REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

Energy Efficient Dynamic Task Scheduling and Load Balancing in Autonomous Fog Computing: Low Power & Time-Consuming Data Analysis for Precision Agriculture

Avishek Jana[1], Jayanta Kumar Pahari[1], Arindam Roy[2]

[1]Lecturer, Dept. of Computer Sc. & App., Prabhat Kumar College, Contai, Purba Medinipur, 721401, WB, India
[2]Associate Professor, Dept. of Computer Sc. & App., Prabhat Kumar College, Contai, Purba Medinipur, 721401, WB, India

Abstract
In precision agriculture, timely decisions based on data analysis are crucial. Despite the growing usage of cloud computing, there are still unresolved issues arising from inherent problems such as unreliable latency, lack of mobilization support, and location awareness. In the meantime, optimized cloud computing is known as fog computing, which is an efficient method for low power consumption. Both low power consumption and efficient time utilization are crucial factors in today's world. Therefore, it is essential to address these issues effectively. One of the significant benefits of fog computing is that it minimizes power consumption, making it an attractive solution. Until now, most of the work related to calculation, power consumption, and latency was done using a centralized approach. However, we have now developed a distributed method of calculating and measuring power consumption and delay. This new method minimizes the amount of error and decreases the rate of error, resulting in lower power and time consumption. The advantage of fog computing is minimizing power and time consumption. We aim to validate the hardware infrastructure in our lab and verify it using the ifog sim simulator.

Keywords: Cloud computing, Fog computing, Offloading.

I. Introduction

In this paper, we introduce a new technique that minimizes power consumption and work duration by applying the Fog computing concept in a distributed manner. It is predicted that the number of cloud users will continue to increase each day until 2016. The IT sector is expected to see a surge in cloud users as well. Therefore, a new technology called fog has been introduced to address this issue. Cloud computing is a highly effective solution for addressing large-scale problems, due to its ability to handle massive power and energy requirements. However, for smaller computations, it may not be necessary to offload the problem onto the cloud, particularly if distance is a factor. To address this complexity, fog computing technology has been proposed, which offers geo-distribution capabilities. Fog technology is particularly useful for close communication, as it delivers the desired results within a minimum time frame.

Nowadays, trafficking is a major issue, and it's important to minimize the number of users to avoid collisions. Cloud computing is useful for processing large amounts of data, but it faces a big challenge when it comes to unnecessary trafficking. To avoid this problem, Cisco has introduced the technology of Fogging. This new technology was proposed in 2014 and uses edge computers and devices. Fog computing is efficient in terms of time and power consumption [1]. The Internet of Things is a technology that allows each electronic device to have a unique identity and perform remote sensing and actuation. IoT devices can exchange data with other connected devices, collect and analyze the data, remove noise, and compute results [2]. The proposed technology for smart cities and homes has greatly improved our quality of life. With this new technique, resource utilization has also increased. It has made managing health services and disaster situations much easier nowadays.[3]. IoT devices include utility meters, Bluetooth-connected devices, thermostats, irrigation pumps, sensors, and control circuits for electronic car engines. For instance, let's take a surveillance camera that can adapt its modes based on the time of day. Experts predict that by 2025, IoT devices will increase the GDP by almost 11% of the world economy. Realistically, IoT devices are very useful, and the technology advancements have reduced the time required for normal tasks. However, excessive data volume has created bottlenecks, which is why experts are working on ways to avoid this situation. Currently, nearby devices offload data to the cloud device for computation, which leads to latency issues. A new technology called fog computing minimizes latency and processing power while supporting device mobility.

Usage of Fog Computing:

- Fogging is an extended form of cloud computing that can be used for shared resources and to develop system
- It supports a variety of development approaches and devices. The use of fog is a testing and development approach that requires securing a budget and setting up the environment with physical assets, significant manpower, and time. The next step involves the installation and configuration of the platform.
- Thanks to this new technology, applications use fewer device resources.
- Mobile devices can be connected to services delivered on an Application Programming Interface (API) architecture

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

• , which makes it easier to access fog-supported services. Furthermore, information is backed up and stored in the cloud, which improves reliability and saves time. Backing up data has always been a complex and time-consuming operation, but with fog computing, this process is made simpler and more efficient.

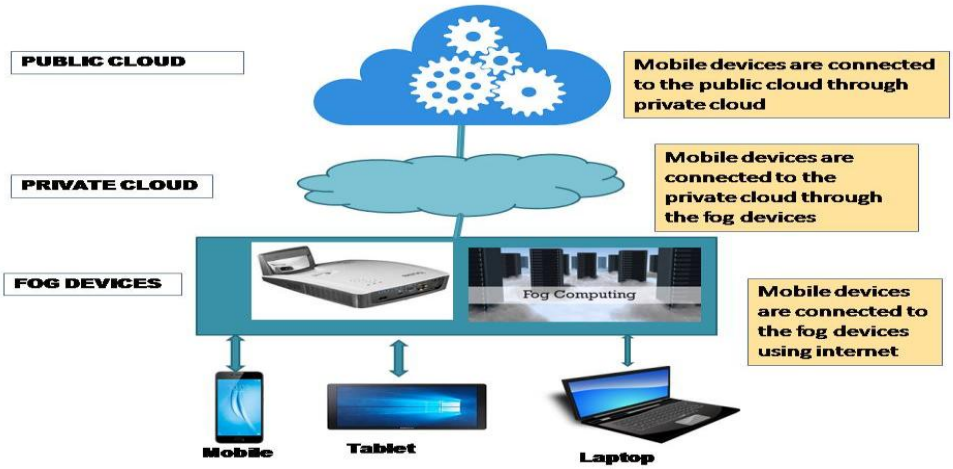


Fig1: Mobile devices are connected to the cloud through the edge devices

In this figure, we can see that many IoT devices are naturally distributed and often embedded in environments with different types of operating services. This is called a heterogeneous network. Data travels through many points from one host to another, and each step is crucial because the data is offloaded and fetched. In this architecture, it is clear that the data travels from one point to another, and each time, the transmission delay and offloading delays are included, leading to extra time and power wastage. To address this issue, experts have proposed a new technique called "fog," as all devices in between are capable of data computation, and the concept of distributed computing is utilized. All devices act as computational devices, and the main focus is on resource sharing, which is made easy by utilizing the storage capacity and computational capabilities of intermediate devices. This new technique improves the quality of services of the computer network and saves battery life and time. However, due to faraway devices, data offloading, and data fetching become complex, and noise levels can be high. To overcome this, technologists are considering increasing the number of repeaters between two devices to minimize overhead. The Internet of Things is often costly and cannot provide desirable features or computation within the budget, so ifog sim is proposed in this case. Currently, the case study is ongoing, and in the future, it will be completed. We calculate the ifogsim and compare two basic resource management policies, and the fragmentation is done for resource sharing and network usage.

II. Related Works

Nowadays, people prefer portable devices such as smartphones, tablets, and iPads over desktops and personal computers. However, one major issue with these mobile devices is that they often have poor network coverage. Due to the high speed of these devices, the network bar may appear too low.

A: Fog Computing

Ruilong Deng et al. proposed a framework to reduce latency and improve service rates using centralized fog computing [1]. In my paper, I introduced a new scenario called distributed fog computing that is more efficient and minimizes latency. The paper I referred to is titled "Fog Computing: A Platform for the Internet of Things and Analytics" by Bonomi F, Milito R, Natarajan P, and Zhu J. In this paper, we also cover the communication between IoT devices and fog devices [3]. "A Power and Latency Aware Cloudlet Selection Strategy for Multi-Cloudlet Environment" - this paper emphasizes low power consumption techniques in the cloudlet scenario. These techniques are also implemented in our research paper [5].

In this research paper [1], the authors present an approach to achieve optimal workload allocation between fog and cloud computing. The objective is to balance delay and power consumption. They propose an approach to decompose the primal problem into three sub-problems and solve them individually. On the other hand, our paper focuses on distributing tasks among nearby devices. We introduce a new fog technology to accomplish this task.

Stavros et al. divided the total area into small parts to utilize the entire portion of the device [2]. But in our paper, we designed a new infrastructure through which we can access more than one resource at a time.

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

B: Cloud Computing:
This is an emerging era where cloud devices are used as a more powerful alternative to mobile devices for compiling data. These devices can be used as backup devices in case a mobile device fails to process data.
If we offload data into cloud infrastructure, the result can be high latency and power consumption. Sensors collect data and remove any noise from it before computing and outputting the results. This distributed computing helps speed up the entire procedure, and data is stored in an organized manner, increasing computation speed and reliability while decreasing redundancy. Propagation latency is less compared to cloud computing. Fog computing requires real-time data processing, where both logical correctness and timeliness are important features.
In the figure above, we see sensors working as IoT devices, collecting data from various fields such as agriculture, medical systems, cars, and other electronic gadgets. Data is collected from various fields like computer robotics, fly-by-wire aircraft, or anti-lock brakes on vehicles. In the case of cloud computing, resources are far away, causing high latency, whereas in fog computing, devices are closer, resulting in faster computation and output. However, communication failures can occur in fog computing. Although cloud computing is useful for virtual infinite resources, bottlenecks, and data hazards can occur in this environment.

III.Motivation and Contribution of the Proposed Work
Firstly, the main issue is insufficient bandwidth, which can be resolved by addressing the high bandwidth requirements necessary for offloading data to the cloud. One solution is to use distributed fog devices that can fulfil the network bandwidth needs by using high-speed network cables or 5G technology. This is possible because fog devices are located locally.
Solution: This distributed fog device is capable of meeting network requirements through 5G technology.
Solution: This fog computing technology enables the simultaneous use of multiple devices.
Problem: "When dealing with a large volume of data, the main drawback is the bottleneck problem that can occur."
Secondly, there is a bottleneck problem when computing large amounts of data. This can be overcome by using distributed fog technology, which allows for the use of more than one device at a time.

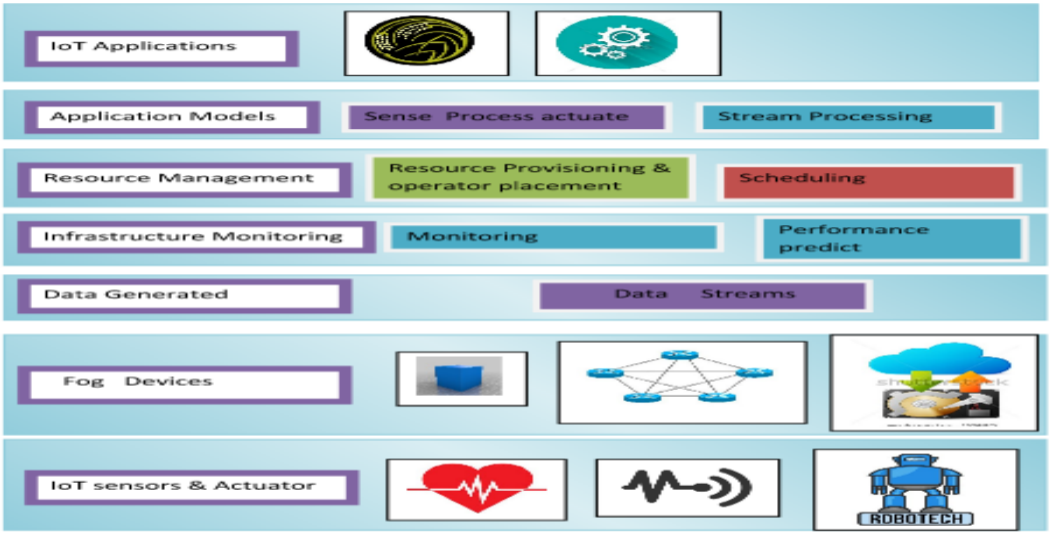


Fig3: - Fog computing generic architecture

Explanation of the architecture: System Model and Problem Formulation-Until now, the architecture of fog devices has only consisted of four layers. However, there are limitations to this architecture, as it is not always applicable. In this architecture, sensors collect data from the environment, which is then coded by the devices. If the fog layer fails to execute this data, it is offloaded to the private cloud layer. If any failure occurs in the private cloud layer, then it is offloaded to the uppermost layer, known as the public cloud layer.

IV.5-layered architecture
In this paper, we propose a five-layer architecture that is believed to be better than any other architecture proposed so far. The five layers are as follows: the Mobile Device Layer, the Cloud Manager, the Fog Computing Device, the Private Cloud, and the Public Cloud. We will describe each of these layers in detail, one by one.

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

Fig4: 5-layered architecture

The layered architecture involves different layers of devices that perform various functions. The first layer includes devices such as mobile phones, PCs, laptops, tablets, and PDAs. However, if an application is complex and too big, it cannot be executed on any first-layered device. To execute such an application, more powerful devices with greater computing and storage capabilities are required.

To avoid this problem, first-layer devices send a request to the third-layer devices that are located after the cloud manager, which is located in the second layer. The cloud manager decides which application should be sent to the next layer and which instance available in the next layer should execute the application.

The third layer consists of Fog devices that are located locally to enable the operation of computing, storage, and networking services between end devices and cloud computing data centers. If an application can be executed at the Fog computing layer, it sends a response to the downstream layer. Otherwise, the existing Decision-Making System (DMS) decides to send the request to the next layered devices.

The fourth layer consists of private cloud devices that provide a computing infrastructure across different users with self-service and scalability. It is multi-purpose and can prepare machines, change computing resources as per requirement, and create multiple instances for complex computing jobs. The private cloud layer also contains a private cloud manager (**Mng_{pr}**), which decides which instance is free for executing the application. If the application can be executed at the private cloud layer, it sends an acknowledgment to the downstream layer. Otherwise, the private cloud manager sends a request to the public cloud to execute the application.

The fifth layer consists of pubic cloud devices that share a computing infrastructure across different users, business units, or businesses globally. The public cloud layer holds a public cloud manager (**Mng_p**), which decides which instance can compute the application. If the application can be executed at the public cloud layer, it sends a response to the downstream layer. Otherwise, the public cloud manager sends an "ERROR MSG" signal.

V. System Model

Table1: Details of parameter table

Symbol	Definition	Unit
i,N,S	Index, number, set of fog devices	n/a
D_i^{fog} **,S**T, **R**T, Y_T, P_{Delay}, **Q**T, D_{ST}	Time for fog device, sending time, Receiving Time, processing time, propagation Delay, queuing time, data storing time	Time(mili-second)
Xi	computation amount	Request/sec
ξ_s , λ_r, κ_p, ψ_p, ϖ_t, δ_d, P_i^{fog}	sending power, receiving power, processing power, power for propagating data, power to maintain queue, power for storing data, total power of fog device	Unit power
S_t^{pr}, R_t^{pr}, X_t^{pr}, P_t^{pr}, Q_t^{pr}	For private cloud: sending time, receiving time, processing time, propagation delay, queuing delay	Millisecond

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

φ_i, ∂_i, f^p	Phase of the processor, number of processors, frequency of the processor	Unit
ϖ_p^{pri}, ψ_p^{pri}, Θ_p^{pr}, ζ_p^{pr}, χ_p^{pr}, Ω_p^{pr}	For private cloud: Processing power to process unit data, power needed for receiving data, power to propagate data, sending power, queuing power, power to storing	Unit power
Make it large		
D_i^{pub}, S_t^{pub}, R_t^{pub}, Y_t^{pub}, Q_t^{pub}, P_t^{pub}	For public cloud: Total Delay, Sending Time, Receiving Time, Processing Time, Queuing Time, propagation delay	millisecond
P_i^{pub}, ζ_p^{pub}, ψ_p^{pub}, Θ_p^{pub}, χ_p^{pub}, φ_i^{pub}, ∂_i^{pub}	For public cloud: Total power consumption, sending power, Receiving Power, propagation power, Queuing power, processing power, storing power	Unit power

Defination1: The time required for data offloading depends on various parameters such as sending time, receiving time, processing time, propagation delay, queuing time, and data storing time
. These collectively contribute to the delay of the fog device(for data offloading):
Delay^{fog} (Adjourn) = Sending Time +Receiving Time + Processing Time + Propagation delay+ Queuing Time +Data Storing Time

$$D_i^{fog} = \sum_{i=1}^{n} X_i^{fog} \ (S_T + R_T + Y_T + P_{Delay} + Q_T + D_{ST})$$

If we want to transfer a certain amount of data (Xi) from the cloud manager to an edge device, we need to take into account various factors such as sending time (S_T), receiving time (R_T), processing time Y_T, propagation delay P_{Delay}, queuing time (QT), and time needed to store the data (DST). The sum of all these times will give us the total duration required D_i^{fog} to complete the transfer.

Definition 2: The power consumption of fog devices can be mathematically represented by a set of tuples.

$$P_i^{fog} = \sum_{i=1}^{n} X_i(\xi_s S_T + \lambda_r R_T + \kappa_p P_T + \psi_p P_{Delay} + \varpi_t Q_T + \delta_d D_{ST})$$

If power X_i is needed to transfer data to an edge device, then the sending power ξ_s, receiving power λ_r, processing κ_p power, power for data propagation ψ_p ϖ_t, power for queue maintenance, and power for storage are all taken into δ_d account. The total power consumption P_i^{fog} is calculated with the summation of all these powers, where the value of "i" varies from 1 to "n".

Defination3: Below is the mathematical representation of the Computation Delay of Private Cloud Server for data offloading scenarios

$$D_i^{private} = S_t^{pr} + R_t^{pr} + X_t^{pr} + P_t^{pr} + Q_t^{pr} + D_{st}^{pr}$$

. It includes $D_i^{private}$ sending time S_t^{pr}, receiving time R_t^{pr}, processing time X_t^{pr}, propagation delay P_t^{pr}, and queuing time. Q_t^{pr}

Definition 4: The power consumption of

$$P_i^{private} = \sum_{i=1}^{n} X_i^{pr}[S_t^{pr}\zeta_p^{pr} + R_t^{pr}\psi_p^{pri} + P_t^{pr}\Theta_p^{pr} + D_{st}^{pr}\Omega_p^{pr} + \varphi_i\partial_i(Y_t^{pr})\, f^p \varpi_p^{pri} + Q_t^{pr}\chi_p^{pr})]$$

a public cloud server is determined by several factors. These include the phase of the processor φ_i (whether it is on or off), the amount of computation X_i^{pr} required, the number of processors needed ∂_i, the frequency of the processor f^p, the power required to process unit data ϖ_p^{pri}, power needed for receiving ψ_p^{pri} and propagating data Θ_p^{pr}, power

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

consumption du ζ_p^{pr} χ_p^{pr} ring queuing, and power requirement for storing data Ω_p^{pr} .

Definition 5: There is a delay in accessing the public cloud for data offloading.

$$D_i^{pub} = \sum_{i=1}^{n} X_i^{pub} (S_t^{pub} + R_t^{pub} + Y_t^{pub} + P_t^{pub} + Q_t^{pub} + D_{st}^{pub})$$

This delay is calculated using seven tuples, which include the total delay D_i^{pub} for offloading X_i^{pub} , the amount of data being offloaded, and various factors such as sending time S_t^{pub} , receiving time R_t^{pub} , propagation delay P_t^{pub} , queuing time Q_t^{pub} , processing time Y_t^{pub} , and storing delay D_{st}^{pub} .

Definition 6: Power consumption at public Cloud: for data offloading: Power consumption in public cloud instances is

$$P_i^{pub} = \sum_{i=1}^{n} X_i^{pub} (S_t^{pub} \zeta_p^{pub} + R_t^{pub} \psi_p^{pub} + P_t^{pub} \Theta_p^{pub} + Q_t^{pub} \chi_p^{pub} + \varphi_i^{pub} \partial_i^{pub} (Y_t^{pub} f^{p_x}) \varpi_p^{pub} + D_{st}^{pub} \Omega_p^{pub})$$

If we consider the total power consumption P_i^{pub} , an equation can be constructed based on all the values including power required for storing purpose ζ_p^{pub} s, power required for receiving the result ψ_p^{pub} , power required to propagate the data Θ_p^{pub} , power required for queuing purpose χ_p^{pub} s, the phase of the processor on/off φ_i^{pub} , the number of processors needed ∂_i^{pub} , the power to process the data ϖ_p^{pub} , and the power consumed for storing purpose Ω_p^{pub} s.

Definition 7: Delay at Fog device: for Code offloading: As like as the data offloading scenario time requirement is calculated through some parameters.

$$D_j^{fog} = \sum_{i=1}^{m} Z_j^{fog} (S_{Tc} + R_{Tc} + \mathrm{E}_{Tc} + P_{Dc} + Q_{Tc} + \delta_{Tc})$$

In scenarios where code offloading is involved, the total delay is D_j^{fog} calculated based on various parameters such as the sending time S_{Tc} , receiving time R_{Tc} , execution time of the code E_{Tc} , propagation time of the code in the air medium P_{Dc} , queuing time Q_{Tc} , and time taken to save the code in the fog server δ_{Tc} .

Defination8: Power consumption at fog device: for code δ_{Tc} offloading: The power consumption is calculated by some parameters.

$$P_j^{fog} = \sum_{j=1}^{m} Z_j^{fog} (\xi_{s'} S_{Tc} + \lambda_{r'} R_{Tc} + \kappa_{p'} E_{Tc} + \psi_{p'} \mathrm{P}_{Dc} + \varpi_{t'} Q_{Tc} + \phi_{\delta'} \delta_{Tc})$$

If we consider the total power consumption P_j^{fog} , it is calculated based on several parameters. To offload Z_j^{fog} the amount of code in the fog device, we need $\xi_{s'}$ power to store a unit amount of code, $\lambda_{r'}$ power to receive the result, power to execute a unit amount of code $\kappa_{p'}$, power to propagate a unit amount of code P_{Dc} , power for queuing purpose $\varpi_{t'}$ s, and power to store the code $\phi_{\delta'}$.

Definition 9: Delay at private cloud: for code offloading: As the data offloading in the private cloud same condition happens for the code offloading cases, Which is mathematically expressed by some parameters.

$$D_j^{private} = \sum_{j=1}^{m} Z_j^{pr} (S_{t'}^{pr} + R_{t'}^{pr} + E_{t'}^{pr} + P_{t'}^{pr} + Q_{t'}^{pr} + \delta_{t'}^{pr})$$

The total time required to Z_j^{pr} transfer a certain amount of code to the private cloud i $D_j^{private}$ includes the time it takes to send the code $S_{t'}^{pr}$, receive the necessary result $R_{t'}^{pr}$ s, and execute the code $E_{t'}^{pr}$. This also takes into account the propagation delay $P_{t'}^{pr}$, queuing delay $Q_{t'}^{pr}$, and the time required to store the code $\delta_{t'}^{pr}$.

Definition 10: Power consumption at private cloud: for code offloading: χ_{pc}^{pr}

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

$$P_j^{private} = \sum_{j=1}^{m} Z_j^{pr}[S_{t'}^{pr}\zeta_{pc}^{pr} + R_{t'}^{pr}\psi_{pc}^{pri} + P_{t'}^{pr}\Theta_{pc}^{pr} + \varphi_j^{pr}\partial_j^{pr}(E_{t'}^{pr})f^{p'}\varpi_{pc}^{pr} + Q_{t'}^{pr}\chi_{pc}^{pr} + \delta_{t'}^{pr}\phi_{pc}^{pr}]$$

$P_j^{private}$, The text seems to be describing various aspects of processor power usage. It mentions the phase of the processor φ_j^{pr} being on or off, the number of processor ∂_j^{pr} s required, the frequency of the processor $f^{p'}$, the power needed to process code ϖ_{pc}^{pr}, the power required for storage, the ζ_{pc}^{pr} power required for receiving result ψ_{pc}^{pri} s, the power required for propagating code Θ_{pc}^{pr}, and the power required for queuing purposes.

Definition 11: Delay at public cloud: for code offloading: D_j^{pub} to offload Z_j^{pub} the amount of code in the public cloud it's required $S_{t'}^{pub}$ time. Receiving time $R_{t'}^{pub}$, propagation delay $P_{t'}^{pub}$, and queuing delay $Q_{t'}^{pub}$ are.

$$D_j^{pub} = \sum_{j=1}^{m} Z_j^{pub}(S_{t'}^{pub} + R_{t'}^{pub} + E_{t'}^{pub} + P_{t'}^{pub} + Q_{t'}^{pub})$$

Definition 12: Power consumption at public cloud: In the public cloud the power consumption is calculated by all other parameters like the power to send the amount of code at one time unit. Power needed to

$$P_j^{pub} = \sum_{j=1}^{m} Z_j^{pub}[S_{t'}^{pub}\zeta_{pc}^{pub} + R_{t'}^{pub}\psi_{pc}^{pub} + P_{t'}^{pub}\Theta_{pc}^{pub} + \varphi_j^{pub}\partial_j^{pub}(E_{t'}^{pub})f^{px'}\varpi_{pc}^{pub} + Q_{t'}^{pub}\chi_{pc}^{pub})]$$

When it comes to offloading code to a public cloud, several factors affect power consumption. These factors include the total power required P_j^{pub}, the power ζ_{pc}^{pub} needed to send and receive a unit amount of task or code ψ_{pc}^{pub}, the power required to propagate the code Θ_{pc}^{pub}, the device's on/off φ_j^{pub} phase, the number of processors needed ∂_j^{pub}, the frequency of the working processor $f^{px'}$, the power needed to process the code ϖ_{pc}^{pub}, and the power required for queuing purposes in a public cloud server χ_{pc}^{pub}.

Defination13: By adding up the power requirements of the private cloud instance, public cloud instance, and other cases, we can determine the total power requirement for the system.

$$P^{System}(X,Y,Z) = P^{fog}(X) + P^{private}(Y) + P^{public}(Z)$$

VI. Algorithm For 5-layered architecture:

Initialization of inputs:
- PATHS(P) : {P0,P1,P2,P3,.....Pn}
- M: Mobile Devices in the first layer
- Em: Mobile Device in the edge layer
- F: Fog devices
- CPU $^{req}_{m}$: Require processor
- **CPU$^{avail}_{Em}$**: Available Processor
- **t_{Em}:** Time estimation for executing Em on the Fog Computing Device layer
- **t_{rEm}**: Time requirement for executing
- **t_{Prm}**: Time estimation for executing Prm on the private cloud layer

Procedure:
START:
for p € PATHS do across all path
Playlist : = { };
1:For mobile device m € M do
If all processors of m are placed
Then
Add m to place list;
End
Else
Add m to the place list as the first mobile device ;
End

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

End
2:For device Em € place list do
If Em is already placed on device f € p
Then
Merge Em with upstream instance;
While CPU $^{req}_{m}$>= CPU$^{avail}_{Em}$ do
Em := parent(Em);
end

Mobile Device Em placed on Fog Computing Device w.r.t Cloud

3:For device Em € place list do
If Em is already placed on device f € p
Then
Merge Em with upstream instance;
While CPU $^{req}_{m}$>= CPU$^{avail}_{Em}$ do
Em := parent(Em);
end

Mobile Device Em placed on Fog Computing Device w.r.t Cloud

4:for device Em € placelist do
Ift_{Em}>= t_{rEm}
Execute Em device on fog computing device layer;
response back to the mobile device layer;
end
else
Em request private cloud layer ;
While CPU $^{req}_{Em}$>= CPU $^{avail}_{prm}$ do
Prm := parent(Prm) ;
end
Place Prm: = Em on cloud
End

Mobile Device Prm placed on Private Cloud Layer w.r.t Private Cloud

5:for device Prm € private cloud layer
if t_{Prm}>= t_{rPrm}
execute Prm device;
response back to the downstream layer ;
end
else
Prm request to public cloud layer ;
While CPU$^{req}_{Prm}$>= CPU $^{avail}_{Pm}$ do
Pm := parent(Pm);
End
Place Pm:= Prm on Public Cloud Pm;
End

Mobile Device Pm placed on Public Cloud Layer w.r.t Public Cloud manager

6:For device Pm € Public Cloud layer
If executing Pm on a public Cloud Layer
Then
Response back to downstream layer;
End
Else
7: " ERROR MSG " response back to downstream layer;
End
End
End
End

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

End
End

VII. Result: Implementation of specific hardware infrastructure

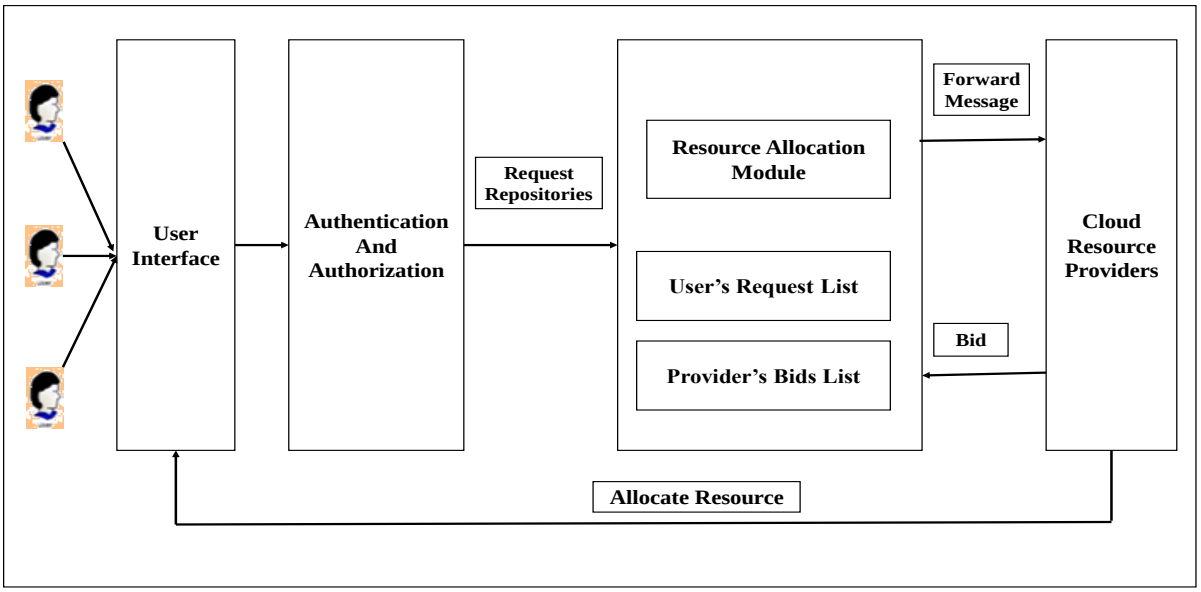


Fig 5: Architecture of Hardware Infrastructure

Table2: Configuration of Local Cloud Servers (second layer)

SL NO	RAM	HDD	PROCESSOR	OPERATING SYSTEM
Fog instance 1	8 GB	500GB	Quad-core	Fedora
Fog instance2	8GB	500GB	Quad Core	Fedora
Fog instance 3	8GB	500GB	Quad Core	Fedora
Fog instance 4	8GB	500GB	Quad Core	Fedora

Table3: Configuration of Private cloud server (Third layer):

Sl No (Private Cloud)	RAM	HDD	PROCESSOR	OPERATING SYSTEM
Instance 1	16 GB	1TB	Octa Core	Fedora22
Instance 2	16 GB	1TB	Octa Core	Fedora22
Instance 3	16 GB	1TB	Hexa Core	Fedora22
Instance 4	16 GB	1TB	Hexa Core	Fedora22

VIII. Numerical Results and Analysis Depending upon the specific hardware infrastructure:

Results for both cases i) In case of data offloading scenario ii)in case of code offloading scenario are presented here. The results show that the power consumption and delay calculation both are optimized in the case of distributive fog computing. The decision-making has gone through according to the five examples of both the code offloading and data offloading cases. According to the result.

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

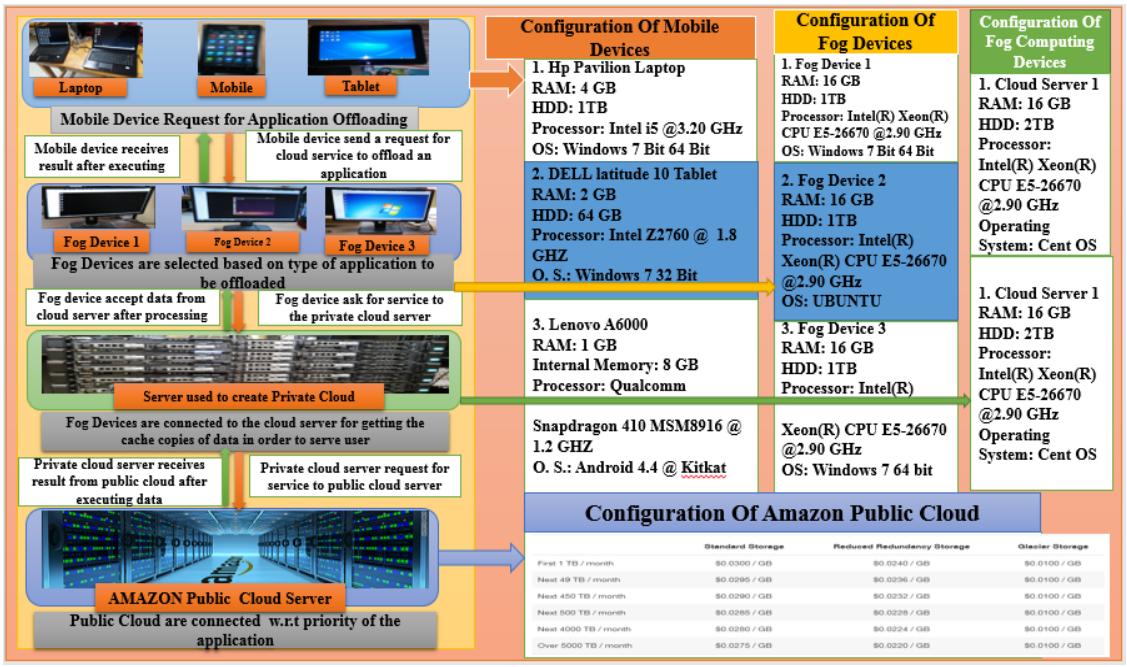


Fig6: Real-time implementation of proposed architecture

Table4: Configuration Table of Public Cloud Server (Fourth Layer)

Sl. No (Public Cloud)	RAM	HDD	PROCESSOR	OPERATING SYSTEM
Instance1	32GB	1TB	HexaCore	Ubuntu
Instance2	32GB	1TB	HexaCore	Ubuntu
Instance3	32GB	1TB	HexaCore	Ubuntu
Instance4	32GB	1TB	HexaCore	Ubuntu

Table5: Experimental Design of fog instance: for the first layer (mobile device to Fog device)

Types of Data	Size of Data	Fog Device (Time)	Fog Device (Power)
Video	16MB	3.245sec	0.299W
	24MB	4.187sec	0.398W
Audio	3MB	2.212sec	0.15W
	6MB	2.255sec	0.23W
Text	18KB	1.136sec	0.130W
	3MB	1.962sec	0.176W

Table6: Experimental Design of private cloud instance: for second layer (mobile device to private cloud instance)

Types of Data	Size of Data	Private cloud instance (Time)	Private cloud instance (Power)
Video	16MB	4.545sec	0.499W
	24MB	5.487sec	0.598W
Audio	3MB	3.112sec	0.25W
	6MB	3.455sec	0.33W
Text	18KB	1.736sec	0.20W
	3MB	2.362sec	0.26W

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

Table7: Experimental Design of Public cloud instance: for third layer mobile device to public cloud instance

Types of Data	Size of Data	Public Cloud instance (Time)	Public Cloud Instance (Power)
Video	16MB	15.638sec	1.69W
	24MB	25.224sec	2.71W
Audio	3MB	11.276sec	0.27W
	6MB	11.129sec	1.148W
Text	18KB	8.066sec	0.887W
	3MB	10.515sec	1.152W

Graphical representation:

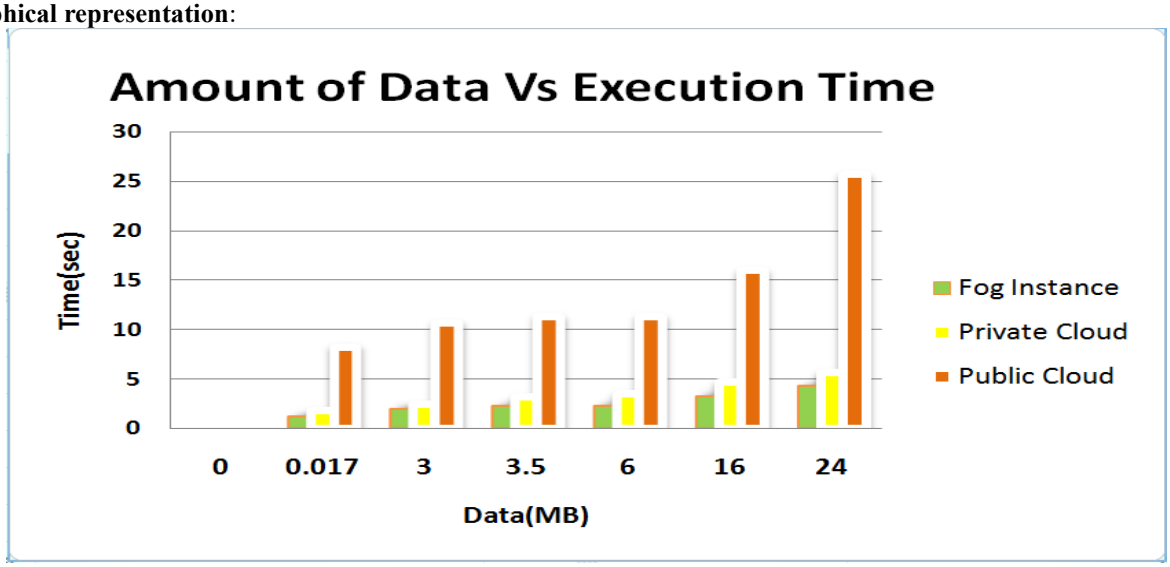
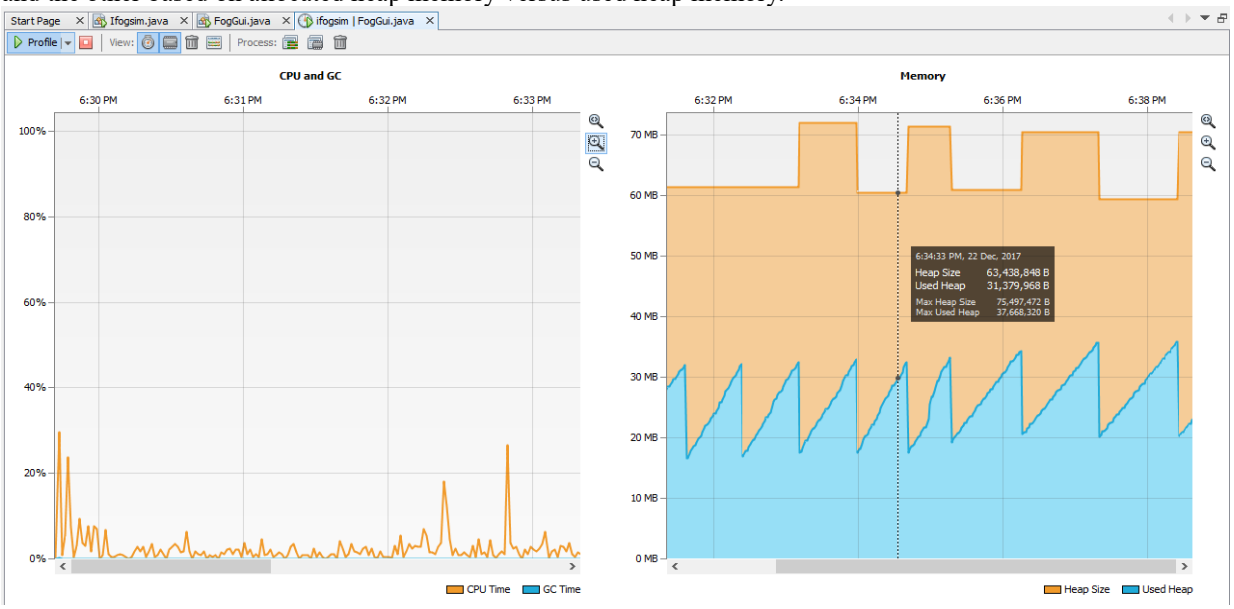


Fig7: Graphical Representation

Using the ifogsim simulator

This newly developed simulator is similar to Cloudsim.[2]. This simulator is specifically designed for Java programming. It allows users to develop research models using an intuitive graphical user interface. The GUI includes actuators, sensors, and fog instances connected by cables [2]. The sensors act like IoT devices found in portable devices and are connected to fog devices through cables. The fog devices are connected to a private cloud device, which in turn is connected to a public cloud server. The key difference between this hardware infrastructure and the simulator is that the simulator has fixed bandwidth and MIPS rates, which are used to generate graphs and calculate execution times. The simulator is also capable of offloading three types of data files. Finally, the simulator generates two graphs: one based on CPU utilization, and the other based on allocated heap memory versus used heap memory.



REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

Fig8: Graphical representation of offloading DATA SET-1 file in ifogsim.

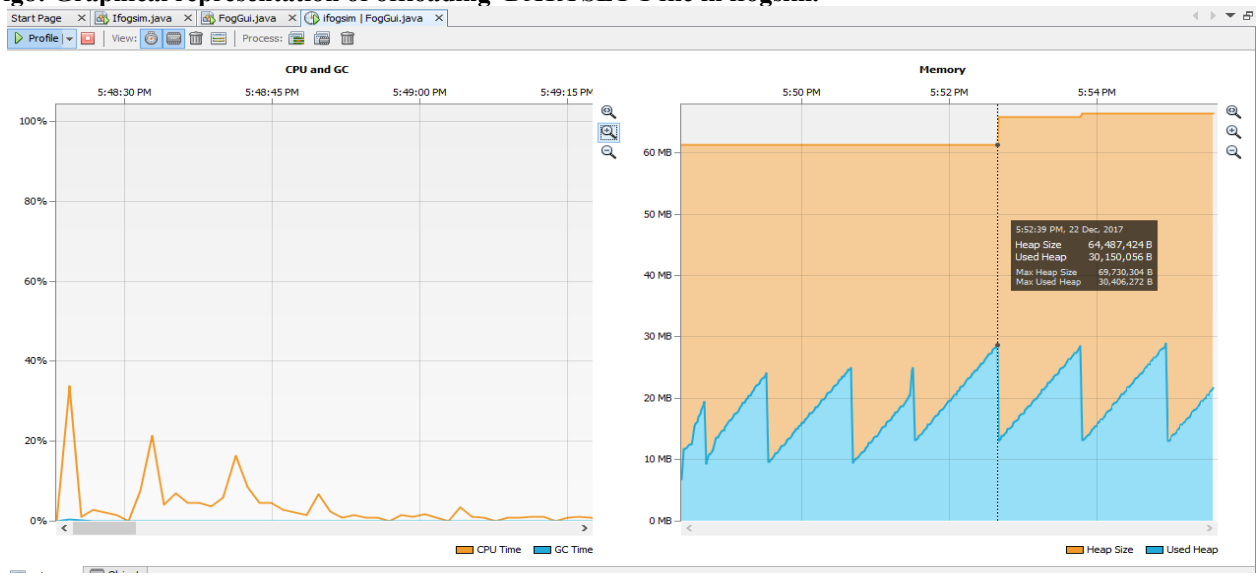


Fig9: Graphical representation of offloading text file DATA SET-2 in ifogsim

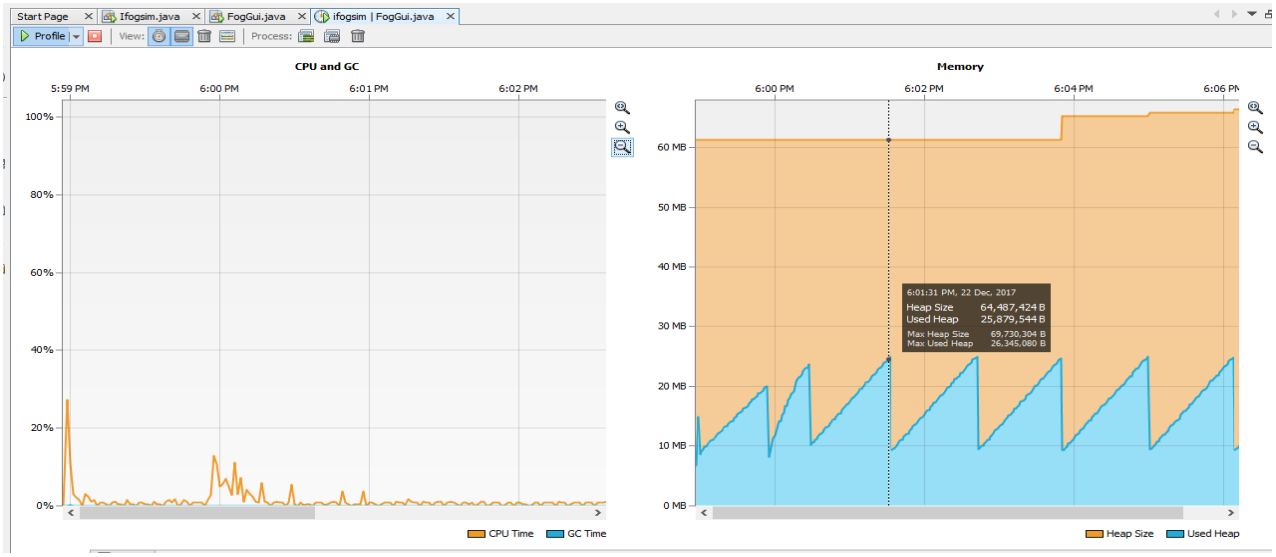


Fig10: Graphical representation of offloading DATA SET-3 file in ifogsim

Table8: Experimental Design using ifogsim of Public cloud instance: for third layer mobile device to Public cloud instance

Types of Data	Size of Data	Public Cloud instance (Time)	Error rate
Video	16MB	15.931sec	1.93%
	24MB	25.224sec	2.693%
Audio	3MB	11.297sec	1.19%
	6MB	11.417sec	1.51%
Text	18KB	8.231sec	2.05%
	3MB	10.68sec	1.62%

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

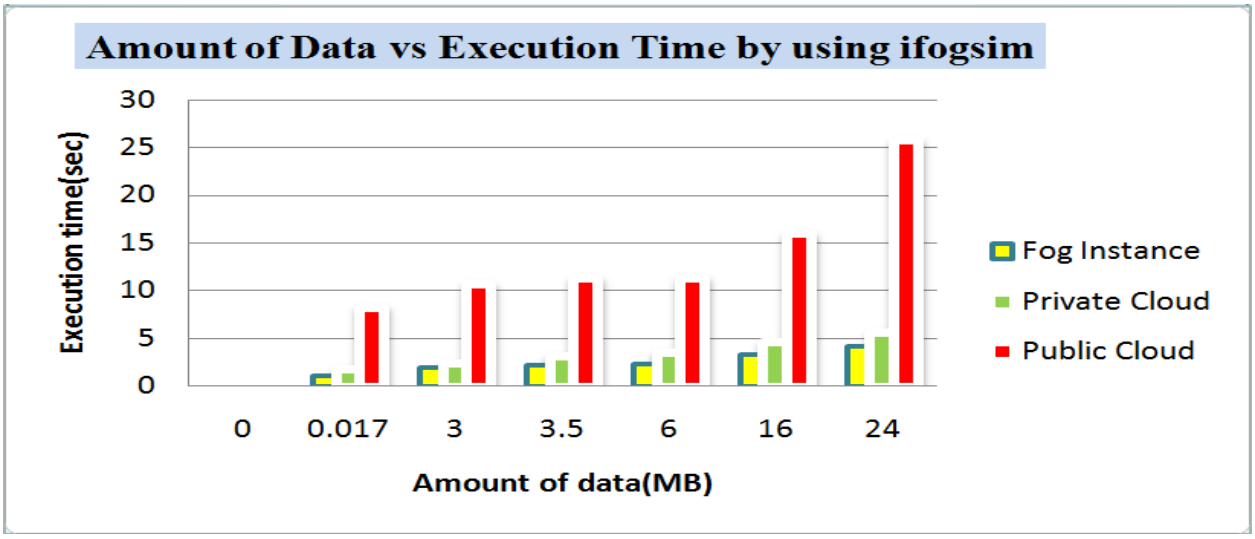


Fig11: Amount of data vs Execution time by using ifogsim

cost matrix of the proposed work,

Let x_{ij} denote the data offloading time from fog devices i the cloud server j. The cost, associated with this transmission/movement, is

Cost * Offloading time = $c_{ij} \times x_{ij}$

The cost of data offloading the commodity from fog devices i to the cloud server j is given by

$\sum_{j=1}^{l}(c_{ij}x_{ij}) = c_{i1}x_{i1} + c_{i2}x_{i2} + \ldots\ldots + c_{il}x_{il}$

The total cost of data/code offloading the commodity from all the fog devices to the cloud server is:

Total cost = $\sum_{i=1}^{k}\sum_{j=1}^{l}(c_{ij}x_{ij})$

To minimize the data offloading cost, the following algorithm will be applied:

Step1: Select the upper left corner cell of the matrix and allocate as much as possible in such a way that either the capacity of the first row is exhausted or the first column of demand or retrieved is satisfied

i.g; $x_{11} = \min(a_1, b_1)$

Step2:

(A) If $b_1 > a_1$, move vertically down to the second row and make the second allocation

$X_{21} = \min(a_2, b_1 - x_{11})$

(B) If $b_1 < a_1$, move horizontally right to the second column and make the second allocation

$X_{12} = \min(a_2 - x_{11}, b_1)$

(C) If $b_1 = a_1$, one can choose any of the following allocations

$X_{21} = \min(a_1 - a_1, b_1) = 0$

OR,

$X_{21} = \min(a_2, b_1 - b_1) = 0$

Step 3: Repeat step1 and step2 while moving down towards the lower right corner of the. matrix.

REDVET - Revista electrónica de Veterinaria - ISSN 1695-7504
Vol 24, No.4 (2023)
http://www.veterinaria.org
Article Received: 28/11/2023 Revised: 18/12/2023 Published: 30/12/2023

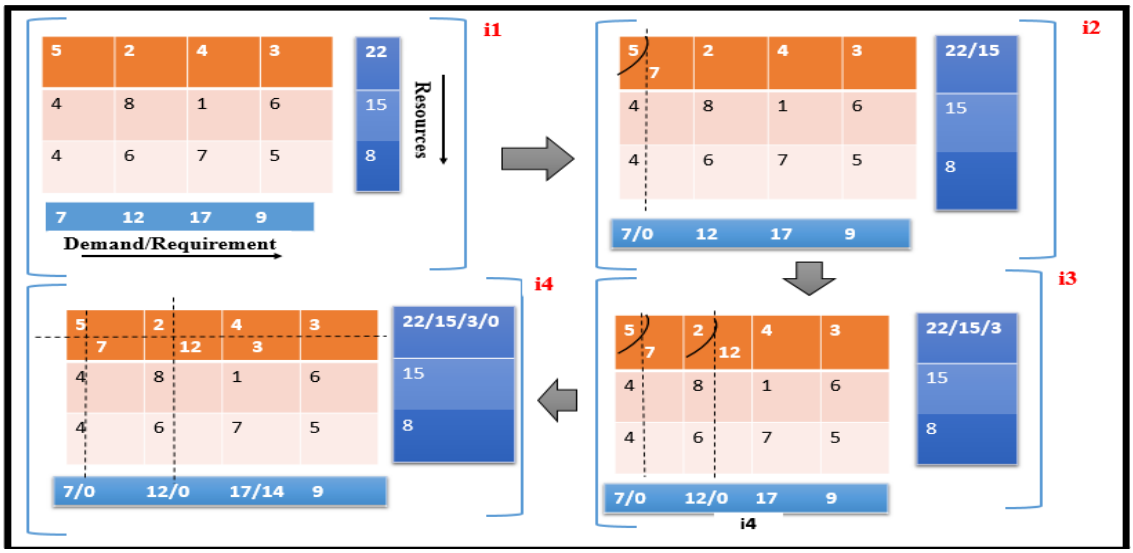


Fig12: Applying Algorithm for some data

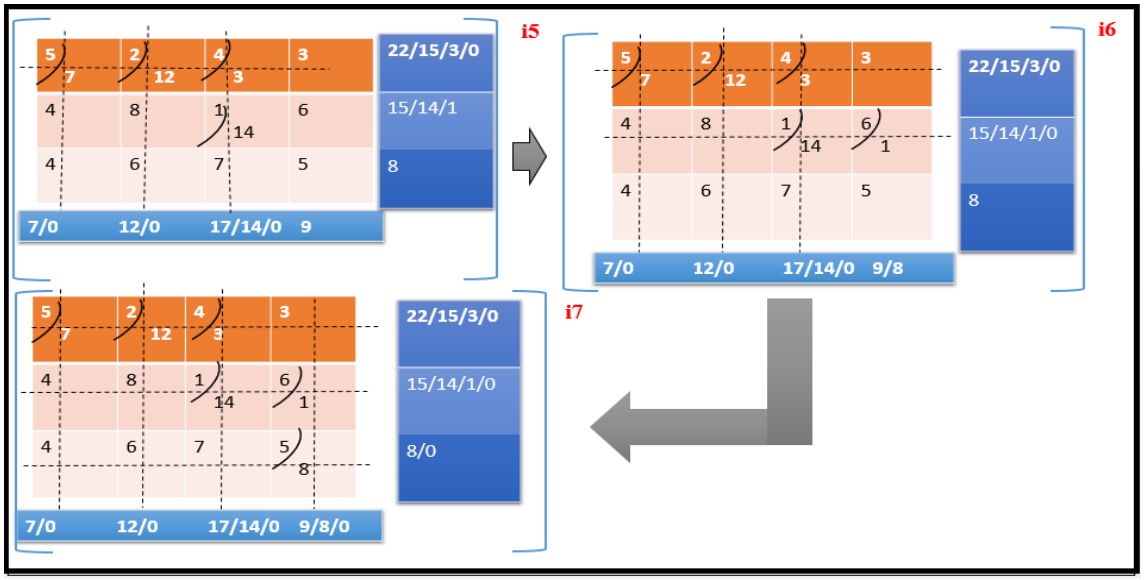


Fig13: Applying Algorithm for some data

So, Total cost = (5*7)+(2*12)+(4*3)+(1*14)+(6*1)+(5*8) = 191 units
We use a specific transportation algorithm. The data of the model include
1. The level of provision at each source and the amount of requirement at each destination.
2. The text below talks about calculating the unit data offloading cost of a commodity from various fog devices to a Cloud Server
. Since there is only one service, a target can receive its claim from more than one source. The objective is to figure out how much should be delivered from each source to each target in a way that minimizes the total transportation cost.
The cost of x_{ij} is defined as cij. The c_{ij} values are determined beforehand when the matrix is formed, and then resources x_{ij} are allocated based on the demand or requirement. The bottom of the matrix represents the demand or requirement matrix, whereas the right side of the matrix represents the resources matrix. In this case, c_{ij}= {5,2,4,3,4,8,1,6,4,6,7,5}, demand or requirement matrix is {7,12,17,9}, and resources matrix is {22,15,8}.
The output of the matrix defines the minimum cost for data offloading.

Output: Total cost = $\sum_{i=1}^{k} \sum_{j=1}^{l} (c_{ij} x_{ij})$

$= c_{11}x_1 + c_{12}x_{12} + \ldots\ldots + c_{1n}x_{1l} +$
$c_{21}x_{21} + c_{22}x_{22} + \ldots\ldots + c_{2n}x_{2l} +$
$+ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots +$
$C_{k1}x_{k1} + c_{k2}x_{k2} + \ldots\ldots + c_{kl}x_{kl}$
Where,

c_{ij}: Cost for that resources which are allocated.
X_{ij}: Allocated resources
i: Row of the matrix
j: Column of the matrix
We will only consider the cost of data offloading as the Quality of Service (QoS) metric, which is dependent on the allocation of resources (x_{ij}) and their cost (c_{ij}). As Fog computing utilizes wireless access to the Internet and brings computation closer to the end user, it is practical to model asymmetric link bandwidth.
A distributed QoS-aware scheduler can enhance application performance, especially when the Distributed Signal Processing (DSP) system is deployed on a network with significant latency and QoS changes. Finally, the simulation and experimental analysis will be concluded with factual evidence. and figure.

IX. Profitability outcome(point-wise):
i)The focus of this paper is on distributed fog computing, where the number of edge devices is increasing daily. Fog devices receive requests from various types of sensors and compute the collected data. The computation is divided into four parts: code offloading, data offloading, and a combination of both. These offloading methods are divided into smaller parts, including propagation delay computation delay, queueing time, and storing time, all of which are individually calculated. The power consumption is calculated by combining all of these factors in equation (13). The concept of fog computing was already proposed in references [1] and [2]. The calculation portion is more prominent and is individually calculated by considering all of these delays. By reducing the failure rate and power consumption, this paper aims to increase the quality of experience. It will help to calculate time and power consumption more accurately than any other fog-related paper. The paper presents three types of offloading: to the fog device, to the private cloud, and the public cloud. Users' mobile devices try to handle requests first. If they fail, the nearest fog device is selected. If the fog device is already involved, the private cloud is selected. If any failure occurs, the public cloud is the final option. A fog manager is present at each step to make decisions. The paper proposes a four-layer architecture, which helps to minimize the failure rate. Our work plan is based on the referenced papers, and it is not possible to propose further architecture and equations that are truly helpful in calculating power consumption and time requirements.

X. Conclusion
We have focused on the emerging paradigm of fog computing, also known as edge computing. Fog computing is an extension of cloud computing. In this paper, we propose a five-layer architecture that differs from the typical centralized approach used in fog computing. We also introduce the concept of delay trade-off and power consumption in the case of a distributed fog computing scenario. Our main contribution is the calculation of two cases: data offloading and code offloading. We divide the problem into sub-problems and execute them according to the decision made by the cloud manager. We have also included simulations and numerical results to demonstrate the efficiency of the newly developed architecture. Our goal is to pioneer a new direction in fog computing, as all previous works in this area have been centralized, whereas our optimization is performed in a distributed manner.

References:
[1] Ruilong Deng, Member, IEEE, Rongxing Lu, Senior Member, IEEE, Chengzhe Lai, Member, IEEE, Tom H. Luan, Member, IEEE, and Hao Liang, Member, IEEE "Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption".
[2] iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge, and Fog computing environments Harshit Gupta1,2 Amir Vahid Dastjerdi1 Soumya K. Ghosh3 Rajkumar Buyya1
[3] Bonomi F, Milito R, Natarajan P, Zhu J. 2014. Fog computing: a platform for the Internet of things and analytics. In: Big Data and Internet of Things: A Roadmap for Smart Environments Springer; 169-186.
[4] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in the big data era," IEEE Network, vol. 28, no. 4, pp. 46–50, 2014.
[5] "A Power and Latency Aware Cloudlet Selection Strategy for Multi-Cloudlet Environment" Anwesha Mukherjee, Student Member, IEEE, Debashis De, Senior Member, IEEE,andDeepsubhraGuha Roy, Student Member, IEEE
[6] N. Kumar, S. Misra, J. Rodrigues, and M. Obaidat, "Coalition games for spatio-temporal big data in Internet of vehicles environment: a comparative analysis," IEEE Internet of Things Journal, vol. 2, no. 4, pp. 310–320, 2015.
[7] Debashis De1,2, Anwesha Mukherjee1 ✉, Anindita Ray1, DeepsubhraGuha Roy1, Suchismita Mukherjee1 Department of Computer Science and Engineering, West Bengal University of Technology, BF-142, Sector-I, Salt Lake, Kolkata 700064, West Bengal, India Department of Physics, University of Western Australia, Perth, Australia "Architecture of green sensor mobile cloud computing"ISSN 2043-6386 Received on 1st April 2015 Revised on 2nd February 2016 Accepted on 15th May 2016 doi: 10.1049/iet-wss.2015.0050
[8] L. Rao, X. Liu, M. D. Ilic, and J. Liu, "Distributed coordination of Internet data centers under multiregional electricity markets," Proceedings of the IEEE, vol. 100, no. 1, pp. 269–282, 2022.

[9] DeepsubhraGuha Roy1 · Debashis De1 · Anwesha Mukherjee1 · Rajkumar Buyya2, "Application-aware cloudlet selection for computation offloading in multi-cloudlet environment"
[10] F. Ahmad and T. Vijaykumar, "Joint optimization of idle and cooling power in data centers while maintaining response time," in ACM SigplanNotices, vol. 45, no. 3, 2020, pp. 243–256.
[11] S. He, J. Chen, X. Li, X. S. Shen, and Y. Sun, "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," IEEE Transactions on Mobile Computing, vol. 13, no. 6, pp. 1268–1282, 2014.
[12] C. Lai, R. Lu, D. Zheng, H. Li, and X. Shen, "Toward secure large-scale machine-to-machine communications in 3GPP networks: challenges and solutions," IEEE Communications Magazine, vol. 53, no. 12, pp. 12–19, 2015.
[13] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches," Wireless Communications and Mobile Computing, vol. 13, no. 18, pp. 1587-1611, 2023.
[14] A. Mukherjee and D. De, "Low Power Offloading Strategy for Femto-Cloud Mobile Network," Engineering Science and Technology, an International Journal, vol. 19, no 1, pp. 260-270, 2016.
[15] J. Li, X. Tan, X. Chen, D. Wong, and F. Xhafa, "OPoR: Enabling Proof of Retrievability in Cloud Computing with Resource-Constrained Devices," IEEE Trans. Cloud Computing, vol. 3, no. 2, pp. 195-205, 2015.
[16] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-based Cloudlets in Mobile Computing," Pervasive Computing, IEEE, vol. 8, no. 4, pp. 14-23, 2019.
[17] Yick, J., Mukherjee, B., Ghosal, D.: 'Wireless sensor network survey', Comput. Netw., 2018, 52, (12),pp. 2292–2330
[18] De, D., Mukherjee, A.: 'Femto-cloud based secure and economic distributed diagnosis and home health care system', J. Med. Imag. Health Inf., 2015, 5, (3), pp. 435–447
[19] Ray, A., De, D.: 'Energy efficient clustering algorithm for multi-hop green wireless sensor network using gateway node', Adv. Sci. Eng. Med., 2013, 5, (11), pp. 1199–1204
[20] Lu, K., Liu, G., Mao, R.: 'Relay node placement based on balancing power consumption in wireless sensor networks', IET Wirel. Sens. Syst., 2019, 1, (1), pp. 1–6
[21] Buyya R, Broberg J, Goscinski AM (2021) Cloud computing: principles and paradigms. Wiley, New York.
[22] Buyya R, Beloglazov A, Abawajy J (2010) Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010), July 12–15 2020, CSREA Press, Las Vegas.