A Comprehensive Survey on QoS-aware Service Placement Algorithm on Fog Cluster in an Edge Computing Environment

Dhananjaya M K [#1], Dr.Kalpana Sharma[*2], Dr. Amit Kumar Chaturvedi [*3]

#1Ph.D. Scholar, Computer Sc Dept.,Bhagwant University, Ajmer

**2Assistant Prof. Computer Sc Dept.,Bhagwant University, Ajmer*

**3Assistant Prof. MCA Dept., Govt. Engineering College, Ajmer*

#[1]dhanunitte@gmail.com, *[2]kalpanasharma56@gmail.com, *[3]amit0581gmail.com

Abstract: *Cloud computing comes with several inherent capabilities such as scalability, on-demand resource allocation, reduced management efforts, flexible pricing model and service provisioning.The distance between the cloud and the end devices might be an issue for latency sensitive applications such as disaster management and content delivery applications.The fog being closer to the end user is more vulnerable than the cloud.So, Fog is "Cloud closer to ground" and in the fog computing, processing of some application components can take place at the edge of the network while others can happen in the cloud.Fog computing is an epitome to enable provisioning resources and services beyond the cloud, at the edge of the network, and nearer to end devices.For this reason, in this paper, planned for to proposingarchitecture for Quality of Service (QoS) aware fog service provisioning, which allows scheduling the execution of IoT applications tasks on a cluster of fog nodes.Next, step is to analyzetwo fog-based frameworks, FogPlan and FogOffload that can improve the QoS in terms of the reduced service delay. This paper also determines the impact of those security issues and possible solutions, providing future security relevant directions to those responsible for designing, developing, and maintaining Fog system. Results demonstrate that theproposed algorithm achieves significant improvement in terms of service latency and execution cost compared to simulator's built-in policies.*

Keywords: **Cloud Computing, Internet of Things (IoT),Fog computing, Quality of Service (QoS), FogPlan, and FogOffload.**

1. INTRODUCTION

The recent advances in the Internet of Things (IoT), Big Data, and Machine Learning (ML) have contributed to the rise of a growing number of complex and intelligent applications.Examples of such applications are real-time disease detection, self-driving vehicles, drone package delivery, and smart manufacturing. These emerging applications are often real time, data-intensive, and delay-sensitive, and ensuring Quality of Service (QoS) for these applications is a challenge. Fog computing is seen as one promising solution for providing QoS for these applications, as it puts compute, storage, and networking resources closer to the user.However, the fog provides some advantages,

such as low latency, by allowing processing to take place at the network edge, near the end devices, called fog nodes, being at the edge of the network, pose several security threats and the ability to enable processing at specific locations[1].Two novel schemes, ResiliNet and deepFogGuard, were proposed for improving the QoS of deep learning applications that are deployed over networkdevices, such as fog nodes and edge devices. Specifically, the two schemes are for improving the failure-resiliency of inference in distributed neural network in the presence of physical node failures. deep Fog Guard's design is based on the concept of skip connections, whereas, the ResiliNet uses a novel technique, Next, we propose two fog-based frameworks, FogPlan and Fogoffload, that can improve the QoS in terms of the reduced service delay. FogPlan is a planning framework, for the dynamic deployment or release of application services on fog nodes, in order to meet the low latency and the QoS requirements of applications while minimizing the cost. Fogoffload is a delay-minimizing collaboration and offloading policy for fog-capable devices that aims to reduce the service delay of IoT applications [2]. These four frameworks are designed to improve the QoS in emerging IoT and ML applications through Fog Computing. For contemporary cloud-based systems, all service requests and resource demands areanalyzed and processed with in the data centers (DCs).However, with the steeprise in the number of Internet-connected devices and in the light of the emerging technology of the Internet of Things (IoT), the amount of data to be handled by the cloud DCs is paramount.

Cloud computing, in recent years, has added a new dimension to the traditional means of computation, data storage, service provisioning. One of the fast developing trends in intelligent data analysis is deep neural networks (DNNs). They offer a wide range of machine learning (ML) methods that often outperforms the state-of-the-art models. To be effectively used, deep models have special requirements such as on high computing resources or a large amount of data examples.

The remainder of the paper is organized as follows: Section 2 discusses the related work and related Surveys. Section 4 presents service placement algorithm and Learning Repository Fog-Cloud approach.

2. Related work

Cloud computing is a common approach to address low performance issues in IoT devices. The proposed work simulated the data stream by making a connection with the fog nodes. In order to save energy and communicate as few packets a possible, the updated parameters of the learned model at the sensor devices were communicated in longer time intervals to the FC system. In general, available resources off fog node devices are uneven in the real world for many reasons, such as in coming data size, establishment time, and network infrastructure. Nonetheless, when all fog nodes work their best according to each available resource, the burdens of cloud and network traffic are reduced, and more DL application can be accommodated in an FC environment. Hence, we intended to treat one of the main challenges for implementing DL in FC system, the decision of optimal layers despite uneven available resource in fog nodes.

- **Mobility of end-users in Fog computing**

It provides reduced latency and cooperation in annulling or lessening traffic over-crowding in the core network. Nevertheless, this might be more complex to handle each end-user's computation priorities, Quality of Service (QoS) and prioritization of low delay applications. In the traditional scheduling mechanism, the fog computing server is to render the offloading priority order towards the end-users [3]. It relies on the users 'different local computing data, channel gains and latency demands. In case of multiusers fog computing systems, the static scheduling strategy cannot be directly utilized with mobility because of dynamic environments.

Fog computing servers can reserve some dedicated computational resources for mobile users to deliver consistent computing service and guarantee QoS for latency sensitive users. For instance, the end-user is watching a video content and roams from a source fog-enabled access point [4].

- **QoS-Aware Service Provisioning in Fog Computing**

The existing proposed techniques in fog computing lacks the adaptive and intelligent behavior. They only consider the direct communication of end devices preferably with closest fog nodes. Further, it is quite complex to implement adaptive and intelligent-learning-based task scheduling in this kind of architecture. Subsequently, we cannot take the real advantage of heterogeneity of fog nodes surrounded by IoT devices. In order to facilitate intelligent and adaptive task scheduling policies in this dynamic and heterogeneous environment, we need a smart layer (Aazam and Huh,2014) between end devices and fog nodes that should have three main capabilities[12]: 1) the ability to define whether the incoming request should be served by a cloud or a fog node, 2) the capability to schedule the incoming task to most appropriate fog node among the available fog devices, and 3) and for efficient task scheduling this layer should have the functionality that extensively implements adaptive and intelligent learning-based task scheduling. Here, proposed a QoS-Aware approach (referred as the Learning Repository Fog-Cloud - LRFC).The proposed service has been provisioned at multiple geographically distributed gateways deployed among IoE devices and their corresponding Fog nodes proximity, Consequently, making our proposed scheme highly scalable, and thus reliving potential performance bottlenecks, Finally, the proposed deployment model significantly suits nearly all existing and futuristic environments(i.e., IoT, IoE, smart X(smart city, smart grid, smart building, smart forest etc.)

- **Task scheduling in fog-cloud environment**

In (Bitam et al., 2017), the authors proposed a bio-inspired optimization approach (i.e., Bees Life Algorithm (BLA)), seeking to address the job scheduling challenge in a fog computing environment.The approach is based on the optimized distribution of a set of tasks among fog nodes to deal with user' excessive requests to computational resources[13]. This approach also seeks to minimize energy consumption and CPU execution time. In a different work, the authors of (Intharawijitr et al., 2016) proposed three different strategies for optimum resource utilization. Firstly, a random

methodology is used to select the fog nodes to execute tasks upon arrival. Secondly, the focus will be on the lowest latency fog devices. Finally, the fog resources having the maximum available capacity should be primarily considered. The three proposal policies were then evaluated using a mathematical model [14]. We propose a hybrid approach based on the idea of both rule-based learning and case-based reasoning to produce efficient results.

3. Model and Methodology

Suarez et al. propose Foglets, a programming model that facilitates distributed programming across fog nodes. Foglets provides APIs for spatio-temporal data abstraction for storing and retrieving application-generated data on the local nodes. Through the Foglets API, Foglets API, Foglets processes are set for a certain geospatial region and Foglets manages the application components on the Fog nodes. Foglets is implemented through container-based visualization. The Foglets API takes into account QoS and load balancing when migrating persistent (stateful) data between fog nodes. As a future direction, one could consider other QoS attributes and include cost information to get a richer classification of eligible fog deployments.

- **Service provisioning**

The study in focuses on dynamic service provisioning in edge clouds from a theoretical perspective. The model in the study captures the limited capacity of fog nodes, the unknown arrival process of requests, the cost of forwarding requests to the remote cloud, and the cost of on loading a new service on a fog node. In a simulation study, the authors of suggest a conceptual framework for fog resource provisioning [5]. They introduce the concept of “fog cell,” which is a software component running on fog devices that controls and monitors a particular group of IoT devices. Using this and other related concepts, they model orchestration of IoT devices using a hierarchical cloud/fog resource control and provide a suitable resource provisioning solution for distributing tasks among them.

4. Proposed Algorithms

Algorithm 1: Service Placement Algorithm

Our proposed algorithm runs on the controller component of each fog control node. In order to deal with the issue of service placement in the fog-cloud computing environment, in this work, we propose an efficient heuristic algorithm called the Most Delay-sensitive Application First (MDAF). It is worth noting that our proposed algorithm runs on the controller component of each fog control node [6]. The main idea behind MDAF is that the delay-sensitive application services are placed on resources that are closer to the clients as much as possible. Based on the assumed architecture, each IoT device is located inside a colony and is directly connected to a specific fog cell. The

MDAF algorithm is run there to place the processing services of each application on appropriate computational devices.

The main purpose of this efficient heuristic algorithm is proposed to solve the service placement problem in fog cloud computing environments. This algorithm places the most delay-sensitive application services as closer as possible to the IoT devices.The performance of this algorithm based with two baselines, Edge-ward and Cloud-only. Unlike the baselines, the proposed algorithm does not violate deadlines of applications. More specifically, the proposed approach reduces the execution cost by about 37% and 62% compared to Edge-ward and Cloud-only policies, respectively [7].

In this work, we assume that processing services can run on one of the three computational devices F, N and C. The pseudo code of the proposed algorithm is mentioned in Service Placement algorithm.

Algorithm 1: Service Placement Algorithm

1. Sort application list A in ascending order of deadline.
2. Sort Computational devices in the list D in order of closeness to the client.
3. for each application Ai in list A do
4. Curdev = F;
5. for each processing service in Ai do // say service S i, j
6. While service S i, j has not been placed do
7. $S_i^c \leq C_{cur}^{cpu}$ && $S_{i,}^r \leq C_{cur}^{ran}$ then
8. Place Si,j on Curdev ;
9. $C_{cur}^{cpu} = C_{cur}^{cpu} -- S_i^c$;
10. $C_{cur}^{ran} = C_{cur}^{ran} -- S_{i,}^r$;
11. break;
12. else
13. Curdev = next device in the list D

The proposed algorithm first sorts applications in ascending order based on their deadlines (Line 1). Computational devices are sorted according to their proximity to users in the corresponding colony (Line 2). In other words, at first the resources of the fog cotrol node are utilized, after which the resources of the nearest neighbour colony and finally the cloud resources are utilized. This ensures that application services that have a short deadline will be placed on devices that have less response time for them. So, for each application Ai , its services are first placed on the fog control node as far as possible. If this node does not have enough resources to allocate one of the services of this application, its services will be sent to the next computational devices(Lines 3-11). Furthermore, the algorithm tries to place services as far as possible on the fog computing resources to not only providing QoS, but also reducing the cost of execution in the cloud.

Algorithm 2: An intelligent task scheduling approach (Learning Repository Fog-Cloud Approach)

We propose a hybrid approach based on the idea of both rule-based learning and case-based reasoning to produce efficient results. For this purpose, a learning repository is created that stores the particulars of each incoming task such as tuple identification (ID), tuple type, and the information of the resource where the task is served. Moreover, it also maintains the information Such as propagation time, execution time, energy consumption of the tuple at a specific resource. The tasks are scheduled to the available resources based on the information stored in learning repository [8]. Additionally, the selection of the service type (i.e., Fog, Cloud) is also made through the learning repository information.

We have tp_{total} –The total number of tuples and fg_{total} – The total number of fog servers. Initially, we are creating learning repository. 5% of the tp_{total} are used for training our proposed research. The remaining 95% of tp_{total} is used for testing purpose. This provides a detailed and self- explanatory description of our proposed approach named Learning Repository Fog-Cloud Approach (LRFC).

The learning repository storage Fs and Cs are created for Fog and Cloud, respectively. The initial tuples used for training are abbreviated as tp_{ini}. The tuples used for are testing are presented as tp_{final}. The SR stands for Storage Repository. Whereas DT represents the type of IoT job. Internal processing time ITP is the time that a tuple/job takes for processing using a resource of Fog or Cloud. The link propagation time is abbreviated as PTL .In the LR_{exe} process, the selection of fog server is decided on the distance between the request generated from and the server location where it is physically placed. The nearest server is selected with its server id fg_{id}. The fg_{id} of the serving server will be saved in LR storage as well [9]. In the case of cooperation, if the selected server fg_i is busy then the tuple will wait in the queue or will be sent to the cloud.

Algorithm 2. Learning Repository Fog-Cloud Approach

Procedure LRFC(tp_{total} x fg_{total})
1. C_T = [true,false]
2. S_t = {S_1,S_2,S_3}
3. D_c
4. C_o ={true,false}
5. G_w = {true,false}
6. F_s, C_s→Storage for fog and cloud
7. tp_{ini} =(tp_{total} x 5/100) ->initial execute
8. tp_{final} = (tp_{total} x 95/100) ->95% of remaining
9. split = splitting tp_{ini} into 5 chunks
10. for (j=0;to j=$split_{length}$) do
11. for (t_i in split[j]) do
12. Rd_{num}=[0,1]
13. If(Rd_{num}= 0) then
14. fg_i = execFOGSIM(fg_{total},t_i,C_T,0)
15. else
16. S_i=get service from S_i
17. SendToCloud(t_i.false,S_i,D_c)
18. temp = {t_i.D_T,t_i.ID,t_i.ITP,ti.PT,D_c.Name,t_i.Link}
19. C_i+ = temp
20. end if
21. end for
22. j++
23. end for
24. for each(t_i in tp_{final}) do
25. F_{fs} = getting t_i from F_s on the basis of D_T and orderby ti.IPT ∧ ti.PT_L
26. C_{cs} = getting t_i from C_s on the basis of D_T and orderby ti.IPT ∧ ti.PT_L
27. Boolean f.c
28. f= set time if F_{fs} is null
29. c= set time ifC_{cs} is null
30. if(F_{fs} is null OR C_{cs} is null) then
31. if(f) then
32. exec FOGSIM (fg_{total},t_i,C_t,0)
33. endif

Relationship of each pair of service S_i and S_j is one of three types, including the following

(i) Service S_i executes after S_j

(ii) Service S_i will execute before service S_j.

(iii) There is no rule on execution order between service S_i and service S_j.

5. Experimentation Results

Fog computing is a model in which the system tries to push data processing from cloud servers to “near” IoT devices in order to reduce latency time.According to Fig. 1 shows 36% of the research papers have implemented the proposed approach using iFogsim Simulation tool[10].Also 11% of the sudies have presented in evaluation of their case study using CloudSim tool.The execution orderings and the deployed places of services make significant effect on the overall response time of an application[11].The service deployment is a multiobjective optimization problem, there are so many proposed solutions for various targets, such as response time, communication cost, and energy consumption. In this paper,we have conducted experiments on two service deployment strategies,called cloudy and foggy strategies. In our work,we try to make an

execution plan which is aware of three issues (application's response time,network congestion,and server usage).Each Fog server has specific computing capacity in terms of MIPS. Similarly, the incoming job also has a specific size in terms of MIPS. The size, mean inter-arrival time and the change in number of tuples directly affect the utilization of Fog device [8].

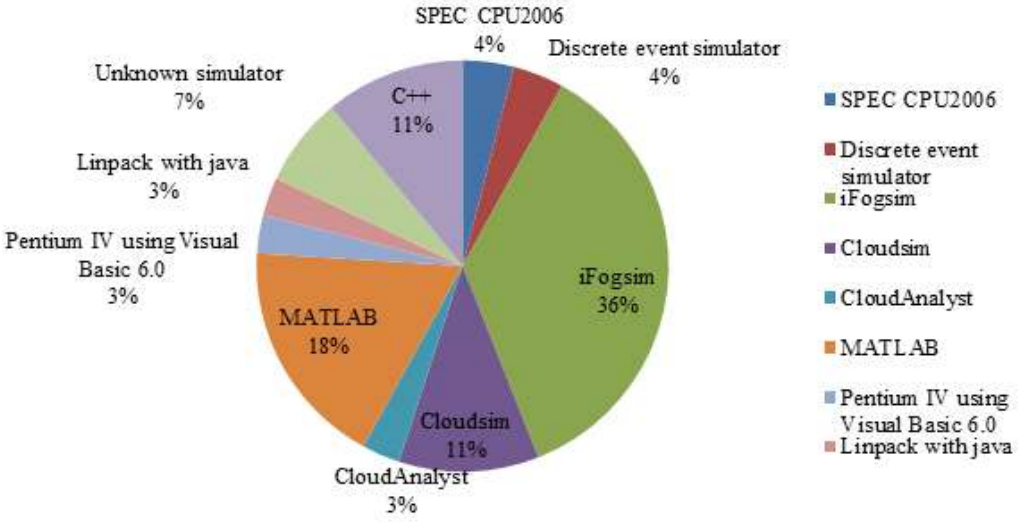

Fig.1: Comparison of presentation & evaluation tools

Fig 2 shows the utilization of 7 Fog servers. The X-axis presents fog devices and Y-axis shows their percent utilization. In case of LRFC, the minimum utilization of Fog servers is 86.295 and it reaches the 90.435 at maximum.

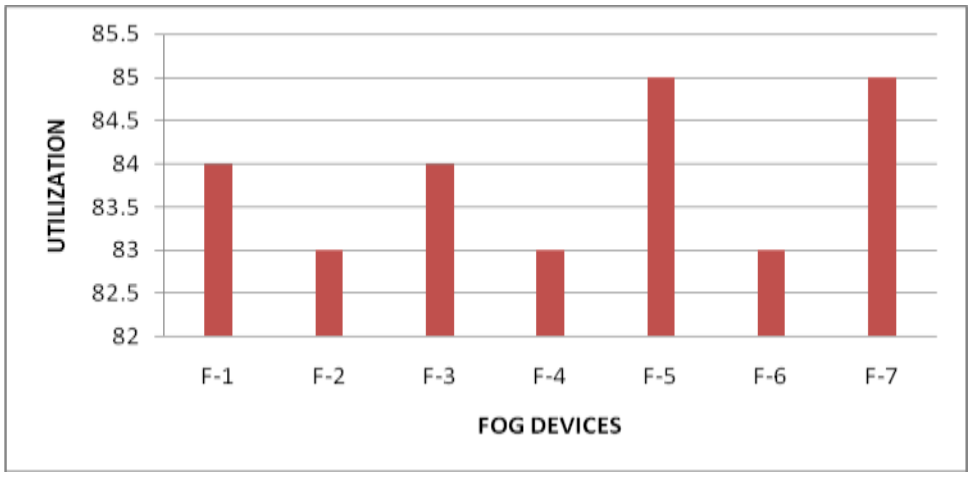

Fig 2: Fog Device Utilization

It can be seen that the proposed policy exhibits the lowest power consumption of fog resources.The efficient task scheduling results in reduced power consumption in distributed environment.Similarly, the energy consumption that is power consumed in a specific time period is also considerately decreased while allocating resources efficiently.

6. Conclusion

In Fog computing,edge devices are used to process the user request which have more processing power and are closer to the data sources than cloud resources.In this research paper, QoS-aware resource management technique is proposed for efficient management of resources which considers execution time, network usage and energy consumption as QoS parameters. However, providing an efficient solution to service placement problem in such systems is a critical challenge. To address this challenge, Qos-aware service placement policy for fog-cloud computing systems that places the most delay-sensitive application services as closer to the clients as possible. The performance of the techniques has been evaluated in Fog computing environment using iFogSim toolkit and the experimental results show that the proposed technique performs better in terms of QoS parameters. We have analyzed here service as a computing function of an application. Each service has three basic components, input, processing, and output. We assume that a service is an undividable processing unit. It means that a service has to be deployed only on a single device. Sometimes, the output of a service is used as input of other services.

Therefore, we regulate the executing priority between two services. Services having relationship together create a string of services. This paper explores task scheduling thoroughly in Fog computing environments. Task scheduling is aimed to assign a set of tasks to fog nodes to meet the satisfaction of QoS requirements in such a way that the execution and transmission time of tasks is minimized. An adaptive and intelligent task scheduling technique, Learning Repository Fog-Cloud (LRFC) has been proposed to improve QoS (i.e., response time ,and processing time of tuples) and energy consumption(i.e., power consumption of fog devices).Our strategy is to combine three considered components (application's response time, network congestion, and server usage) into one objective function. The verification shows promising results are obtained for LRFC both in terms of energy efficiency and QoS..

Acknowledgments

The authors are thankful to all the people who directly or indirectly supported us for the preparation of this research paper.

REFERENCES

[1] *O.Skarlat, M.Nardelli, S.Schulte,M.Borkowski, andP.Leitner,"OptimizedIoTserviceplacement in the fog," Service Oriented Computing and Applications, vol. 11, no. 4, pp. 427-443,2017.*

[2] *A. Yousefpour et al., "Qos-aware dynamic fog service provisioning," arXiv preprint arXiv: 1802.00800,2018.*

[3] *Elarbi Badidi, Awatif Ragmani Presented An Architecture for QoS-Aware Fog Service ProvisioningVolume 170,2020, pages 411-418*

[4] *S. Singh and I. Chana, "QoS-aware Autonomic Resource Management in Cloud*

Computing: A System Review", ACM Computing Surveys, vol. 48, no. 3, (2015), pp.1-46.

[5] *A. Vahid Dastjerdi, H. Gupta, R. N. Calheiros, Soumya K. Ghosh, and Rajkumar Buyya, "Fog Computing: Principals, Architectures, and Applications." arXiv:1601.02752(2016).*

[6] *Q. T. Minh,D.T. Nguyen, A. Van Le,H.D. Nguyen, and A.Truong, "Toward service placement on Fog computing landscape",in 2017 4th NAFOSTED conference on information and computer science,2017.pp.291- 296:IEEE.*

[7] *M. –Q. Tran, D. T. Nguyen, V. A. Le, D.H. Nguyen, and T. V. Pham, "Task placement onFog Computing made efficient for IoT application provision", Wireless communication andmobile computing,Vol.2019,2019.*

[8] *S. Agarwal, S. Yadav and A. K. Yadav, "An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing", International Journal of Information Engineering and Electronic Business, vol. 8, no. 1, (2016), pp.48.*

[9] *Carla Mouradian, Diala Naboulsi, Sami yangu, Roch H. Glitho, Monique J. Morrow, and Paul, A. Polakos. A Comprehensive Survey on Fog Computing: State-of-the art and Research Challenges.*

[10] *H. Gupta, A. VahidDastjerdi, S. K. Ghosh and R. Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments" arXiv preprint arXiv: 1606.02007,(2016).*

[11] *Chiang Mung and Zhang Tao. Fog and IoT: An Overview of Research Opportunities. IEEE Internet of Things Journal,3(6):854–864, December 2020.*

[12] *Mohammad Aazam Eui-Nam Huh, "Fog Computing and Smart Gateway Based Communication for Cloud of Things" August 2014, International Conference on Future Internet of Things and Cloud.*

[13] *Salim Bitam,Sherali zeadally & Abdal hamid Mellouk, "Fog computing job scheduling optimization based on bees swarm"-pages 373-397, Accepted 05 March 2017,Published online:10 April 2017.*

[14] *Intharawijitr, K.; Iida, K.;Koga,H."Analysis of fog model considering computing and communication latency in 5G cellular networks". In Proceedings of the 2016 IEEE International Conference on Pervasive Computing and Communication Workshops(PerCom Workshops),Sydney,NSW,Australia,14-18 March 2016;pp.1-4.*