compile time by static. Static algorithms are not good at balancing load properly at run time but monitor nodes continuously where dynamic balance loads efficiently and do the monitoring continuously by event base or time interval. Static algorithms take more time to solve but do not give the optimal solution for the complex computational problem, dynamic takes less time and gives useful solutions. Traditional types of algorithms come under the static ones and the meta-heuristic algorithms come under the dynamic algorithms.

2.1 Static load balancing algorithms

2.1.1 *Round Robin:* Round Robin is one of the traditional methods of assigning tasks to nodes. It uses the circular manner to assign tasks to the virtual machine. If there are *n* tasks to process then they will be assigned processes depending upon their arrival of request. The disadvantage of this method is some nodes can remain underloaded due to their high computing power then other nodes while other nodes which have low computing power are heavily overloaded [19].

2.1.2 *Min-min:* Min-min is a static load balancing algorithm where the arrangement of tasks is done in order of their length and completion time. They are arranged in increasing order of task with minimum completion time, those tasks are selected first and the next task is selected in that order only [20].

2.1.3 *Max-min:* This method is very similar to the min-min algorithm but here we arrange the tasks according to their maximum completion time. The task with maximum completion time is assigned first and other tasks wait till the execution of that task [20].

2.1.4 *Opportunistic Load Balancing (OLB):* It keeps all the servers busy and never considers the load of the task which is currently running on the server. Besides the current task running on the server it randomly allocated another task [21].

2.1.5 *Weighted Round Robin:* It was developed to overcome the drawback of round robin algorithm. Here it assigns weight to the servers according to their load handling capacity. with respect to the weight tasks are allocated. This algorithm helps to maintain the load on the server as well as their capacity. If two tasks are assigned to two different servers then the third task will be allocated to the higher weightage server [22].

2.1.6 *Least Connection:* In this algorithm it considers currently running tasks also which is not taken by the above mentioned algorithms. According to which other tasks are assigned, these lead to the least number of active sessions in current time [23].

2.1.7 *Weighted Least Connection:* This algorithm is similar to the least connection algorithm. As seen in Weighted Round Robin where weights are assigned to the server likewise in Weighted Least Connection also weights are assigned in numerical form [24].

2.2 Dynamic Load balancing

2.2.1 *Honey bee foraging:* This dynamic algorithm is derived from the honey bees; the detailed study is done on their behaviour of how they search for food and reap food. Scout is a class of bees which finds food and informs others through the dance called vibration. This gives the estimate about the quality and quantity of the food. This technique is used in load balancing to inform the underloaded and overloaded VMs. The tasks from the overloaded machines are shifted to the underloaded machine. Similarly, the tasks from the overloaded VMs are considered as the honey bees. Submitting tasks to the underloaded VM, the task will update the number of tasks and load of that particular VM to all other waiting tasks. Based on the load and the priority tasks choose VMs. Whenever a high priority task has to be submitted to other VMs, it should consider the VM that has a minimum number of high priority tasks so that the particular task will be executed at the earliest. Since all VMs will be sorted in ascending order based on load, the task removed will be submitted to under loaded VM. In essence, the tasks are the honey bees and the VMs are the food sources. Loading of a task to a VM is similar to a honey bee foraging a food source [25].

2.2.2 *Particle Swarm Optimization (PSO) Algorithm*: PSO was first introduced by Kennedy and Eberhart; it is one type of a meta-heuristics method. It is a self-adaptive global search based optimization technique. The PSO algorithm is similar to other population-based algorithms like GA but, there is no direct recombination of individuals of the population. PSO follows a model or a pattern for the social interaction and creates the communication between them, for example, fish schools and bird flock. PSO algorithm is very similar to the swarm intelligence optimization algorithms. It mainly focuses on reducing the total cost of computation of an application on the cloud computing environment.

2.2.3 *Heuristic Algorithm:* Traditional methods of solving problems were very slow and not compatible for solving NP-complete problems. Heuristic algorithms solve problems in a faster and efficient way than the traditional way and designed to solve decision problems. Example is the Traveling Salesmen Problem where there is a list of cities to visit and the distance between the two cities are given and the optimal solution is to visit all the cities by travelling a minimum distance.

2.2.4 *Ant colony optimization:* ACO is a technique of problem-solving inspired by the behaviour of ants in searching the optimal paths from the nest to food; they all work together and search new sources of food while some ants parallel works on shifting food from source to nest. Many researchers are inspired by this behaviour of ants and helped them to solve real life problems in different fields [26]. In this method, a pheromone table of a node is maintained and an ant updates the entries of the node from source to destination. The other routing ants reference the pheromone table and calls that have it as their destination. However, for asymmetric networks, the costs from to and from to may be different. Hence, In this approach for updating pheromone is only appropriate for routing in symmetric networks.

2.2.5 *K subset algorithm:* The original k-subset algorithm and its successors do not reflect the internal characteristics within the heterogeneous clustering system. These algorithms include two steps: 1) selecting k appropriate nodes and 2) choosing the least loaded one. Every node is selected at the same probability. If the nodes are ordered by current load status, the resulting request arrival rate at a node depends only on the node's rank in the sorted list, not on the magnitude of difference in the capabilities or the capacities between nodes.

2.2.6 *Genetic Algorithms (GA):* GA is one of the most used algorithms which solves the NP(Non-polynomial) -complete problems. It is derived from the soft computing method. It comes under the heuristic search process. GA is inspired by natural evolution from the human mind and genetics.

2.2.7 *Hill climbing:* A Hill climbing is the simple and iterative method based optimization algorithm; it moves towards the best solution step by step. It selects the increasing value and moves towards the peak or uphill. Peak is the top of the hill or the place where no neighboring value is more than the peak. The algorithm stops once it reaches the peak or to the stopping criteria. Sometimes it can reach a local optimum solution instead of global optimum. For the load balancing this algorithm maintains an index table with the list of VMs and their states (BUSY/AVAILABLE). When the new task request arrives randomly generates VM id and allocates the task to tha VM if the state is available otherwise generates another VM id randomly. It updates the index table accordingly. Once the task is completed, de-allocates the VM and updates the table.

III. Proposed Algorithms Review

In [27], authors have proposed a predictive load balancing algorithm in the cloud, it helps to maintain service quality and minimize energy consumption of cloud networks. Here they have compared two algorithms round robin and random algorithm with the proposed C-Rule algorithm, it is a rule based algorithm. This algorithm helps to allow cloud service providers to prepare their resources to allocate, the prediction is based on Cicada predictions. It helps in achieving faster and more reliable load balancing algorithms for cloud environments. The proposed algorithm reduces the CPU waiting time.

In [28] paper, author proposed Hybrid genetic algorithm and particle swarm optimization algorithm (HGAPSO) for the load balancing of molecular dynamics simulations. They have combined two algorithms Genetic Algorithm and Particle Swarm Optimization on heterogeneous supercomputers. Both the algorithms are nature inspired and used CUDA for the simulation. It gives less waiting time for more VMs.

In [29] paper, proposed The Heuristic-based load-balancing algorithm (HBLBA) is a load balancing and task scheduling algorithm for the IaaS cloud environment. The algorithm is divided into two steps. In the first step an efficient strategy is used to find the best feasible server configurations to decide the requirement and type of VMs. In the second step to minimize the waiting time mapping, a queuing model is used to schedule the tasks on VMs. The proposed algorithm is compared with the existing algorithms, round robin and min-max in terms of makespan, waiting time, and resource utilizations.

In [30] paper, proposed algorithm is an Improved Honey Bee Based Load Balancing for cloud computing. Honey bee algorithm has been improved to reduce the makespan and assign the resource to the task to improve the throughput in the cloud environment. The drawback of this algorithm is lack in Quality of Service.

In [31] paper, proposed an Improved Particle Swarm Optimization for load balancing in cloud computing it is praised highly nowadays. The proposed algorithm is compared with the red-black tree algorithm; it improves the efficiency and reduces the drawback algorithm. From the result it can be found that the speed of PSO and improved PSO is the same. It gives better results in terms of task solving and load balancing compared to the other algorithms.

In [32] paper, proposed Bat Algorithm based on the natural behaviour of bats. When bats chase its prey, it flies unpredictably by changing the velocity, and positions based on the distance between the prey and itself. In the proposed algorithm, the tasks are considered as artificial bats and virtual machines are considered as prey. They have improved the response time and imbalance degree. Drawback of this algorithm is it gives low migration time and scalability.

In [33] paper, proposed algorithm by combining improved max-min and ant colony algorithms is a hybrid approach based load balancing algorithm for clouds. Improved Max-Min is used to reduce the execution time and combine with Ant colony to find optimum load balancing. It minimizes the processing time and cost.

In [34] paper, proposed Genetic Algorithm and Gravitational Emulation Based Hybrid Load Balancing algorithm, the authors have combined two algorithms genetic algorithm with the gravitational emulation local search (GEL). GA has a global nature towards the problem space where GEL searches locally which helps to improve makespan and reduces the number of virtual machines who won't be much useful. The drawback of GEL algorithm is it never stops at 1 best possible solution.

In [35] paper, proposed dynamic algorithms for load balancing in a grid system. The parameter they have focused on is reducing the execution time of the algorithm. It performs better with a high amount of data. It gives a better result compared to existing once.

In [36] paper, proposed an Improved k-Subset static algorithm for load balancing in cloud computing. It is inspired by the k-subset algorithm and improves simplicity, stability and flexibility of the algorithm. The proposed algorithm minimizes the execution time compared to the original. On paper they have provided improved results using the CloudSim simulator.

In [37] paper, authors have proposed a load balancing algorithm based on Genetic algorithm (GA) for the cloud environment. The proposed algorithm is a dynamic algorithm. It tries to minimize the completion time of the task. GA is a soft computing approach and comes under heuristic search processes. They have tried to overcome the problem of the inappropriate distribution of the execution time, which is used to create the traffic on the server.

In [38] paper, proposed algorithm is Stochastic Hill Climbing algorithm for balancing the load in the cloud. This algorithm is based on the traditional hill climbing method where it selects the next best neighbouring value and moves towards uphill, until it reaches the peak or local optimum solution. It gives better results when compared with two commonly used algorithms First Come First Search and Round Robin.

In [39] paper, proposed algorithm for rebalancing the load on the VMs in the cloud computing environment. For the simulation process the Hadoop system is used. It is developed in a dynamic Hadoop Distributed File System (HDFS) environment to gain the best throughput.

In [40] paper, proposed Load Balancing Ant Colony Optimization (LBACO) algorithm for cloud computing. It minimizes the makespan and distributes the even load on all the virtual machines and also calculates the degree of imbalance. They have shown the improved result in the paper. The proposed algorithm is compared with the First Come First Search algorithm with 500 independent tasks using the CloudSim simulator.

In [41] paper, proposed Weighted Signature based Load Balancing (WSLB) algorithm for cloud computing. This algorithm finds the load assignment factor for each host in a datacenter and maps the VMs according to that factor. Virtual machine id is sent by the load balancer which is available on the highest configuration host having maximum load assignment factor then lowest one and so on. The proposed algorithm reduces the average response time.

TABLE 1. Comparison Table

S. No	Authors name	Algorithm Used	Type	Tools Used	Advantages	Limitation	Performance metrics
1	A Predictive Workload Balancing Algorithm in Cloud Services [27]	Round Robin and Random Algorithms	Dynamic	Cicada Toolkit CloudSim	It helps to achieve faster and more reliable workload balancing in cloud environments. They can utilize their resources more efficiently by preventing any over-provisioning.	Waiting time depends on the number of VM and number of tasks.	Waiting time
2	A Hybrid Particle Swarm Optimization Algorithm for Load Balancing of MDS on Heterogeneous Computing Systems [28]	Hybrid genetic algorithm and particle swarm optimization algorithm	Dynamic	CUDA	Comparing the algorithm with respect to various parameters such as waiting time, makespan, and scheduled length ration, CPU and VM utilization	Time consuming	Runtime of program and Makespan
3	Heuristic-based load-balancing algorithm for IaaS cloud [29]	Heuristic-based load-balancing algorithm	Dynamic	MATLAB	It minimizes the completion time of the tasks and due to which makespan minimizes.	It did not consider loads of the servers for the long term process.	Completion time and Makespan
4	A Novel Improved Honey Bee Based Load Balancing Technique in Cloud Computing Environment [30]	Improved Honey Bee Based Load Balancing	Dynamic	Cloudsim	The technique works on or reducing service makespan.	Lack in Quality of service.	Reliability, Accuracy, Makespan, Migration time
5	A Novel Load Balancing Algorithm Based on Improved Particle Swarm Optimization in Cloud Computing	Improved Particle Swarm Optimization	Dynamic	Cloudsim	Have a powerful ability of global search.	The efficiency of PSO algorithm is not satisfying and the performance in load balancing is not that good.	Runtime processing and Speed balance degree

	Environment [31]						
6	An Optimal Load Balancing Technique for Cloud Computing Environment using Bat Algorithm [32]	Bat Algorithm	Dynamic	MATLAB	Optimum and finest result is obtained.	Migration time and scalability are low.	Response time and Migration time
7	Dynamic Combination of Improved Max-Min and Ant Colony Algorithm for Load [33]	Improved Max-Min and Ant Colony Algorithm	Dynamic	CloudSim	It helps in enhancing the performance of distributed systems.	Time consuming	Processing time and Associated Cost
8	A Genetic Algorithm and Gravitational Emulation Based Hybrid Load Balancing Strategy In Cloud computing [34]	Genetic Algorithm and Gravitational Emulation	Dynamic	Cloud Analyst	Minimization of make span	Priority is not considered for the request which cannot be the real scenario.	Throughput, Makespan, Scalability
9	A Dynamic and Optimal Approach of Load Balancing in Heterogeneous Grid Computing Environment [35]	Dynamic and Optimal Approach of Load Balancing	Dynamic	GridSim	It minimizes the overall task mean response time and maximizes the system utilization.	It gives better performance with high Amount of workload.	Execution time, Throughput, Response time, Associated Cost
10	An Improved K-Subset Algorithm For Load Balance Problems in Cloud Computing [36]	Improved K-Subset	Static	CloudSim	Simplicity, stability and flexibility. minimize the execution time		Execution time
11	A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud	Genetic Algorithm	Dynamic	Cloud Analyst	Balance data distribution to improve cloud computing performance in data-intensive applications,	Time consuming to reach to optimal solution	Average response time

	Computing [37]				such as distributed data mining.		
12	Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach [38]	Hill Climbing	Dynamic	Cloud Analyst	Traditional approaches like Round Robin and First Come First Serve are compared with the new approach of soft computing which gives encouraging results.	Stuck in local optimal solution	Average response time
13	Load Rebalancing for Distributed File Systems in Clouds [39]	Load Rebalancing	Dynamic	Hadoop System	It divides a program into tiny tasks and executed simultaneously and computational resource are assigned to these jobs.	Migration cost, future cost parameter prediction and the impact of prediction error are not considered.	Associated Overhead and Associated Cost
14	Cloud Task scheduling based on Load Balancing Ant Colony Optimization [40]	Load Balancing Ant Colony Optimization	Dynamic	CloudSim	New task allocation depends upon the past results.	Works for applications with the number of tasks varying from 100 to 500.	Degree of imbalance and Makespan
15	VM Level Load Balancing in Cloud Environment [41]	Weighted Signature based load balancing algorithm	Static/Dynamic	Cloud analyst	It minimizes response time and improves services efficiently.	Low scalability	Associated Cost and Response time

Table 1 shows the comparison between different proposed load balancing algorithms with their advantages and limitations.

IV. SIMULATION TOOLS FOR IOT

IoT applications are recognized to be a crucial source of big data. Generally, these IoT applications are supported by clouds where data storage and processing is done by big data processing systems. In order to efficiently support IoT big data applications and to scale the efficiency of cloud infrastructure it is highly important to determine the performance process of these applications and the corresponding big data processing systems in cloud computing environments. In this section some of the simulation tools are discussed.

4.1 *IOTSim:* IOTSim is a simulation tool for the IoT devices and processes big data by using MapReduce technology. It works in the cloud computing environment with minimum cost and lesser time. It does the modelling and the simulation of multiple IoT applications which are running at the same time in the cloud and gives accurate results [42].

4.2 *MATLAB:* MATLAB is simulation software that helps to design, prototype, and deploy IoT applications such as predictive maintenance, operations optimization, supervisory control, and more. It manages the access and pre-processing of data, streaming and archived data using built-in interfaces to cloud storage, relational and nonrelational databases, and protocols. It can be used to design custom IoT analytics and algorithms quickly. Use ThingSpeak, a ready-to-run IoT platform with MATLAB analytics, to prototype and operationalize smaller-scale systems.

4.3 *CloudSim:* CloudSim is a simulation tool provided for the cloud environment. The architecture of Cloudsim consists of layers called simulation engines, cloud services and source code. CloudSim is an open source software. It is built on JAVA and supports both windows and LINUX systems. On the CloudSim other simulation tool are built like Cloud Analyst [43].

4.4 *CloudAnalyst:* CloudAnalyst is a tool featuring visual modelling and simulation of large scale applications that are deployed on Cloud Infrastructures. CloudAnalyst, built on CloudSim, allows description of multiple parameters such as application workloads, including information about geographical location of users, generating track and location of data centres, number of users and data centres, and number of resources in each data centre. Using this information, Cloud Analyst generates reports about response time of requests, processing time of requests, and other factors. Using CloudAnalyst, application developers can achieve the better strategy for allocating resources among available data centres, strategies for prioritizing data centres to serve specific requests, and costs involved in such operations [44].

4.5 *NetSim:* NetSim is a network simulator that can also be used for the IoT system simulator. It can be used to test the performance of real apps over a virtual network. If the user is building a new IoT network from the ground or expanding an existing one, the user can use NetSim to predict how the respective network will perform.

4.6 *Iotify:* Iotify is an IoT simulator that is used to develop IoT solutions in the cloud. This tool helps the user to install an IoT network in their own virtual IoT lab and simulate this IoT network. It can generate customizable traffic from thousands of virtual endpoints and test platforms for scale, security and reliability in order to identify and fix issues before rolling out the final product. It can simulate heavy network traffic to see how network latency affects users overall system performance.

V. CONCLUSION

In recent years the use of IoT devices has started and it is increasing day by day. The devices around us are becoming smarter as they can communicate and make decisions without interacting with humans. They do this entire thing with the help of IoT, It provides every physical object a unique identifier and embedded sensors All these sensors collect the data and store it in a single place called an IoT platform. IoT platform is a cloud service that gives the common platform and language so the data can be understood by all the devices and transfer information. As the IoT devices are increasing the data collected by them is also increasing gradually which can lead to a heavy load on the servers. To maintain the load we use load balancing algorithms. In this paper we have seen the different types of load balancing algorithms and simulation tools present today for the cloud environment.

VI. REFERENCES

[1] Sha, Kewei, et al. "A survey of edge computing based designs for IoT security." *Digital Communications and Networks* (2019).

[2] Aloi, G., Caliciuri, G., Fortino, G., Gravina, R., Pace, P., Russo, W., & Savaglio, C. (2017). Enabling IoT interoperability through opportunistic smartphone-based mobile gateways. *Journal of Network and Computer Applications*, *81*, 74-84.

[3] Sundhari, RP Meenaakshi, and K. Jaikumar. "IoT assisted Hierarchical Computation Strategic Making (HCSM) and Dynamic Stochastic Optimization Technique (DSOT) for energy optimization in wireless sensor networks for smart city monitoring." *Computer Communications* 150 (2020): 226-234.

[4] Gill, Sukhpal Singh, et al. "Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges." *Internet of Things* (2019): 100118.

[5] Amanullah, Mohamed Ahzam, et al. "Deep learning and big data technologies for IoT security." *Computer Communications* (2020).

[6] Santamaria, Amilcare Francesco, et al. "A real IoT device deployment for e-Health applications under lightweight communication protocols, activity classifier and edge data filtering." *Computer Communications* 128 (2018): 60-73

[7] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, Virtual infrastructure management in private and hybrid clouds, IEEE Internet Computing, 13(5):14_22, September/October, 2009.

[8] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, The Eucalyptus open source cloud computing system, in Proceedings of IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2009), Shanghai, China, pp. 124_131, University of California, Santa Barbara. (2009, Sep.) Eucalyptus [online]. http://open.eucalyptus.com.

[9] "Chapter 9: Using Amazon Web Services." *Cloud Computing Bible*, by Barrie Sosinsky, Wiley Publ., 2011, pp. 179–204.

[10] Appistry Inc., Cloud Platforms vs. Cloud Infrastructure, White Paper, 2009.

[11] "Chapter 10: Using Microsoft Cloud Services." *Cloud Computing Bible*, by Barrie Sosinsky, Wiley Publ., 2011, pp.210-218.

[12] L. Youseff, M. Butrico, and D. Da Silva, Toward a unified ontology of cloud computing, in Proceedings of the 2008 Grid Computing Environments Workshop, 2008, pp. 1_10.

[13] B. Hayes, Cloud computing, Communications of the ACM, 51:9_11, 2008.

[14] "Oracle Internet of Things (IoT) SaaS Applications." *Internet of Things (IoT) SaaS Applications | Oracle India*, www.oracle.com/in/internet-of-things/saas-applications.html

[15] Ragmania, Awatif, et al. "An improved Hybrid Fuzzy-Ant Colony Algorithm Applied to Load Balancing in Cloud Computing Environment." *Procedia Computer Science* 151 (2019): 519-526.

[16] Mishra, Sambit Kumar, Bibhudatta Sahoo, and Priti Paramita Parida. "Load balancing in cloud computing: a big picture." *Journal of King Saud University-Computer and Information Sciences* (2018).

[17] LD, Dhinesh Babu, and P. Venkata Krishna. "Honey bee behavior inspired load balancing of tasks in cloud computing environments." *Applied Soft Computing* 13.5 (2013): 2292-2303.

[18] Geetha, P., and CR Rene Robin. "A comparative-study of load-cloud balancing algorithms in cloud environments." *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE, 2017.

[19] Rasmussen, Rasmus V., and Michael A. Trick. "Round robin scheduling–a survey." *European Journal of Operational Research* 188.3 (2008): 617-636.

[20] Rajeshkannan, R., and M. Aramudhan. "Comparative study of load balancing algorithms in Cloud Computing environment." *Indian Journal of Science and Technology* 9.20 (2016):

[21] Hung, Che-Lun, Hsiao-hsi Wang, and Yu-Chen Hu. "Efficient load balancing algorithm for cloud computing network." International Conference on Information Science and Technology (IST 2012), April. 2012.

[22] Katevenis, M., Sidiropoulos, S., & Courcoubetis, C. (1991). Weighted round-robin cell multiplexing in a general-purpose ATM switch chip. IEEE Journal on Selected Areas in Communications, 9(8), 1265–1279. doi:10.1109/49.105173

[23] Improved dynamic load balancing algorithm based on Least-Connection Scheduling Liangshuai Zhu1 , Jianming Cui2 , Gaofeng Xiong1 1.School of information science and engineering, Guilin University of Technology 2.School of continuing education, Guilin University of Technology; Guilin, China

[24] "Load Balancing Algorithms and Techniques." *Load Balancer*, kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques/.

[25] LD, Dhinesh Babu, and P. Venkata Krishna. "Honey bee behavior inspired load balancing of tasks in cloud computing environments." *Applied Soft Computing* 13.5 (2013): 2292-2303.

[26] Nishant, Kumar, et al. "Load balancing of nodes in cloud using ant colony optimization." *2012 UKSim 14th international conference on computer modelling and simulation*. IEEE, 2012.

[27] Jodayree, Mahdee, Mahmoud Abaza, and Qing Tan. "A Predictive Workload Balancing Algorithm in Cloud Services." *Procedia Computer Science* 159 (2019): 902-912.

[28] Li, Dapu, et al. "A hybrid particle swarm optimization algorithm for load balancing of MDS on heterogeneous computing systems." *Neurocomputing* 330 (2019): 380-393.

[29] Adhikari, Mainak, and Tarachand Amgoth. "Heuristic-based load-balancing algorithm for IaaS cloud." *Future Generation Computer Systems* 81 (2018): 156-165.

[30] Vasudevan, Shriram K., et al. "A novel improved honey bee based load balancing technique in cloud computing environment." *Asian Journal of Information Technology* 15.9 (2016): 1425-1430.

[31] Zhu, Yongfei, et al. "A novel load balancing algorithm based on improved particle swarm optimization in cloud computing environment." *International Conference on Human Centered Computing*. Springer, Cham, 2016.

[32] Sharma, Shabnam, Ashish Kr Luhach, and S. A. Sinha. "An optimal load balancing technique for cloud computing environment using bat algorithm." *Indian J Sci Technol* 9.28 (2016): 1-4.

[33] Ghumman, Navtej Singh, and Rajwinder Kaur. "Dynamic combination of improved max-min and ant colony algorithm for load balancing in cloud system." *2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2015.

[34] Dam, Scintami, et al. "Genetic algorithm and gravitational emulation based hybrid load balancing strategy in cloud computing." *Proceedings of the 2015 third international conference on computer, communication, control and information technology (C3IT)*. IEEE, 2015.

[35] Patni, Jagdish Chandra, et al. "A dynamic and optimal approach of load balancing in heterogeneous grid computing environment." *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2*. Springer, Cham, 2015.

[36] Tang, Linlin, Pingfei Ren, and Jeng Shyang Pan. "An improved k-subset algorithm for load balance problems in Cloud Computing." *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*. IEEE, 2014.

[37] Dasgupta, Kousik, et al. "A genetic algorithm (ga) based load balancing strategy for cloud computing." *Procedia Technology* 10 (2013): 340-347.

[38] Mondal, Brototi, Kousik Dasgupta, and Paramartha Dutta. "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach." *Procedia Technology* 4 (2012): 783-789.

[39] Hsiao, Hung-Chang, et al. "Load rebalancing for distributed file systems in clouds." *IEEE transactions on parallel and distributed systems* 24.5 (2012): 951-962.

[40] Li, Kun, et al. "Cloud task scheduling based on load balancing ant colony optimization." *2011 sixth annual ChinaGrid conference*. IEEE, 2011.

[41] Ajit, M., and G. Vidya. "VM level load balancing in cloud environment." *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE, 2013.

[42] Zeng, Xuezhi, et al. "IOTSim: A simulator for analysing IoT applications." *Journal of Systems Architecture* 72 (2017): 93-107.

[43] Goyal, Tarun, Ajit Singh, and Aakanksha Agrawal. "Cloudsim: simulator for cloud computing infrastructure and modeling." *Procedia Engineering* 38 (2012): 3566-3572.

[44] Wickremasinghe, Bhathiya, and Rajkumar Buyya. "CloudAnalyst: A CloudSim-based tool for modelling and analysis of large scale cloud computing environments." *MEDC project report* 22.6 (2009): 433-65.