RESEARCH ARTICLE OPEN ACCESS

# Performance Evaluation of Load Balancing Algorithms on Fog Computing Platform

Obada Qusai Hatem [1], Ahmad Saker Ahmad [2]
Department of System and Computer Networks Engineering
Tishreen University
Latakia - Syria

**ABSTRACT**
Fog Computing is the new era technology, we will get massive benefits in areas of agriculture, business, industry, smart cities and many more. Fog Computing is introduced which easliy transfer sensitive data without delaying to distributed devices. Fog is similar to the cloud ,only difference lies in the fact that it is located more close to end users to process and give response to the client in less time. It is beneficial to the real-time applications, sensor networks, Internet of Things which need high speed and reliable internet connectivity. Fog introduced To get rid out of the issue that Cloud computing suffers like network congestion, less bandwidth utilization, fault tolerance and security etc. Cloud computing which is entirely dependent on the Internet to maintain large applications, where data is shared over one platform to provide better services to clients belonging to a different organization. It ensures maximum utilization of computational resources by making the availability of data, software and infrastructure with lower cost in a secure, reliable and flexible manner. To make fog computing more effective for optimal utilization of bandwidth, and to reduce costs, we have to equally transfer load from clients to all servers, such that no process has to wait for a long time, so here comes load balancing. It attempts to speed up the execution of applications on available resources with proper use of storage to give quick response time to submit user request. In this research, three load balancing algorithms were selected to evaluates their performance in a homogeneous environment in terms of processing time and response time. Throttled , Round Robin and Suffrage are algorithms that have been studied and evaluated their performance using a cloud analyst tool.
***Keywords* :-** Fog Computing, IoT, Cloud Computing, Load Balancing, Fog Nodes.

## I. INTRODUCTION

In today's world, cloud computing has enabled the use of resources, commercial applications and the exchange of database through the Internet faster in the academic and industrial medium. There is no doubt that cloud computing offers many benefits, but it also has some limitations. As the delay time increases which prevents end users from accessing data faster, so it affects the cost of use. Cloud computing is also facing security issues as data has to go a lot further from the cloud to end users, increasing the possibility of data loss. To overcome the limitations of cloud computing,Cisco introduced Fog Computing. Fog Computing offers enormous facilities for storage, speed up processing and communication between sensors, end-users, along with the expansion of cloud computing data centres to the edge of the enterprise network. In the year 2025, nearly 45 per cent of the world's data will be moved to the edge of the network. Fog computing will make this possible because it is a very virtual technology that works in real-time, and acts as an intermediate layer placed between end-users and cloud data canters. Its main features are low delay, location awareness, geographical distribution and support for mobility and real-time interaction [1,2]. To make fog computing more efficient for optimal use of bandwidth, and to reduce costs, we have to transfer the load on an equal footing from clients to all servers, so that no process has to wait for a long time, so here comes the role of load balancing. The load should be balanced first between users and the fog layer, then between the fog and the cloud layer. To try to accelerate the implementation of applications on available resources with the proper use of storage to give a fast response time to serve the user's request. In load balancing, we must ensure that the processing unit, that is, virtual machines, while running tasks should not be overloaded or in an idle state and the system throughput should be at its maximum is a must.[3]

Many load balancing algorithms are presented to many authors. These algorithms are studied and revised based on various parameters such as execution time, bandwidth, cost, priority, reliability, scalability, and task length Basically, effective load balancing algorithms have been implemented in cloud technologies. In this thesis, a set of proposed algorithms has been compared to be implemented in a fog computing environment. In fog computing, load balancing makes the balancing process more feasible and effective with limited resources. Provides access to resources with less bandwidth and less time. In [4] a framework is presented to manage resource efficiency in residential buildings in a fog work environment where the performance of the RR and SJP algorithms was evaluated in terms of processing time and response time and it was observed that the performance of RR was better than the other algorithm.

In [5], a fog-and-cloud model is proposed to calculate energy use for residential buildings in 6 regions of the world. The reason for this work lies in managing the energy requirements of buildings. For this reason, a unique model of energy management is presented and applied as services via the fog and cloud computing system. This app provides real-time features required for power management, flexibility, connectivity and operability. The simulation was performed on the Java platform. Priority Load Algorithm is suggested for efficient selection of virtual machines inside the fog, as consumers get a quick response with minimal delay. However, the results compared to Throttled are not much better, and when compared to RR, the proposed algorithm works very well. In [6] the researchers evaluated the performance of load-balancing algorithms between cloud computing and fog computing where a set of tasks were performed using the two approaches and the results were that fog computing also reduces traffic flow and data congestion to the cloud, and overall response to the task also improved and the results showed that it is better to use fog with the cloud to improve service quality and resource consumption to its maximum value. In [7], the authors proposed a solution to the cloud computing model of the load-balancing problem. If customer requirements are more important, they will be handled by the cloud or else the services will be implemented by the fog layer. The results show that network performance can be improved in terms of delay and load balancing.

In [8], a new architecture is proposed by integrating Cloud with Fog. Fog computing was recently designed to reduce the load on cloud servers. With this integration, a new three-layer architecture has been formed where the first layer contains user devices, the second layer the fog and the third layer the cloud. In this work an example was taken to address energy use requests and consider that addressing this demand can be more efficient compared to cloud computing. To process the request, they proposed the FCFS algorithm. They analyzed the performance in the fog layer., Compared three different algorithms, and it seemed that FCFS performed better than others in response time. In [9] a fog computing approach that is used to provide convenience to consumers and solve all the problems they encountered during the traditional network system was discussed. Fog receives a massive amount of requests from users. Therefore, it was suggested that the HCLB load balancing algorithm reduced the processing time PT and response time RT, but there was an increase in the cost. In [10], an architecture for cloud-and-fog-based power system management was proposed, load management was implemented using virtual machines (VMs) on fog servers using an MSJF algorithm. The purpose of load balancing is to improve the performance of the integrated cloud-based fog structure. The overall performance of this algorithm is measured by means of performance measures PT, RT and cost. From the results, the MSJF algorithm could not outperform the RR and Throttled algorithm due to its limitations in network delay. In [11], three load balancing algorithms were implemented, Round Robin, Odds and Throttled. Throttled algorithm performed better than RR. Under some conditions, the Odds algorithm performs better than the Round Robin and Throttled algorithm. But the more requests on fog from different users residing in different regions of the world, Odds algorithm behaves ineffectively. In [12], the researchers suggested a load management model in a fog computing environment, simulations are performed in Cloud Analyst to compare the performance of different load balancing algorithms. The results of all the mentioned algorithms are compared using the same scenario, Throttled and PSO algorithms outperform the RR algorithm

## II. FOG COMPUTING

fog computing is a model with limited processing, storage and network services capabilities distributed across different peripherals and classic cloud computing, providing a good solution to delayed IoT applications. Most applications of the Internet of things require real-time processing, for example: monitoring systems - safety and fire systems, where failure in these systems can lead to catastrophic results and therefore the cloud cannot be relied upon to manage such systems [14]. There are four deployment model for fog services:

**I. Private Fog:**

A deployment method for providing exclusive use by a single organization that includes multiple consumers.

**II. Community Fog:**

This fog node is provided to a specific community of consumers from organizations with common interests.

**III. Public Fog:**

The fog node that is provided for public use by the general public. It is located at the data canter of the fog service provider.

**IV. Hybrid Fog:**

It is a fog node that is composed of one or more distinctive node (private, community, or public) that remain unique entities but are linked to each other by standard or private technology that allows the ability to transfer data and applications.

## III. LOAD BALANCING

Load Balancing is a systematic method for resetting total loads from overloaded servers to lightweight servers, data centres, or other computing resources. It is basically a process of distributing traffic between different servers with the help of a network-based device or load balancer such as the adapter and router that intercepts traffic to target the site or server and redirects traffic or divides it into individual requests to the desired replica servers based on their availability, when Different servers are loaded via user requests, we need a load balancing policy to distribute the load on unused servers. The essence of load balancing is the distribution of tasks between the nodes, as these nodes are under the management of the load balancer who receives requests and distributes them between their nodes for processing, in a fair manner among all [15]. In distributed systems such as fog computing, load balancing plays an important role, as systems in fog must meet all customer requests at best possible time. Load balancing techniques are mainly classified into two categories:

1)Static: Static load balancing techniques follow a fixed set of rules that are not dependent on the current state of the system. Fixed algorithms are inflexible and require prior knowledge of resources, such as connection time, memory, node storage, node processing capacity, etc. This technique is simple and easy but generally unable to detect attached servers, which leads to uneven distribution of resources, the main problem in This technique is that the current state of the system is not considered during decision making. Therefore, it is not suitable for distributed systems that dynamically change the condition

2)Dynamic: These technologies consider the current state of the system and make a decision on this basis. The main advantage of these technologies is that they allow the transfer of tasks from an overloaded machine to a low-load machine. The dynamic load stabilization technologies are flexible, which improves system performance. During processing, the dynamic technique takes the following steps. It continuously monitors the load in the nodes, in a certain period of time, it exchanges pregnancy and status information between the nodes to calculate the workload of the nodes and redistribute the workload between the nodes. In the event that the node is overloaded, the load is transferred to a light-weight [13].

The algorithms to be studied and evaluate their performance are:

**I. Round Robin algorithm:**

One of the simplest load-balancing algorithms that use the principle of time slices, this algorithm allocates all incoming requests to available virtual machines without considering the current load on all VMs. If the process is not completed at a specific time, it will be placed at the end of the wait queue [12].

**II. Throttled algorithm:**

In this algorithm, the load balancer maintains a table of indexes for virtual machines, as well as their status (available or busy). The request is sent to data centres to find a suitable virtual machine (VM) to perform the recommended task. Then the data canter requests a load balancer to choose a virtual machine. Load Balancer scans the index table from above until the first available default machine is found or the entire index table is checked. If a virtual machine is found, the data canter passes a request to the virtual machine identified by the identifier. Additionally, the data canter installs the new load balancing distribution, and updates the indexes table appropriately. If all VMs are busy, the request will be queued, and you must wait until the VM gets the available status [11].

**III. Suffrage algorithm:**

This algorithm calculates the Minimum Completion Time (MCT) for all incoming tasks on all virtual machines and after that takes the lowest MCT and the second-lowest MCT for each of the incoming tasks. This algorithm calculates the peak of suffering which is the difference between these two values [12]. Then the suffering value is calculated for all the tasks in the system, and in the next step a task is chosen that has the greatest suffering value and allocates it to the virtual machine that executes it with the least MCT. Then the task is deleted from the set of incoming tasks and update the status of available resources and repeat the steps. The basic of this algorithm is that the task that has the greatest value of suffering is that there is a big difference if it is assigned between the machine that gives the best time and the machine that gives the second best time and therefore has priority in obtaining the resource that ends this task with the least possible time.

## IV. PRACTICAL WORK

To compare and evaluate the performance of the load-balancing algorithms that we mentioned earlier, we will use the Cloud Analyst tool. Cloud Analyst was formed primarily to assess the performance and cost of geospatially distributed cloud-based systems that contain massive amounts of user loads based on different parameters. Cloud Analyst is a tool developed at the University of Melbourne that aims to support the evaluation of social networking tools according to the geographical distribution of users and data centres. This tool has a very attractive graphical user interface and huge flexibility to configure any geospatial system such as setting hardware parameters (storage, main memory, bandwidth limit, network delays etc.) for VMS and data centres [16]. For simulation, one scenario is created for all of these algorithms. All of these algorithms run under the same conditions and in the same work environment. The proposed system model is a three-tiered structure, as shown in Figure 1.

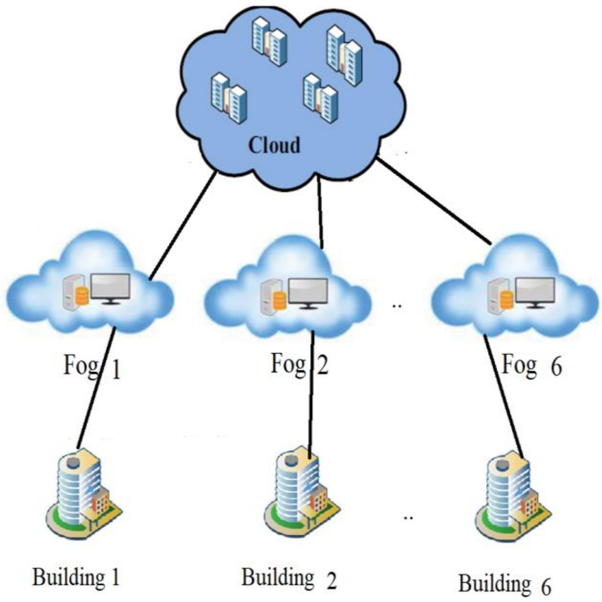


Fig. 1 proposed system model

Tier 1 consists of groups of residential buildings from which requests are received. Tier 2 consists of fog and Tier 3 consists of cloud service providers. Any house in any group of buildings generates the request that is moved to the closest available fog node. The fog processes the request. If this fog node is unable to process, the fog sends the request to the cloud to process this request. In the proposed scenario, 6 regions are taken around the world, and each region has a Fog

Data Centre node, and that fog node deals with a group of buildings in that region, each group of buildings contains a random amount of buildings between 80 To 100 buildings. Each building contains a random amount of houses ranging from 80 to 100 homes. Each fog node contains 50 VM, and each VM can process 100 requests each time. Simulation runs for 24 hours, in this simulation we use the closest data canter policy for Service Broker.

## V. RESULTS

### I. Response Time:

From Figure (2), we note that the Throttled algorithm has the best response time from the rest of the algorithms because this algorithm gives importance to the occupancy status of virtual machines. Suffrage algorithm comes second because it assigns tasks to virtual machines in a way that does not take into account the status of these machines The RR algorithm comes in the last place in terms of response time, where it suffers when a large number of tasks are received, as for up to 2000 tasks, the response time for this algorithm remains good, but with the increase in incoming tasks we notice a significant increase in response time.

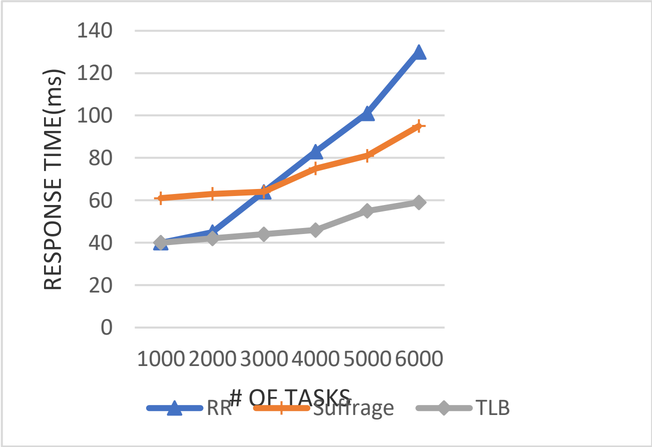


Fig 2 Response time

### II. Processing Time:

We notice from Figure (3) that Throttled algorithm has the least processing time because this algorithm has information about the state of virtual machines and how busy they are, so it makes the best choice when balancing among the rest of the algorithms, and with the increase in the number of incoming tasks this time increases, but it remains less than the time of the rest of the Algorithms. The RR algorithm comes in the second level because when assigning a task to a virtual machine it does not take into account the state of this machine, The Suffrage algorithm is considered the worst among the algorithms in terms of processing time.

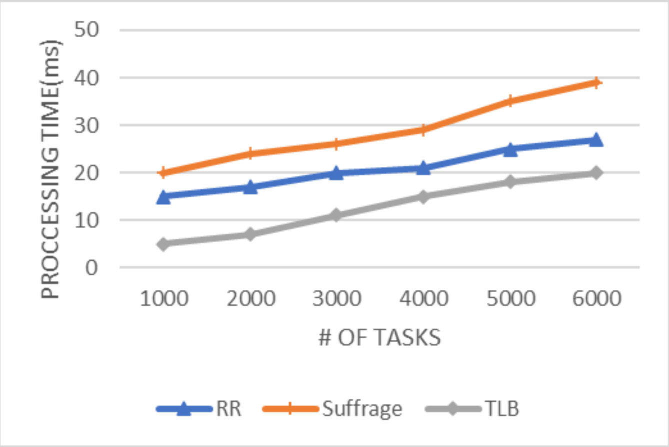


Fig 3 Processing Time

## VI. CONCLUSIONS

Increasing the number of IoT devices in a short time led to challenges in the cloud computing environment through response time and delay, and traffic congestion in order to overcome these problems, Fog Computing architecture emerged in 2012. Fog Computing is an internet-based computing environment where dealing with Some applications that are closest to the end-user. Load balancing is an important factor for the effectiveness of Fog computing, and therefore choosing an appropriate algorithm for load balancing in a Fog computing environment is an important factor in making Fog computing services more effective for users. The performance of three algorithms to balance the load has been evaluated in this research. These algorithms are the suffering algorithm, Throttled algorithm, and Round Robin algorithm through response time and processing time, using the service broker policy closest Data canter. A simulation scenario is suggested in which 6 groups of tasks are generated, and the number of tasks is increased, and then the results are observed. These algorithms are simulated using the Cloud Analyst tool, through simulation results it can be considered that the Throttled algorithm is the best among all algorithms and has achieved better performance than the suffering algorithm and RR algorithm.

## ACKNOWLEDGMENT

The authors wish to acknowledge the Faculty of Information Engineering at Tishreen University for their support of this research.

## REFERENCES

[1] Antonini, Mattia, Massimo Vecchio, and Fabio Antonelli. "Fog Computing Architectures: A Reference for Practitioners." IEEE Internet of Things Magazine 2.3 (2019): 19-25.

[2] Tychalas, Dimitrios, and Helen Karatza. "A scheduling algorithm for a fog computing system with bag-of-tasks

jobs: Simulation and performance evaluation." Simulation Modelling Practice and Theory 98 (2020): 101982.

[3] Téllez, Nadim, et al. "A tabu search method for load balancing in fog computing." Int. J. Artif. Intell 16.2 (2018).

[4] JAVAID, Sakeena, et al. Resource allocation using Fog-2-Cloud based environment for smart buildings. In: 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE, 2018. p. 1173-1177.

[5] TARIQ, Subhan, et al. Priority Based Load Balancing in Cloud and Fog Based Systems. In: International Conference on Broadband and Wireless Computing, Communication and Applications. Springer, Cham, 2018. p. 725-736.

[6] Bhat, Saini. "Load balancing in Fog-Cloud Environment". June (2019)INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING 7(6):71-77

[7] KHATTAK, Hasan Ali, et al. Utilization and load balancing in fog servers for health applications. EURASIP Journal on Wireless Communications and Networking, 2019, 2019.1: 91.

[8] SAEED, Faizan, et al. Load Balancing on Cloud Analyst Using First Come First Serve Scheduling Algorithm. In: International Conference on Intelligent Networking and Collaborative Systems. Springer, Cham, 2018. p. 463-472.

[9] ZAHID, Maheen, et al. Hill Climbing Load Balancing Algorithm on Fog Computing. In: International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. Springer, Cham, 2018. p. 238-251.

[10] NAZAR, Tooba, et al. Modified shortest job first for load balancing in cloud-fog computing. In: International Conference on Broadband and Wireless Computing, Communication and Applications. Springer, Cham, 2018. p. 63-76.

[11] MEHMOOD, Mubashar, et al. Efficient resource distribution in cloud and fog computing. In: International Conference on Network-Based Information Systems. Springer, Cham, 2018. p. 209-221.

[12] ABBASI, Sadam Hussain, et al. Load Stabilizing in Fog Computing Environment Using Load Balancing Algorithm. In: International Conference on Broadband and Wireless Computing, Communication and Applications. Springer, Cham, 2018. p. 737-750.

[13] ATLAM, Hany F.; WILLS, Gary B. IoT Security, privacy, safety and ethics. In: Digital Twin Technologies and Smart Cities. Springer, Cham, 2020. p. 123-149.& Mobile Computing Conference (IWCMC).

[14] Yang, Y. FA2ST: Fog as a Service Technology. In Proceedings of the 2017 IEEE 41st IEEE Annual Computer Software and Applications Conference, Turin, Italy, 4–8 July 2017; p. 708..

[15] PEREIRA, Eder, et al. A Load Balancing Algorithm for Fog Computing Environments. In: Latin American High Performance Computing Conference. Springer, Cham, 2019. p. 65-77.

[16] SINGH, Simar Preet; SHARMA, Anju; KUMAR, Rajesh. Analysis of load balancing algorithms using cloud analyst. International Journal of Grid and Distributed Computing, 2016, 9.9: 11-24..