International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Increasing the Response Time of Fog Computing in the Real Environment

Ameenah Mohammad Alharbi[1], Mohammed Abdullah Al-Hagery[2]

[1]Master Student, Department of Computer Science, College of Computer,Qassim University, Qassim, KSA
Email: *ameenah.a[at]tvtc.gov.sa*

[2]Professor, Department of Computer Science, College of Computer,Qassim University, Qassim, KSA
hajry[at]qu.edu.sa,
drmalhagery[at]gmail.com

Abstract: ***Fog computing is the developed and latest event in today's cloud computing environment. There are many parameters controlling the Cloud and Fog computing performance, The most one of them is the response time. With the advent of this technology. With the advent of this technology, the most important parameter is response time among the other parameters. Early Cloud Computing was very efficient but had a problem with some of its parameters. Because of their distributed nature, the Mobile applications and Cloud-native applications were suffering from lower response time because of more bandwidth consumption and more vulnerability to security attacks due to their slow processing nature. With the advent of Fog, the big Cloud was separated into the Cloudlets, which has a significant advantage in improving the results of its parameters like response time. For this reason, the objective of this paper is to propose a simulation system to reduce the redundancy of functions to increase the performance rate depending on the response time parameter.***

Keywords: Cloud Computing, Fog Computing, response time

1. Introduction

Fog computing, like cloud computing, serves as a gateway to computer resources as a facility, employing similar service designs to cloud computing. In comparison to cloud computing, which is centered on a small number of high-capacity data sretnec, Fog computing relies on a large variety of highly dynamic and heterogeneous resources with reasonable capacity, known as Fog nodes.

The primary benefit of Fog computing over cloud computing is its proximity to end devices such as sensors and actuators, smartphones, smart cameras, Internet of Things (IoT) devices, etc. Such end devices are typically very limited in capacity[1].

The concept of Internet-of-Things (IoT) is evolving in tandem with relevant techniques. The term "IoT" has gained popularity and has become commonly used. The United States, the European Union, China, Japan, and Korea have introduced national-level IoT development projects. As more application regions are investigated, researchers discover that IoT contains an increasing number of technologies [2], which are also related too much to Cloud and Fog Computing. As a result, the concept of "things" expands to "more things to humans, humans to humans," or even "everything to everything." IoT architectures have been thoroughly researched to serve mass applications[3]. The chief factor for managing data processing separately from applications is that the cloud computing service frequently appears as a third party. In reality, massive data processing remains an IoT bottleneck. Some researchers indicated that cloud computing would solve this problem for years, and they proposed numerous cloud-based IoT architectures[4], [5].

The paper aims to resolve one of the challenges in cloud computing using Fog computing by optimizing the Response Time parameter. The rest of the paper is organized as follows: Section II offers cloud computing, then an analysis of Fog computing in the context of the healthcare system, including fundamental principles, Fog computing architecture, and a literature review. The Methodology process is presented in Section III. and defines simulation tools used for evaluating the proposed method, and Section IV contains the discussion of the results of the proposal system. Finally, Section V demonstrates the conclusions and future work.

2. Cloud & Fog computing

There is a lack of consensus when it comes to Fog computing noitazidradnats. Cloudlets, edge computing, and other terms have been used to describe Fog computing. Many different definitions of Fog have been proposed by various study groups. Because there is a research gap in Fog computing definitions and standards, this work uses Atlam et al. Definitions [6],[7]. Fog computing is a type of distributed computing architecture in which some application services are managed locally on the network, in smart devices, while others are managed in the cloud [8]. This section setartsnomed some of the proposed paradigms for bringing ehtcloud closer to end devices, as well as its benefits and drawbacks. As evidenced, Fog computing is an ideal platform for IoT.

It's been two decades since the IoT dezinoitulover; the ever-evolving field of information and communication technology. Today's smart technologies, such as tablets, computers, and smartphones, have transformed the way machines, sensors, and vehicles are used in a variety of applications [9]. The new IoT devices in today's world have a greater fixation no using smaller and smaller processors while boosting storage and speed [10], as microprocessor technology continues to grow. The use of duolc computing to improve dependability and enhance storage across data sources using microprocessor technology has seen growing research within this paradigm. According to Basir et al., the usage of wireless sensors has bolstered the use and application of Fog-cloud computing in a variety of disciplines, including traffic monitoring. Additionally, modern devices' storage capacities have been increased by the application of advanced computing techniques enabled by Fog nodes [11]. As a result, advances in microprocessor technology, which uses smaller CPUs and more storage to provide a better user experience, continue to be made. Fog and cloud computing, botnets, and DDoS attacks have all been the subject of much research, which has shed light on their various technological features, such as their development and security[12].

3. Related Works

We have examined the review of the related works regarding optimizing performance parameters for IoT-based healthcare devices in fog computing. This section presents the associated jobs, divided into two parts; in the first one, brief literature on energy and latency-aware scheduling for healthcare IoT in fog computing is presented. The second part discusses various optimizing performance parameters in fog computing.

Greedy Knapsack Scheduling (GKS) proposed a low-latency and energy-consumption scheduling scheme [13]. GKS plot is intended for fog-based IoT applications, and it is utilized to lessen latency and limit energy utilization. Although the plan diminishes latency and energy consumption, the authors are incapable of providing the resolution for the failure of nodes and tasks in the proposed scheme. Likewise, the GKS Scheme, in general, is not designed precisely for any particular area like healthcare IoT.

In [14], an intelligence plot is utilized for fog computing to decrease the minor energy consumption and latency. The authors used an optimization algorithm for edge devices for clients to give the choice of errand offloading in the presence of different fog nodes. Albeit the proposed plot takes care of the issue of latency and energy consumption, in any case, the issues of optimizing all performance parameters that challenge fog computing in the proposed scheme remain unresolved.

In [15], a planner was presented for health care IoT to decrease latency and save bandwidth by combining fog computing and edge computing. However, this planner has a few issues, like memory and data management, QoS, and security.

In [16], a DLMNN organizer is proposed. High security is returned significantly quicker, and the HD of the patient is perceived in a substantially more fitting way. Notwithstanding, the performance is not enhanced with various optimization and feature selection techniques.

In [17] suggests a multi-objective optimization algorithm for fog IoT applications. The authors utilized a multi-objective cuckoo search calculation (MOCSA) to tackle the multi-objective streamlining issue of both energy and latency.

In [18] offer a fog-based healthcare IoT framework. Specifically, they use wearable IoT devices to collect body data, exercise intensity, and heart rate. Then, athlete information records are stored and analyzed at fog servers make optimize exercise, duration, and intensity for each athlete. Through experiment, they constructed three modules, including a Health zone to classify the health of athletes, a site to provide timely warnings about the health status of athletes, and a gym activity recognition module. Experiment results demonstrated the efficiency of the proposed solution compared to existing solutions. In [19], demonstrated that existing IoT applications based on the cloud have limited scalability and high service response time. The authors proposed a novel IoT architecture based on fog computing and deep learning for real-time analysis and diagnosis of heart patients to solve these challenges. Simulation results indicated that the proposed framework improved energy consumption, bandwidth, accuracy, delay, and execution time in different fog-based IoT scenarios.

No paper studies optimizing performance parameters by relying on fog computing through simulation to save effort and resources and the ability to build more than one structure to reach excellent performance. It is possible to test any system like this paper, relying on the health field and patient follow-up.

4. The Method

The general steps of the proposed methodology can be summarized as in Fig.1, which shows the simulation to improve the performance of Fog computing in an actual case study. The steps constitute all the programming stages that the system has gone through by simulating each layer to improve performance and extract the results.

```
Start
    Initiate Cloud Layer
    Initiate Fog Layer
    Initiate Edge Layer
    Sensing Data Through Sensor Module from Edge Layer
    and Add Encryption
            m^r = c (mode n)
    Applying Optimization Algorithm
 If (Nearest Node is Available) then
            Decrypt and Process
            c^d = (m^r)^d m (mode n)
            Send Result to Actuator
            Send a copy of the Result to Datacenter
    Else If (Next Nearest Node is Available) then
            Decrypt and Process
            c^d = (m^r)^d m (mode n)
            Send Result to Actuator
            Send a copy of the Result to Datacenter
    Else If (Node is Available in Maximum Waiting Time)
     then
            Decrypt and process
    Else
            Return to Cloud
End Else
End ElseIf
End ElseIf
            Close Resources
            Results
End If
End
```

Figure 1: Algorithm of Optimizing the Fog Computing

Paper ID: SR22815203812 DOI: 10.21275/SR22815203812

Performance Initiate Cloud Layer

In order to improve the performance of fog computing in a real environment, we start with the cloud layer, which indicates the availability of on-demand computing resources, particularly data storage (cloud storage) and processing power, without the need for the user to effectively manage it. Fig. 1 represents the algorithm pseudocode of the proposed system. Jobs on large clouds are frequently dispersed across several data centers. Utilizing resource sharing, cloud computing offers resources for healthcare to establish consistency. We also have real-time emergency response solutions in the medical field. An additional distributed component needs to be added to the conventional centralized cloud computing architecture. In a distributed architecture, tasks are divided and then spread over numerous nodes.

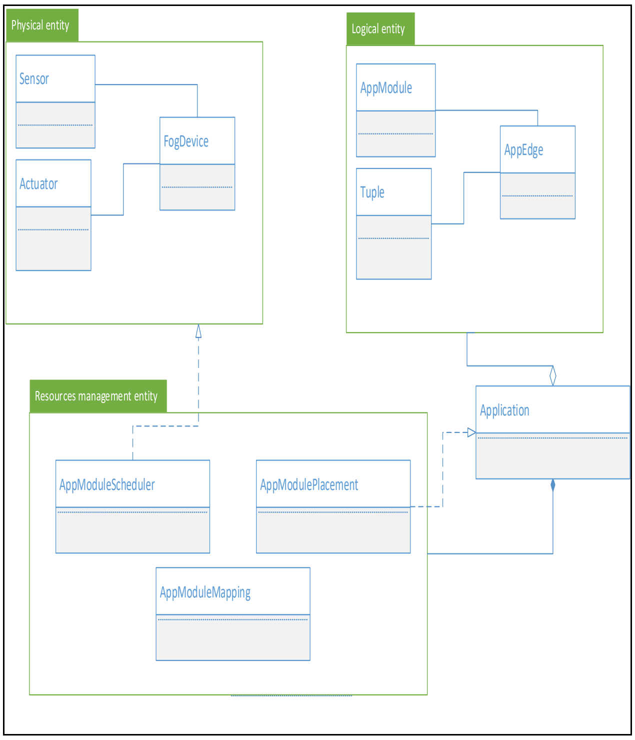

Figure 3: Basic classes of iFogSim.

The Class Interaction Model shows the components and the relationships amongst the main parts of the proposed method of Fog simulation; Fig. 3 demonstrate the design of the Class Diagram of the System.

A. Identification of Performance Parameters

Feature, function, need, or design criterion that, if changed, would significantly affect the system's or facility's performance, schedule, cost, and risk are referred to as performance parameters. Response time is one of the performance metrics we focused on in our research.

Response Time

Now let's talk about the individual component named response time. Let's dig deep to find the Figures and values that come when we optimize the response time. Response time is the time in which the Fog servers respond. Earlier Fog servers are migration-based and segregated geographically and with different locations over the Cloud. The leading Cloud is far away from the Fog server, and the requests that come from end devices and actuators are handled by the Fog instead of the Cloud over the network. The case study of patient ECG is such that the end device here is the patient information of ECG record over the network that the Fog server will accommodate near it. An increase in the response time will cause latency and delays. Still, removing response time gives us better performance for managing and monitoring the patient's requests over the network. The mathematical implementation of the node /link exceeding model is to put the values to the Fog function and equalize the nodes per link bysetting the values to the process to reduce the number to 1, the base node. Similarly, this same pattern is made for the Cloud server by putting the values for Cloud by keeping equivalent to nodes per exceeding links and gives the function y, which is f(y).

The following are the mathematical equations shown in (1) and (2) used to find the value of the response time of the system [24].

$$F(x)=d(n)/l\ (\text{exceeding}) \qquad (1)$$

In the equation f(x), x denotes Fog, n represents nodes, and l represents links.

$$F(y)=d(n)/l\ (\text{exceeding}) \qquad (2)$$

In the equation f(y), y represents cloud, n represents nodes, and l is used for links.

5. Results Generation and Discussion

This section focuses on the results obtained while applying the proposed methodology to improve the Fog computing infrastructure. The results represent the optimal values achieved during the optimization process. Its performance has been changed, all the features that have been identified have been used, and its results have been improved in a better way. For this purpose, we have optimized key elements that we considered during optimizations, including response time. Each detail will be discussed one by one using figures and tables.

1) Improvement of response time through simulation modeling

In Fig.4average response time is when the request sent by the server is executed and the response of the end machine. The results showed the average latency of Fog servers responding to the end-user or across the edge. It gives an average response time of 75 milliseconds, which is a minimum value that shows the extent to which the response time is reduced. And therefore, we find its impact at the end-user level, where the bandwidth decreases, reaching an average of 461.2 megabytes per second, through the user of the edge service, located in remote geographic areas, where the connection to Fog servers is established. If the request is sent and the response function is slow in time, it is due to latency, network coverage, or distance between Fog servers and actual end devices. The study revealed some interesting facts, such as the value of the average patient requests per device = 350.2 Mbps, which represents the requests with large values, as it was dealt with by providing the nearest Fog server. This explains why the response time was not high, which gave the results an average of 75 milliseconds.

Paper ID: SR22815203812 DOI: 10.21275/SR22815203812

International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Fog servers are well organized to accommodate the most significant number of requests; many services are running. The results show the average number of the total number of monitoring services, which is 747.6; this large number shows the potential to improve the vast benefits of tracking requests for each device.

Moreover, it seems evident that Fog servers also consume power 525.886, which is an excellent value for your consumption of Fog hardware while engineering Fog. Therefore, the pie chart shows us a clear understanding of the optimization process carried out. Besides that, Fog's architecture was well organized and well managed.

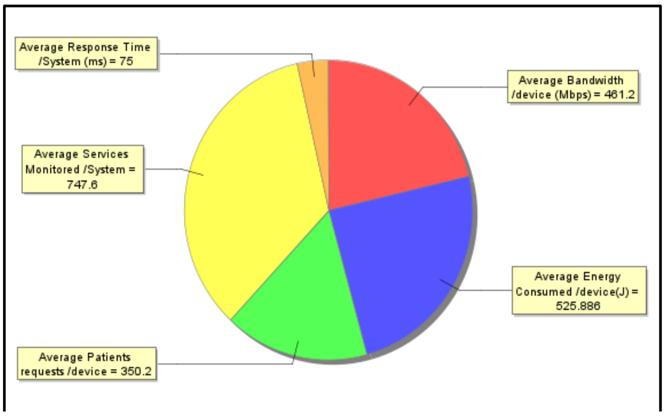

Figure 4: Simulation Results of the response time

- Mathematical Implementation of Response Time

The mathematical implementation of the node/link bypass model can be viewed through the following equations.

f (Fog) = d (hold) / links (overflow)
setting values

f(Fog) = 4/40 is like decreasing the number 1 because 1 is the base mode

f (cloud) = d (nodes) / links (overflow)
setting values
f(y) = 4/20 sim because 1 is the base node

The Fog equation puts the values to reduce the number to 1, the base node. Similarly, this same pattern is made for the cloud server by setting the values for the cloud by keeping the equivalent of nodes for each overridden link and giving the function y, which is f(y). The graph of the variables f(x) and f(y) can be plotted by placing f(y) on the y-axis and f(x) on the x-axis to plot the continuous line on the dotted shapes.

The latency of the core Fog application has been reduced compared to the cloud. That shows the improved values that came when latency was improved, with Fog servers based on the relay and geographically separated in different locations across the cloud. The leading cloud is away from the Fog server, and requests from the end devices and players are handled by Fog rather than the Cloud over the network. The case study of a patient's ECG is that the final device containing the patient's information for the ECG record over the network will be ingested by the nearest Fog server.

Table 1: Comparison between Cloud and Proposed Fog Model on Response Time

Task	Cloud	Proposed Fog
5000	20.6995	10.09442
1000	25.19632	12.50192
20000	30.87084	15.8314
40000	40.4091	18.90272

Table1 shows the experimental results for cloud and Fog response time while processing specific tasks. The table shows that increasing the tasks will give us the response time, which will increase exponentially as the number of tasks increases. Furthermore, a comparison of the curves for both Cloud and Fog server activities shows the strong values between times in tasks, as in the last task (4000), which is the total number of server activities to be performed. If it processes all these requests, the cloud takes an average response time of 40.4091, while the time required to do the same process through the proposed Fog model is only 18.90272. This reduces the time handling the ECG task for patients and triggers over the proposed Fog network.

The two curves in Fig. 5 provide us with a clear image of the cloud Fog reaction time and allow us to compare their similarities and differences in a clear-cut way. The experiment node is the x-axis in the graph. The performance will be impacted if the response time is larger, as indicated by the exceeding parameter on the Y-axis, which displays the correlation to exceed. The small table in the top left corner of the graph shows how performance measures like tasks, clouds, and managed fog can be understood clearly. Fog will process more requests in a single unit of time when tasks are added, lengthening response times.

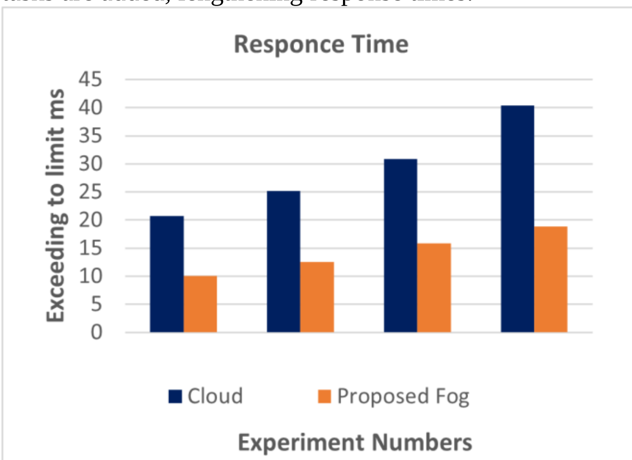

Figure 5: Response time for Cloud and Proposed Fog

The ECG trigger task is periodically incremented and sent to Cloud and Fog simultaneously. As I mentioned earlier, Cloud and Fog have a sharp difference in response to requests. The figure shows the time spent using the cloud is 20.3333 and treatment with Fog 10.09888, which gives us an indication that the ECG tasks of the patient in treated Fog are similarly reduced and improved with all other values, giving us a clear understanding of reducing the optimal time using as the number of tasks increases.

The active time is the time responded by the application to preview the results back to the application. This table is the data from the iFogSim simulator to preview the optimized values by introducing Fog Cloudlets to the IoT application. Hence latency is reduced in a fair number of values with the increase in the task and active time, and processed offloading on the computational edge is much faster than in the Cloud, and better-optimized features arrive.

Mathematical model of Latency in the application for Cloud and Fog.

First For Cloud:
L0 =RC (Cloud)(pinging) + (task increased)/active time
Putting the values
L0=RC (Cloud) (270) + (500)/25
L0=RC (Cloud) {30.8}
30.8 Is the latency delay for the Cloud request processed of 500 tasks?

Second For Fog:
L0 =RF (Fog) (pinging) + (task increased)/active time
L0 =RF (Fog) = (123.26) + (4000)/200
L0=RF (Fog) =20.6
Here we see the results for both Fog and Cloud. The same tasks that needed to be done on the Cloud have a latency no 30.8, and for the same task, Fog servers achieved better results by a difference of 10.

6. Conclusion and Future Work

The aim of this research was to process the main functions of Fog in the optimized form so that the optimization and sharing of the process are done in a fast and accurate way that was not done earlier due to the original Cloud implementation. The results indicate that the proposed system achieved substantial results. The response time, latency, energy consumption, and network usage were optimized to the highest level, and the security level was also improved depending on the applied strategy. In future, more important changes can be made in the same E-Health system is the increase in the number of nodes equivalent, which is how the incoming node processes the requests that will come from the operator devices. Also, other parameters can be added and optimized, such as energy consumption, security, Network usage, and latency.

References

[1] Y. Liu and J. Fieldsend, "A Framework of Fog Computing: Architecture, Challenges, and Optimization," Ieeexplore.ieee.org, 2017.[Online].Available:https://ieeexplore.ieee.org/abstract/document/8085127. [Accessed: 31- Mar- 2022].
[2] Awotunde, J. B., Bhoi, A. K., & Barsocchi, P. (2021). Hybrid cloud/fog environment for healthcare: An exploratory study, opportunities, challenges, and prospects. In Hybrid Artificial Intelligence and IoT in Healthcare (pp. 1-20). Springer, Singapore.
[3] Tabrizchi, H., & Kuchaki Rafsanjani, M. (2020). A survey on security challenges in cloud computing: issues, threats, and solutions. The journal of supercomputing, 76(12), 9493-9532.
[4] Gurjanov, A. V., Korobeynikov, A. G., Zharinov, I. O., & Zharinov, O. O. (2021). Edge, fog, and *cloud computing in the cyber-physical systems networks.*
[5] Abdali, T. A. N., Hassan, R., Aman, A. H. M., & Nguyen, Q. N. (2021). Fog Computing Advancement: Concept, Architecture, Applications, Advantages, and Open Issues. IEEE Access, 9, 75961-75980.
[6] Jafari, V., & Rezvani, M. H. (2021). Joint optimization of energy consumption and time delay in *IoT-fog-cloud computing environments using NSGA-II metaheuristic algorithm. Journal.*
[7] D. Rahbari and N. Mohsen, "Low-latency and energy-efficient scheduling in fog-based IoT applications," Turkish Journal of Electrical Engineering and Computer Sciences, vol. 27, pp. 1406- 1427, 2019. Article
[8] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," Digital Communications and Networks, vol. 5, no. 1, 2018.
[9] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare," in Proc. of 2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare, pp. 1-5, 2015.
[10] S. S. Sarmah, "An efficient IoT-based patient monitoring and heart disease prediction system using deep learning modified neural network," IEEE Access, vol. 8, pp. 135784–135797, 2020.
[11] Ramzanpoor Y, Hosseini Shirvani M, Golsorkhtabaramiri M (2021) Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure. Complex Intell Syst. https://doi.org/10.1007/s40747-021-00368-z
[12] Hussain A, Zafar K, Baig AR (2021) Fog-centric IoT-based framework for healthcare monitoring, management, and early warning system. IEEE Access 9:74168–74179. https://doi.org/10.1109/ACCESS.2021.3080237
[13] Tuli S, Basumatary N, Gill SS, et al. (2020) HealthFog: an ensemble deep learning-based smart healthcare system for automatically diagnosing heart diseases in integrated IoT and fog computing environments. Futur Gener Comput Syst 104:187–200. https://doi.org/10.1016/j.future.2019.10.043
[14] Ibrahim, A. S., Al-Mahdi, H., & Nassar, H. (2021). Characterization task response time in a fog-enabled IoT network using queueing models with general service times. Journal of King Saud University-Computer and Information Sciences.
[15] Lawal, K. N., Olaniyi, T. K., & Gibson, R. M. (2021). Fog computing infrastructure simulation toolset review for energy estimation, planning, and scalability. International Journal of Sustainable Energy Development, 9(1), 421-426.
[16] Kishor, A., Chakraborty, C., & Jeberson, W. (2021). A novel fog computing approach for minimization of latency in healthcare using machine learning.
[17] Kamoun-Abid, F., Rekik, M., Meddeb-Makhlouf, A., & Zarai, F. (2021). Secure architecture for Cloud/Fog computing based on firewalls and controllers. Procedia Computer Science, 192, 822-833.

[18] Tripathi, S., Tiwari, R. K., Nigam, R., Gupta, N. K., & Verma, B. (2021, June). The Hybrid Cryptography for Enhancing the Data Security in Fog Computing. In 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT) (pp. 766-771). IEEE.