Optimization of Fog Computing Efficiency by Decreasing the Latency Level in the Medical Environment

Mohammed Abdullah Al-Hagery 1, Ameenah Mohammad Alharbi 2

1 Department of Computer Science, College of Computer, Qassim University, Saudi Arabia
2 Department of Computer Science, College of Computer, Qassim University, Saudi Arabia

Abstract: Fog computing is the latest technique in today's cloud computing environment, with the advent of this technology. One of the most important parameters affecting fog computing performance is latency. Cloud computing was very effective but needed help in its efficiency according to the values of this parameter, mainly due to its distributed nature. With the advent of the "fog" technology, the big cloud was separated into many small cloudlets, where each cloudlet is a mobility-enhanced small scale. A new architectural element extends today's cloud computing infrastructure. The term "Cloudlets" combined to form fog Computing, which has the main advantages of improving its characteristics listed above.

This paper aims to improve the efficiency of the cloud-computing environment by developing a simulation system to test the best structure before applying it in a real environment. This system will improve the efficiency based on the latency parameter to provide researchers and interested parties with a clear vision to improve the work in a simulation environment and achieve the same methodology in real environments. The simulation system was developed in such a way as to reduce the delay of tasks to increase the performance rate. The results of the simulation system are promising and can be applied in the medical environment.

Keywords: Fog Computing Performance, Performance Measures, Cloud Computing, Latency.

1. INTRODUCTION

Cloud computing benefits both customers and service providers, but it also has certain drawbacks that must be considered. Some drawbacks include the need for internet access and latency, lack of location awareness, and centralized computing systems that use much bandwidth [1]. Cloud computing was further hampered by the billions of queries from Internet of Things (IoT) devices [2]. A new computing paradigm at the intermediary between end devices and cloud computing is suggested to tackle this cloud computing problem.

Next-generation cloud computing is based on the notion of fog computing compared to cloud computing because fog computing has a more robust design. Because fog computing is more dispersed than cloud computing, it receives a higher grade than cloud computing [3]. Fog's processing power is spread out and smaller in number compared to the complicated current centralized cloud computing. Cloud computing can handle many requests, but the fog is far better at managing the little requests from actuators or end devices than cloud computing [4]. Fog computing is not an alternative to cloud computing but rather an extension designed for emergency response, reducing the time it takes to move data to and from the cloud and other latency-sensitive applications. Fog computing refers to a distributed computing architecture in which some application services are managed locally on the network in smart devices while others are managed in the cloud [5]. Fog also gives proximity to its end users. Additionally, it enhances the quality of the service, making it possible to do analytics in real-time and localization. Cloud computing has been adopted as an effective method to process data because of its high computing power and storage capabilities [6]. However, since the cloud computing paradigm is a concentrated computing architecture, most calculations occur in the cloud [7]. Almost all data and inquiries ought to be delivered to a centralized cloud. Even though the data processor has grown significantly, the bandwidth has remained relatively high. So, the network bandwidth is becoming the bottleneck of cloud computing for such a massive amount of data [8]. Different issues still exist in the environment of fog computing.
However, it offers a more significant advantage over the cloud because the whole cloud architecture is geographically separated and connected via a network called nodes or hops. The concept of nodes and clustered approach reduces energy consumption and provides more scaling in terms of energy and efficiency. With the emergence of the fog computing idea, the current complex cloud computing is partitioned into distinct nodes to segregate the paths of the request being handled independently by using each node's processing capacity without heavy reliance on the complicated cloud computing [9].

The rest of the paper is organized as follows: Section 2 introduces cloud computing and fog computing with IoT in healthcare, including basic principles and a comparison of fog computing and cloud and IoT. Section 3 discusses the related works. The research methodology is presented in section 4, which also explains the simulation of the proposed method, while section 5 demonstrates the results and discussion. Section 6 presents the conclusions and future works.

2. CLOUD, FOG COMPUTING AND IOT

There needs to be more consensus in standardizing fog computing, cloudlets, edge computing, and other terms that have been used to describe fog. Different research groups have proposed many different definitions of fog. Since there is a research gap in definitions and standards for fog computing, this paper uses the definitions of Jafari [4] and Atlam et al. [5], in which fog computing is defined as a type of distributed computing architecture that manages a set of application services locally in the network and smart devices. In contrast, others are managed in the cloud [6]. This section highlights some paradigms proposed to bring the cloud closer to the end devices. In addition, it explains the advantages and disadvantages of introducing fog computing as an ideal platform for IoT. IoT has revolutionized the ever-evolving field of information and communication technology. Today’s innovative technologies, such as tablets, computers, and smartphones, have changed how machines, sensors, and vehicles are used in various applications [7]. Table 1 compares fog Computing, Cloud Computing, and IoT using various factors [8]. Currently, the new IoT devices are more fixated on using smaller and smaller processors while increasing memory and speed as microprocessor technology continues to grow.

### Table 1. Summary of Fog Computing, Cloud Computing, and IoT

<table>
<thead>
<tr>
<th>Features</th>
<th>Cloud Computing</th>
<th>Fog Computing</th>
<th>IoT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target User</td>
<td>General Internet user</td>
<td>Mobile users</td>
<td>Stationary mobile</td>
</tr>
<tr>
<td>Working environment</td>
<td>Indoors with massive space and ventilation</td>
<td>Outdoors or indoors</td>
<td>Outdoors Indoors</td>
</tr>
<tr>
<td>Latency</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Delay jitter</td>
<td>High</td>
<td>Very low</td>
<td>Low</td>
</tr>
<tr>
<td>Service Type</td>
<td>Localized information service is limited to the specific location.</td>
<td>Global information collected worldwide</td>
<td>Information specific to the end device</td>
</tr>
<tr>
<td>Distance from Client to Server</td>
<td>Multiple hops</td>
<td>One hop</td>
<td>One hop</td>
</tr>
<tr>
<td>Security</td>
<td>Undefined</td>
<td>Can be defined</td>
<td>Can be defined</td>
</tr>
<tr>
<td>Attack on Data Enroute</td>
<td>High probability</td>
<td>Very low probability</td>
<td>Low probability</td>
</tr>
<tr>
<td>Location awareness</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Geo Distribution</td>
<td>Centralized</td>
<td>Distributed</td>
<td>Dense and Distributed</td>
</tr>
<tr>
<td>Number of the Server Node</td>
<td>Few</td>
<td>Very large</td>
<td>Very large</td>
</tr>
<tr>
<td>Support for Mobility</td>
<td>Limited</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Real-time Interaction</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Type of Last-mile Connectivity</td>
<td>Leased line / wired</td>
<td>Wireless</td>
<td></td>
</tr>
<tr>
<td>Big Data and Duration of Storage</td>
<td>Months and years as it manages big data</td>
<td>Short duration as it transmits big data</td>
<td>Transient as it is the source of big data</td>
</tr>
<tr>
<td>Major Service Provider</td>
<td>Amazon, Microsoft, IBM</td>
<td>Cisco IOx</td>
<td>ARM, Atmel, Bosch</td>
</tr>
</tbody>
</table>
3. RELATED WORKS

Many research works have been introduced to improve the performance of fog computing architecture, resource management, and task scheduling algorithm concepts. Abdulgalil [10] proposed a multitier fog computing architecture based on the analysis of ECG and EEG signals. The proposed architecture compares three tiers. The first tier generates data and transmits it to the second tier. The second tier of the proposed architecture exploits the benefits of emerging fog computing technology for lightweight analysis based on ECG signals. The third tier is responsible for extensive analysis based on EEG signals and data storage in the cloud. Santos [11] introduced a new concept of collaboration architecture for fog computing. His proposed architecture constitutes a referential model for better design and implementation of fog platforms. It powers the freedom of abstraction to make development and deployment at the fog nodes easier and more efficient. Moreover, it introduces expressive mechanisms to define and abstract objects, data analytics, and services.

Yousefpour et al. [12] introduced an IoT-fog-cloud framework for reducing the service delay for IoT applications. The analytical model was also developed to evaluate their policy of the proposed framework, which shows how it helps reduce IoT service delay [13]. Hao et al. [14] proposed a prototype system in the software architecture of fog computing to incorporate the different designs of communication IoT devices with fog nodes. The authors reported their design of WM-FOG, a computing framework prototype for fog environments that embraces this prototype of software architecture and shows the evaluation of their prototype system [15]. Varghese et al. [16] proposed a hybrid fog cloud architecture for low-response time applications, such as firefighting, to provide the solution for IoT service. It supported a growing variety of applications, fifth-generation (5G) wireless systems, including those in the IoT and embedded artificial intelligence.

Gao [17] investigated a task about an offloading strategy for minimizing energy in mobile devices. He also developed a priority-based task-scheduling algorithm with an edge server. The energy consumption, execution time, and execution cost against both the task data sizes and latency requirements were adopted as the performance metric. The performance evaluation results from his research showed that his proposed algorithm reduces task completion time and improves QoS. Mohan et al. [18] also proposed distributing task processing on Edge-Fog-Cloud, which participates in the network. They developed the Least Processing Cost First method to assign the processing tasks to nodes, providing optimal processing time. Peixoto et al. [19] introduced the scheduling algorithms in the hierarchical layer of fog and cloud computing with a three-layer architecture. Their work showed the strategies of scheduling that can be designed in a fog computing environment. In their research, the scheduler prioritizes cloudlet use and optimizes other objectives like reducing network use and cloud costs. Liu et al. [20] proposed a task scheduling algorithm based on fog computing devices’ classification data mining technique. They contributed a novel classification mining algorithm and task scheduling model based on the Apriori algorithm. The task with the minimum completion time was selected to be executed at the fog node with the minimum completion time.

Wang et al. [21-23] proposed the task scheduling strategy in the fog computing scenario. A task scheduling strategy based on a hybrid heuristic algorithm was proposed that mainly solves the problem of terminal devices with limited computing resources and high energy consumption and makes the scheme feasible for real-time and efficient processing tasks of terminal devices. Nguyen et al. [9] introduced a new approach to optimizing task scheduling problems for Bagof-tasks applications in a cloud-fog environment regarding execution time and operating costs. Their algorithm could flexibly satisfy users’ requirements for high-performance processing or cost efficiency. Their work could have been improved, as they tested the algorithm only on small datasets.

4. THE METHOD

The description of the methodology can be summarized in the following steps and then presented in Figure 1:

i. Identifying the basic system requirements.

ii. Initializing both the cloud layer, the fog layer, and the Initializing the edge layer.

iii. Gathering sensor data from the edge layer and encrypting it.

iv. Applying an optimization algorithm.

v. Verifying node availability.
vi. Calculating and delivering the results.

![Algorithm for improving the fog Computing performance regarding the latency.](image)

**Figure 1.** Algorithm for improving the fog Computing performance regarding the latency.

### 4.1. Classes Interaction of the Proposed Application Execution

The Class Interaction Model shows the components and the relationships among the main parts of the proposed method of fog simulation. Fig 2 demonstrates the design of the Class Diagram of the System.

### 5. LATENCY RESULTS AND DISCUSSION

Latency is defined as the delay in service once requested, possibly due to many factors. The architecture of the service turned out to be not good, and fog Computing jobs are usually offloaded to Edge servers called Cloudlets and sometimes called Edge machines. The main tasks are broken down and assigned to specific nodes or leaps. Our developed application scenario has network latency and service level delay once things are done on the cloud instead of in the fog. Cloud servers fulfill the demands at the cloud level, the application consumes data late, and the information preview comes later than fog.

#### 5.1. Latency Mathematical Models Used in the Application

Two mathematical equations are employed for the latency results calculations, as follows:

For Cloud:

\[ L(\text{Cloud}) = \frac{\text{pinging} + \text{TaskIncreased}}{\text{ActiveTime}} \]  \hspace{1cm} (1)

For Fog:

\[ L(\text{Fog}) = \frac{\text{pinging} + \text{TaskIncreased}}{\text{ActiveTime}} \]  \hspace{1cm} (2)
Figure 2. Demonstrates the design of the Class Diagram of the proposed system.

Figure 3 and Table 2 show the experimental results of this paper using equations 1 and 2, where the first column in the table represents the first experiment and its active time value = 25 and ends with the fourth experiment with active time value = 400 as a doubling of the values. At the same time, the number of tasks is increasing gradually from 500 to 6000 from the first experiment to the last one.

In addition, Table 2 contains the data achieved from the iFogSim simulator to preview the optimized values by introducing fog Cloudlets to the application IoT application. One of the table contents is the active time when the application responds to preview the results back to the application. Here, we see the latency results for both fog and Cloud. The same tasks that need to be done on the Cloud have a latency of 30.8; the same task in fog has a latency of 22.97 when the number of tasks is 500, and the rate of reduction by fog is 10; kindly see Figure 3.
On the other hand, the cloud is at its peak with a fair increase in task size of 4000 latency Cloud, which grows pinging time to 946.5 and active time to 200. Still, the fog curve does not increase significantly as it increases on cloudlets, the latency is improved and fairly controlled by the fog server. But the latency curve is decreased with the fog and vice versa with the Cloud with all given tasks, where all values of the Cloud latency compared with the fog in each experiment give higher values than the fog. The latency is reduced and reasonably controlled with this fog server instead of putting the entire job request on the server. Hence, the results indicate that the latency is reduced with increased tasks and active time, processed offloading on the computational edge is much faster than in the Cloud, and better-optimized features arrive, positively affecting final performance.

<table>
<thead>
<tr>
<th>Active Time</th>
<th>Pinging Time Cloud</th>
<th>Pinging Time Fog</th>
<th>Tasks</th>
<th>L(Cloud-Based)</th>
<th>L(Fog-Based)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>270.904</td>
<td>74.4300408</td>
<td>500</td>
<td>30.836</td>
<td>22.977</td>
</tr>
<tr>
<td>50</td>
<td>353.2144</td>
<td>83.3376611</td>
<td>1000</td>
<td>27.064</td>
<td>21.666</td>
</tr>
<tr>
<td>100</td>
<td>419.092</td>
<td>95.3453056</td>
<td>2000</td>
<td>24.190</td>
<td>20.953</td>
</tr>
<tr>
<td>200</td>
<td>946.5893</td>
<td>123.369048</td>
<td>4000</td>
<td>24.732</td>
<td>20.616</td>
</tr>
<tr>
<td>400</td>
<td>1020.345</td>
<td>123.434544</td>
<td>6000</td>
<td>17.500</td>
<td>15.308</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS AND FUTURE WORKS

The proposed system was implemented using various layers in a simulation environment for the medical environment. The application was packaged and containerized using fog servers connected to the centralized and compact cloud through the fog application network. The objective of this paper was to process the main functions of fog in an optimized form so that the optimization and sharing of the processes are done in a fast and accurate method, which was only possible after the implementation based on cloud computing. The results show that the proposed system achieves very significant results. The latency was optimized to a better level. Based on the results discussed in Section 5. The proposed system can be applied in the real environment and thus provides excellent results and achieves significant benefits in terms of performance optimization according to the Latency parameter. This work can be extended by adding more experiments and improving efficiency depending on other new parameters in different environments.
FUNDING

This research has been funded by the Scientific Research Deanship at Qassim University, Saudi Arabia, under the number (COC-2022-1-1-J-25647) during the academic year 1444 AH/2022 AD.

ACKNOWLEDGMENT

The author(s) gratefully acknowledge Qassim University, represented by the Deanship of "Scientific Research, on the financial support for this research under the number (COC-2022-1-1-J-25647) during the academic year 1444 AH / 2022 AD".

7. REFERENCES


DOI: https://doi.org/10.15379/ijmst.v10i4.2216

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.