Nat.Volatiles&Essent.Oils,2021;8(5):12882-12892

NVEO
Natural Volatiles &
Essential Oils

A Novel Approach for Minimizing the Latency in Fog Computing

Govind Murari Upadhyay, Shashi Kant Gupta
School of Engineering and Technology, ITM University, Gwalior, India

Abstract
When it comes to today's requirements, the most problematic ones are those that involve dealing with the vast generation of multimedia data from Internet of Things (IoT) devices, which becomes extremely tough to handle if done solely through the cloud. The technology of fog computing emerges as an intelligent solution that operates in a distributed environment, according to the researchers. Latency minimization in e-healthcare is the goal of this article, which is achieved with fog computing technology. Therefore, in IoT multimedia data transfer, it is necessary to lower the delay factors such as transmission time, network time, and computing time because of the rising demand for healthcare multimedia analytics. Fog computing brings processing, storage, and analysis of data closer to the Internet of Things and end-users, hence reducing latency. In this research, a novel Intelligent Multimedia Data Segregation (IMDS) strategy based on Machine Learning (k-fold random forest) is developed in a fog computing environment that segregates multimedia data, as well as a model for calculating total latency, is discussed.
Keywords: Fog Computing,Latency, Multimedia Data Segregation

I. INTRODUCTION

Fog computing is a type of cloud computing that adds additional computation, storage, and networking resources to cloud computing platforms that are located in close proximity to end-user devices (also known as edge computing). The closeness of fog computing resources to end users and their Internet of Things devices promises to give exceptionally low network latencies between end-user devices and the fog computing resources serving them, as well as the ability to process transitory data created by end-user devices locally. There is a lot of activity in the field of fog computing right now, and many researchers are proposing new techniques to create the next-generation fog computing platforms [1]. Scientists working on fog computing, on the other hand, are up against a significant challenge: there is currently no publicly available large-scale general-purpose fog computing platform. To develop viable fog computing platforms, however, they must first have a thorough understanding of the types of applications that will make use of fog computing technologies, as well as the requirements that will be placed on the underlying fog computing platforms. On the other hand, until fog computing platforms are already in existence, few (if any) developers will devote considerable time to developing apps that take advantage of their capabilities. The goal of this study is to break the vicious cycle by investigating a sample selection of fog applications that have either already been deployed or have been recommended for future development.

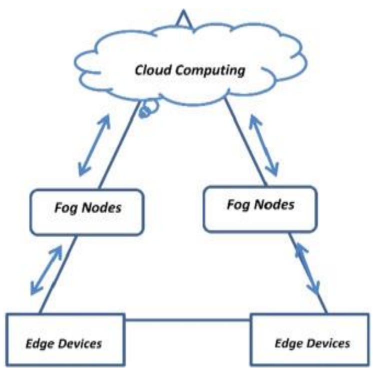

Fig. 1. Basic Fog computing model[11]

We carefully picked 30 real-world or suggested applications that represent a diverse spectrum of possible uses for future fog computing platforms, which we described in detail below. We next use this collection of reference applications to address a number of critical questions about the functional and non-functional needs that should be present in a general-purpose for g computing platform. This paper demonstrates that fog applications and their corresponding needs are extremely diverse, and it highlights the specific features that fog platform builders may want to incorporate into their systems in order to serve specific categories of applications. The following is the structure of this article. Providing a general overview of fog computing platforms and applications is covered in Section 2. We outline our process for selecting a representative group of reference applications in Section 3. The following section, Section 4, examines the demands that these applications place on cloud computing platforms, and the final section, Section 5, brings this essay to a close.

II. RELATED WORK

I. B. Lahmar et al[2] This difficulty was met with the introduction of fog computing. Fog computing was designed to increase the processing capabilities of Cloud computing while also improving the overall quality of service (QoS) for latency-critical applications. When compared to Cloud datacenters, Fog devices are very dynamic and heterogeneous in their configurations. One of the most difficult aspects of deploying IoT applications in a fog environment is dealing with the resource allocation problem. This research intends to examine recent work related to resource allocation in fog environments by employing a systematic mapping study technique in order to provide a comprehensive overview of what has been investigated in the area of resource allocation in fog environments.

M. V. Prakash et al[3]The suggested technique is compared to existing methods in terms of several performance indicators, and the simulation of the proposed method is carried out between them. The results demonstrate that the suggested machine learning approach in the fog IoT environment is more energy efficient than the other ways when compared to the other methods.

S. K. Battula, et al[4] Effective resource monitoring strategies are required in order to increase the overall performance of fog computing and to optimise the use of available resources. Because of this, we present a technique called support and confidence based (SCB), which is designed to optimise the resource use in the resource monitoring service.

H. Wadhwa et al[5]This paper describes the notion of fog computing, the architecture of fog computing, and the application that has been built. It also discusses resource provisioning approaches that can be used to identify instances of overutilization of fog nodes. Additionally, multiple scheduling terminology have been considered in relation to numerous metrics in addition to resource use. The purpose of this survey is to gain a better understanding of how fog computing can be used to improve the current smart healthcare system in use.

E. S. Gama, et al[6]One of the primary objectives is to build and evaluate an extremely stable and high-quality multi-tier services architecture that can be implemented in Smart City contexts. This is accomplished through the introduction of a set of video streaming services in the fog/cloud computing environment, as well as the proposition of how these services may be utilised to improve the Quality of Experience (QoE) for end-users.

D. Gonçalves, et al[7] we'll explain how we enhanced our simulator, named MobFogSim, to include dynamic network slicing, and how MobFogSim can be used for capacity planning and service management for mobile fog services. We also provide the results of an experimental study of the

influence of dynamic network slicing on container migration for the purpose of supporting mobile users in a fog environment. Dynamic network slicing, according to the findings, can improve resource utilisation as well as migration performance in the fog.
Z. Rejiba, et al[8] The VFC scenario was simulated using realistic vehicular movement traces, which allowed us to evaluate this approach. According to the collected data, our proposed strategy enhances the learning performance when compared to the condition in which no advice is leveraged.
Y. Wang, et al[9] A distributed generalised diffusion approach is developed for the "battle cloud-fog" network system in order to reduce latency while simultaneously improving stability and survival. The simulation results show that the load balancing strategy based on the generalised diffusion algorithm can reduce task response latency while also supporting the efficient processing of battlefield information effectively, making it a good fit for the "combat cloud-fog" network architecture described in this study.

High Latency in a Cloud Environment:

Factors Affecting It It is possible to store and process a large amount of data in the cloud. A bottleneck has been created due to the volume of data and the resulting complex latency for time-sensitive systems due to the rise of IoT computing platforms, which have generated an abundance of data and processed it over a centralised computing model that lacks sufficient network bandwidth. Due to the increasing complexity of latency in decentralised platforms like the Internet of Things, heterogeneous services and applications must be routed through the cloud. IoT must use cloud resources and overcome high latency in order to run time-sensitive applications and scale up computational resources [19]. Some of the factors that contribute to cloud computing's high latency are as follows:

Enterprise datacenters, which house all of an organization's data and infrastructure, are dispersed around the world via distributed computing. Latency problems are becoming increasingly complex and difficult to resolve. Cloud latency is highly complex because of the wide range of data transmissions that occur across the internet.

Cloud latency suffers as a result of increased complexity caused by long data transmission delays in a virtualized environment.

Ping and trace-route are diagnostic tools used to measure transit times and show the routing path. There is a lack of measuring tools. Internet Control Message Protocol (ICMP), on the other hand, is not used by dynamic modern web applications, which instead use the Hypertext Transfer Protocol (HTTP) and FTP (FTP). It is essential to prioritise network traffic and service quality. Covenants between cloud participants, such as SLAs and QoS, aim to improve the internet computing experience. Different cloud applications and services have varying tolerances for network latency.

In a distributed cloud environment, a customer has no idea where the data centres housing the cloud services are located. Cloud latency increases when searching nearer or farthest data centres, which complicates the search. Data centres that are close to the source of origination are more efficient at searching.

• Network Traffic Workload: Internet surfing isn't always smooth; it's more like a wave that comes and goes. For example, during peak or prime hours, network lagging is more common than during other hours. The high altitude is also a factor in the high latency. Because many people are online and accessing online real-time content (e.g. gaming and video streaming) simultaneously, congestion in the network results in high latency.

III. Proposed methodology

As a locally operating mechanism, fog computing and the Internet of Things-based health and tactical analytic monitoring model in sports have low latency, as does fog computing in general. As a result, timely intervention, which is critical in the field of health, can be delivered. In the case of technical challenges, it is also critical that data processing and response messages are completed as rapidly as possible. According to research conducted by an artificial intelligence expert at the STATS analytics organisation, it is critical to take action fast in a tactically bad situation: identify a specific circumstance that has the potential to disturb the opponent, providing you, for example, a 30-second window during which the opponent is disorganised. Because the data is processed on fog servers that are not connected to the Internet, the time required to relay the response is minimal. Additionally, the fog task management algorithm, which is based on process priority, reduces reaction time and costs even further. On the fog servers, data is encrypted before being processed or stored on the remote cloud server, regardless of how long the response delay time takes.

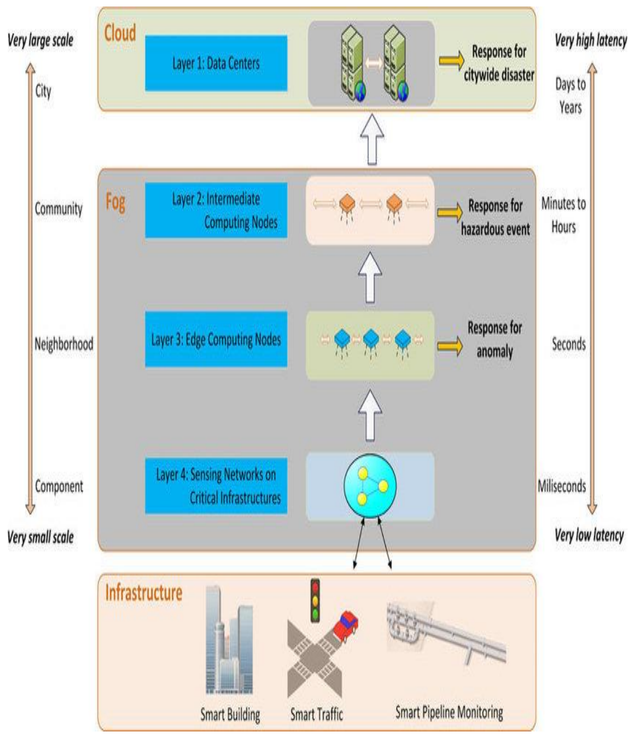

Figure 2: Four-layer Fog Computing architectureB. Tang et al.,[12]

It has recently been published by the OpenFog Consortium in its reference architecture paper [2] that principles for the design of fogs should be prioritised in terms of security, scalability, and dependability of fog nodes [3, 4]. One intriguing component detailed in the reference design is the deployment of fog and cloud nodes in several levels, which is described in detail in the reference architecture. The system model under consideration in this research is composed of three layers. Cloud and fog architecture are discussed in detail. The fog node is described as a cloudlet [3], which is capable of hosting virtual computers at a single hop distance from the ultimate users, whereas the primary node has a large number of physical servers. The third layer is made up of user devices, which are all connected to the fog via a direct connection. This system receives requests in the form of modularized applications, which are then processed by the system. Each application consists of one or more activities that are completed with the help of sensors and actuators. Sensors collect data from the environment or from users, whereas actuators interact with people and carry out actions on the data they collect. The tasks of an application are responsible for performing specialised processing for the application and receiving data from a sensor or from another task in the application. By creating a virtual machine for each activity, resources may be allotted to each task in the system. There are two possible locations for the instantiation: a fog node or a cloud node.

Proposed algorithm step

The RF classification algorithm is employed in two phases. First, the RF algorithm harvests subsamples from the original samples using the bootstrap resampling approach and generates decision trees for each sample. Second, the method classifies the decision trees and performs a simple vote, with the largest vote of the classification as the final outcome of the prediction. The RF algorithm always includes three steps as follows:

(1) Select the workout set. Use the bootstrap random sampling approach to retrieve K training sets from the original dataset (M properties), with the size of each training set the same as that of the original training set.

(2) Build the RF model. Create a classification regression tree for each of the bootstrap training sets to produce K decision trees to form a “forest”; these trees are not trimmed. Looking at the growth of each tree, this strategy does not choose the best features as internal nodes for branches but rather the branching process is a random selection of m⩽M of all features.

(3) Create simple voting. Since the training process of each decision tree is independent, the training of the random forests can function in parallel, which considerably enhances efficiency. The RF can be formed by joining K decision trees trained in the same way. When classifying the input samples, the results depend on the simple voting of the output of each decision tree. The RF algorithm selects the samples by generating a succession of independent and dispersed decision trees and decides the final category of the sample according to each decision tree.

Algorithm 3 is a mapping algorithm.

1: while p PATHS do a cross-section of all paths
2: the location
a list of devices = a list of devices
3: while the fog device d p does its thing
4. while module w is being executed 5. if all of the predec. of w are placed 6. add w to the list of modules
if 7 is true, then 8 is true, and 9 is true while module is in place.
List do 10: if place on d p then 11: d:= Device if place on d p
12. Place d on device 13. End if

There are two types of tasks considered in this work: tasks for which virtual machines (VMs) must be created in the fog and tasks for which virtual machines (VMs) can be instantiated either in the cloud or in the fog. The former are time-sensitive operations that require a quick response from the user, whilst the latter are jobs that require a more flexible reaction time from the user. It is possible for applications to perform numerous functions and to belong to distinct classes at the same time. This can result, for example, in a request having part of its virtual machines (VMs) provided in the cloud and another part in the fog node. The cloud and fog contain agents that determine which physical server should be used to host a virtual machine when it is created in the cloud. If the requested resource cannot be found, the request is denied.iFogSim The iFogSim simulator simulates IoT and fog environments in order to assess the impact of "latency, network congestion, energy consumption, and cost" on these environments. In order to make use of the event simulation capabilities of Cloudsim, we developed iFogsim. The Cloudsim core is in charge of handling events that occur between Fog instances, and it is also in charge of processing those events.
The System Specification Using the PVFOG Simulator, the three models in question can be tested and compared. An application's efficiency is evaluated based on five different factors: CPU utilisation, RAM utilisation, latency, packet throughput, and available bandwidth. The latency of a request or response is represented by three models, each of which represents the amount of time it takes for the request or responding to be processed.

$$L = (Request\ generation\ time\ + Packet\ transaction\ time\ + \\ Network\ delay + Request\ processing\ time\ + Response\ sent\ back\ time)$$

According to the experimental results, the Cloud-Fog with MQTT model has a lower average latency than both the Cloud-only and Cloud-Fog models on a per-second basis. Fog nodes make intermediate decisions and communicate those decisions to MQTT clients in order to reduce the amount of time that clients must wait for their requests to be processed by the Cloud server. This reduces the amount of time that clients must wait for their requests to be processed by the Cloud server. As a result, the proposed method takes 46 percent less time to complete than the traditional Cloud-only approach.
Table 1:Latency on average is measured in milliseconds.

Appraoch	Number of Node	Average Latency in MS
Cloud	10	5.321
Cloud and fog	10	4.23
Proposed	10	2.45
Cloud	20	6.43
Cloud and fog	20	5.51
Proposed	20	2.37
Cloud	30	7.43
Cloud and fog	30	6.60
Proposed	30	4.34
Cloud	40	8.92

Cloud and fog	40	7.03
Proposed	40	3.42
Cloud	50	9.34
Cloud and fog	50	7.86
Proposed	50	4.32

Bandwidth

This rate represents the percentage of available bandwidth that is used by a specific device in comparison to the total amount of bandwidth allotted to three different and identical devices. In order to calculate bandwidth, the following formula must be used:

$$\boldsymbol{BU = \frac{Actual\ Bandwith\ utilized}{Allocated\ Bandwidth} * 100}$$

Appraoch	Number of Node	Average Latency in MS
Cloud	10	4.65
Cloud and fog	10	4.32
Proposed	10	4.01
Cloud	20	5.92
Cloud and fog	20	5.51
Proposed	20	5.02
Cloud	30	6.88
Cloud and fog	30	6.44
Proposed	30	6.02
Cloud	40	8.92
Cloud and fog	40	8.49
Proposed	40	810
Cloud	50	9.90
Cloud and fog	50	9.55
Proposed	50	9.11

Packets are delivered to the Cloud data centre.For the three different models listed in Table 2 the packet transmitted refers to the total number of packets sent and received by a particular device; this is illustrated graphically results that the number of packets sent by IoT devices to the Cloud data centre using Remote Broker in Fog node using MQTT is less when compared to the other two models when using Remote Broker in Fog node.By contrast, when using the proposed method, the number of packets sent is nearly 29 percent lower than when using the Cloud-only model. Because the Remote

Broker handles requests, the number of packets that must be transmitted to the Cloud server is reduced as a result of this.

When comparing the Cloud-Fog with MQTT model to the Cloud only and Cloud-Fog models, the experimental results show that the average bandwidth utilised is less for the Cloud-Fog with MQTT model.

In the proposed model, the requests are processed within the Fog node itself, resulting in a reduction in bandwidth consumption.

III. Conclusion

The most difficult task in e-healthcare is classification of health data and minimising latency. In order to reduce latency in e-healthcare, fog computing technology is required. With machine learning (k-fold random forest) in the fog computing environment, we introduced a novel intelligent multimedia data segregation system. It indicates a reduction in high latency characteristics such as transmission, network, and processing. The suggested paradigm improves e-healthcare service quality and is applicable to diverse networks. QoS considers network latency and usage patterns. Low latency and low network utilisation increase QoS. Using 5G as a greater internet connectivity can increase the quality of e-healthcare services and latency for sensitive data. The fog model can be used to construct a smart healthcare system in a separate facility.

Reference

[1]. B. Lahmar and K. Boukadi, "Resource Allocation in Fog Computing: A Systematic Mapping Study," 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), 2020, pp. 86-93, doi: 10.1109/FMEC49853.2020.9144705.

[2]. M. V. Prakash, V. Porkodi, S. Rajanarayanan, M. Khan, B. F. Ibrahim and M. Sivaram, "Improved Conservation of Energy in Fog IOT Services Using Machine Learning Model," 2020 International Conference on Computing and Information Technology (ICCIT-1441), 2020, pp. 1-4, doi: 10.1109/ICCIT-144147971.2020.9213719.

[3]. S. K. Battula, S. Garg, J. Montgomery and B. Kang, "An Efficient Resource Monitoring Service for Fog Computing Environments," in IEEE Transactions on Services Computing, vol. 13, no. 4, pp. 709-722, 1 July-Aug. 2020, doi: 10.1109/TSC.2019.2962682.

[4]. H. Wadhwa and R. Aron, "Fog Computing with the Integration of Internet of Things: Architecture, Applications and Future Directions," 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), 2018, pp. 987-994, doi: 10.1109/BDCloud.2018.00144.

[5]. E. S. Gama, R. Immich and L. F. Bittencourt, "Towards a Multi-Tier Fog/Cloud Architecture for Video Streaming," 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), 2018, pp. 13-14, doi: 10.1109/UCC-Companion.2018.00022.

[6]. D. Gonçalves, C. Puliafito, E. Mingozzi, O. Rana, L. Bittencourt and E. Madeira, "Dynamic Network Slicing in Fog Computing for Mobile Users in MobFogSim," 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC), 2020, pp. 237-246, doi: 10.1109/UCC48980.2020.00042.

[7]. Z. Rejiba, X. Masip-Bruin and E. Marín-Tordera, "Computation Task Assignment in Vehicular Fog Computing: A Learning Approach via Neighbor Advice," 2019 IEEE 18th International

Symposium on Network Computing and Applications (NCA), 2019, pp. 1-5, doi: 10.1109/NCA.2019.8935033.
[8]. Y. Wang, Z. Ren, H. Zhang, X. Hou and Y. Xiao, "“Combat Cloud-Fog” Network Architecture for Internet of Battlefield Things and Load Balancing Technology," 2018 IEEE International Conference on Smart Internet of Things (SmartIoT), 2018, pp. 263-268, doi: 10.1109/SmartIoT.2018.00054.
[9]. Y. Wang, Z. Ren, H. Zhang, X. Hou and Y. Xiao, "“Combat Cloud-Fog” Network Architecture for Internet of Battlefield Things and Load Balancing Technology," 2018 IEEE International Conference on Smart Internet of Things (SmartIoT), 2018, pp. 263-268, doi: 10.1109/SmartIoT.2018.00054.
[10]. D. Gonçalves, K. Velasquez, M. Curado, L. Bittencourt and E. Madeira, "Proactive Virtual Machine Migration in Fog Environments," 2018 IEEE Symposium on Computers and Communications (ISCC), 2018, pp. 00742-00745, doi: 10.1109/ISCC.2018.8538655.
[11]. H., Sabireen, and Neelanarayanan V. “A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges.” ICT Express, vol. 7, no. 2, June 2021, pp. 162–176, 10.1016/j.icte.2021.05.004. Accessed 13 Aug. 2021.
[12]. B. Tang et al., "Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities," in IEEE Transactions on Industrial Informatics, vol. 13, no. 5, pp. 2140-2150, Oct. 2017, doi: 10.1109/TII.2017.2679740.
[13]. Anand Singh Rajawat, Pradeep Bedi, S. B. Goyal, Piyush Kumar Shukla, Atef Zaguia, Aakriti Jain, Mohammad Monirujjaman Khan, "Reformist Framework for Improving Human Security for Mobile Robots in Industry 4.0", Mobile Information Systems, vol. 2021, Article ID 4744220, 10 pages, 2021. https://doi.org/10.1155/2021/4744220
[14]. F. H. Rahman, S. H. S. Newaz, T. W. Au, W. S. Suhaili and G. M. Lee, "Off-Street Vehicular Fog for Catering Applications in 5G/B5G: A Trust-Based Task Mapping Solution and Open Research Issues," in IEEE Access, vol. 8, pp. 117218-117235, 2020, doi: 10.1109/ACCESS.2020.3004738.
[15]. Z. Rezazadeh, M. Rezaei and M. Nickray, "LAMP: A Hybrid Fog-Cloud Latency-Aware Module Placement Algorithm for IoT Applications," 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), 2019, pp. 845-850, doi: 10.1109/KBEI.2019.8734958.
[16]. V. Jain and B. Kumar, "Optimal Task Offloading and Resource Allotment Towards Fog-Cloud Architecture," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 233-238, doi: 10.1109/Confluence51648.2021.9377033.
[17]. Rajawat, Anand Singh, et al. “Fusion Protocol for Improving Coverage and Connectivity WSNs.” IET Wireless Sensor Systems, vol. 11, no. 4, 16 Mar. 2021, pp. 161–168, 10.1049/wss2.12018.
[18]. V. Jain and B. Kumar, "Optimal Task Offloading and Resource Allotment Towards Fog-Cloud Architecture," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 233-238, doi: 10.1109/Confluence51648.2021.9377033.
[19]. N. Godinho, H. Silva, M. Curado and L. Paquete, "Energy and Latency-aware Resource Reconfiguration in Fog Environments," 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), 2020, pp. 1-8, doi: 10.1109/NCA51143.2020.9306711.
[20]. J. Akram, Z. Najam and A. Rafi, "Efficient Resource Utilization in Cloud-Fog Environment Integrated with Smart Grids," 2018 International Conference on Frontiers of Information Technology (FIT), 2018, pp. 188-193, doi: 10.1109/FIT.2018.00040.

[21].S. -S. Lee and S. Lee, "Resource Allocation for Vehicular Fog Computing Using Reinforcement Learning Combined With Heuristic Information," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10450-10464, Oct. 2020, doi: 10.1109/JIOT.2020.2996213.