

Review on Performance Enhancement of Fog Computing

Anju Babu¹ and Dr.Shobha Rekh²

¹⁻²Karunya Institute of Technology and Science Coimbatore
Email: anjubabu@karunya .edu, shobhapaulson@karunya.edu

Abstract—Fog computing has proved to be one of the efficient method adopted for almost all IoT applications present today. Fog computing mainly focuses on reducing the limitations of cloud computing such as latency, power consumption, storage cost etc. Many IoT application has improved their Quality of Service by incorporating the fog computing technique. This area has gained recent attraction and is found to be one of the most upcoming area of research which has gained attention recently. QoS improvement methods includes many methods such as including efficient placement techniques, offloading algorithms, fog node selection methods etc.This paper concentrates on a review on the fog computing tools and some recent methods adopted to improve the quality of service.

Index Terms— fog computing, Quality of Service, placement techniques, fog computing tools.

I. INTRODUCTION

Cloud computing has emerged as one of the most popular paradigm in the recent years. Combining with IoT it has enhanced the field of automation. In spite of its wide spread acceptance, it faces many challenges. For time critical applications such as medical applications, gaming applications, disaster management etc, the latency experienced by the application seems to be very high. Fog computing was introduced to reduce such latencies. In fog computing not all the data generated by sensors are sent to the cloud which causes a considerable reduction in the latency experienced by the application. Computational capacities of the edge devices are utilised here so that the burden on cloud service can be reduced both in terms of computation and storage. Thus fog computing helps in reducing cloud overhead in computation, storage and also the power consumption, overall cost etc.

Fog computing is arranged as a three layer architecture; cloud layer, fog layer and end devices. Thus it shows that fog computing does not completely eliminate the cloud layer, it just helps the cloud computing technology to reduce its overhead and improves its quality of service.

In this paper section 2 describes about the various types of fog computing applications that helps in automation. section 3 describes about the different fog computing tools available and section 4 describes the various methods that are used to improve the quality of service of fog computing.

II. CLOUD TO FOG

In order to improve the performance parameters such as latency, cost of execution, energy consumption etc many applications has been upgraded from cloud architecture to fog rather than depending on cloud architecture alone. Yan-Da Chen et al. in [2] explained how for a smart home system better efficiency and performance can be achieved with the help of fog c-cloud architecture.

An efficient car parking system which can reduce pollution and gasoline usage was suggested by Kamran Sattar Awaisi et al. in [3]. In that they have reduced the time to find a parking slot for vehicle by using fog computing techniques.

Montesdeoca-Mendez et al. in their research [4] proposes the design of an IoT system which can measure environmental variables, comprising of a Wireless sensor network, fog computing and cloud computing, which uses end-to-end encryption techniques. The architecture comprises of four layers; user layer, fog layer, cloud layer and WSN layer.

Tuan Nguyen Gia et al. [5] have exploited fog computing to improve a health monitoring system. In this the advanced methods such as embedded data mining distributed storage and notification services at the edge of the network were used at the smart gateways to use the concept of fog computing.

John K. Zao et al. [6] have tried to incorporate a fog layer to process the EEG signals generated. And hence was able to reduce the latency during processing.

Kriti Bhargava et al. have introduced a fog service which can perform the real time monitoring of animals in a farm. It can analyse and observe their mobility patterns which can be used to assess the animal health and also provides location based services such as virtual fence applications.

Kaisar Shafi et al. a proposal for an intrusion-detection-system-with-mist [8] by the addition of a control layer and the fog that gives the scale of the IoT network, and the appropriate information technology for the detection of the attack.

All of these actions will let you see how the fog computing to improve the performance of IoT systems, in particular, in order to reduce latency, costs, cloud computing, data security, etc.

III. FOG SIMULATION TOOLS

As the fog computing is starting to gain popularity, along with a wide range of modelling tools, and modelling of the fog computing has also become popular. In this chapter, we will discuss some of the most popular fog simulation tools.

One of the most popular and flexible fog simulation tool is iFogSim, which was developed by Harshit Gupta et al. In their work, they came up with a simulator, called iFogSim [9], which was introduced by the CloudSim extension, which is used for the simulation of the IoT and the damp environment, and that are able to measure the quality of service parameters such as latency, network congestion, energy consumption, and cost. In the process, and the control model, which is widely used in many scientific publications, and a variety of experiments, and popularity after the release. MyiFogSim that you can manage the instance migration for the users of mobile devices, and is an extension of the iFogSim. The main feature of this game is to allow users to simulate the mobility of the features and how they are related to the VM's migration policy.

Another extension is the iFogSim With Data Placement tool [10], which suggests a way to manage data stored in a fog system. In this data set it is done by looking at delays, network usage and power consumption.

EdgeCloudSim is another CloudSim extension for WLAN, a WAN that focuses on network modeling and hardware flow [11]. It also faces network load that is ignored by iFogSim.

Pure Edge Sim is another tool used for modeling and designing cloud, fog and edge computing which is an extension of CloudSim Plus [12]. Here local parameters, power model and virtual machine settings can be adjusted by users.

Another simulation tool used in addition to C++ is FogNetSim++ [13] built on OMNeT++ that performs network simulation and provides configuration options to manage fog networks. Communication agreements such as MQTT, CoAP can be managed by this simulator, and can also be used for different travel models such as Linear Mobility or Tractor Mobility.

Another CloudSim expansion known as Edge network CloudSim focuses on integration and configuration on edge devices [14]. Here the Edge Service is a service chain model, the Edge Datacenter Broker communicates with the user, and Edge VMs help define the app application. A summary of the popular simulation tools are given in Table 1.

In addition to these simulators we can also find simulation solutions in MATLAB, Docker etc.

IV. QoS IMPROVEMENT IN FOG

The study is working to improve the quality of computer fog service which started as soon as the fog computing started to gain popularity. The main focus is on delays, cost of execution, security, reduction of cloud storage etc. A lot of researchers is still working on these areas.

TABLE I. SOME POPULAR FOG SIMULATION TOOLS

Sl.no	Fog simulation tool	characteristics
1	iFogSim	Can measure quality of service parameters easily
2	MyifofSim	Can simulate mobility features
3	iFogSim With Data Placement tool	Manages data stored in the fog system
4	EdgeCloudSim	Supports WLAN and also considers the network load
5	Pure Edge Sim	User can adjust local parameters, power model and visual machine settings
6	FogNetSim ++	Can manage MQTT, CoAP
7	Edge network cloudsim	focuses on integration and configuration on edge devices

Sukhpal Singh Gill et al. propagated the novel resource management process (ROUTER) using the Swarm optimization particle-enabled Cloud computing environments [15]. In this study the authors proposed a cloud-based fog component to achieve efficient resource management of IoT devices. The PSO's multi-pronged resource management methods and activities to ask for software administrators of fog software are also discussed in this paper. The authors also analyze the typical example of a smart home that uses cloud computing-assisted computing.

Mohammad Irfan et al. in [16] focused on distributing system modules between fog devices and cloud data centers to improve the efficient use of cloud fog resources. Performance parameters such as response time, delay, power consumption, etc. Improved by placing system modules on fog devices. The performance test of the dual load balancing algorithm using the iFogSim simulator has also been recommended by the authors.

Karthik Kumar et al. conducted research on computational uploads of mobile systems [17]. Reduction can be used to improve the performance and power consumption of mobile systems. In this paper they have explored and categorized a large area of research associated with computation loading of mobile systems. A variety of algorithms have been used to classify and silence programs and studies are conducted to show how they have improved the performance and energy they have saved.

Mukhtar ME Mahmoud et al. propose a strategic plan to set application modules using the power allocation strategy [18]. They have proposed a policy on the allocation of installation of installation modules in a cost-effective manner. The main purpose of the proposed distribution strategy is to increase the efficiency of fog devices using robots (IRR) and advanced voltage and frequency scaling (DVFS) algorithms.

In [19] Luiz F. Bittencourt et al. discusses resource allocation issues explores the infrastructure of high-capacity and cloud-based data centers, analyzes application classes and various planning policies. They have introduced a number of programming methods that look at user mobility and computer power, in the context of Fog computer infrastructure to address this challenge.

Mohit Taneja et al. in their paper [20], described a Module Mapping Algorithm that gives emphasis to application modules in the construction of fog clouds. These technologies are available to provide efficient use of resources in network infrastructure for IoT applications. Three algorithms used to improve the module map .module mapping, the lower bound algorithm, compare network node with the application module.

Quiliang Zhu et al. proposed an algorithm for performing the task of reducing the load [21]. Depending on their algorithm, there will be an agent which is a control center, with information about cpu usage, band width, memory etc. Depending on this the agent will decide whether it needs to be done on a mobile, local cloud or cloud server.

V.B.C.Souza et al. in [22] has tried to add computer performance metrics. They have introduced and developed solutions to service delivery problems that reduce service-related delays that address and meet energy needs. These service delivery issues aim to reduce service-related delays in accessing resources that meet service requirements.

Xuan-Qui Pham et al. described a high-performance algorithm for job planning problem [23] whose main purpose is to achieve a balance between performance time and cloud resource costs.

Mohit Taneja et al. in [24] has explained about a resource-aware placement of data analytics platform in fog computing architecture, this placement method will effectively deploy the analytic platform to run either on the cloud, or the fog, thus reducing the network costs and response time for the user.

Yiqiu Fang et al. described two methods of task planning based on load balancing in cloud computing [25]. This two-level task algorithm has been proven to be widely used with the resources and user needs it meets. The same has been modeled and validated with the help of the Cloudsim tool kit.

Ruilong Deng et al. in [26] modeled the cloud fog computing program on power consumption and job delays and created a problem of job sharing. The approximate solution to the big problem has been solved by decay, and

Table II : SUMMARY on Qos IMPROVEMENT METHOD

Sl_no	Paper	Methodology/algorithm	Key focus
1	[15]	Resource management using particle swarm optimization technique	
2	[16]	Dual load balancing algorithm	Module placement on fog devices
3	[17]	Computational offloading	Improves performance and energy consumption
4	[18]	Application module allocation using round robin	Improves the energy efficiency
5	[19]	Scheduling policies	
6	[20]	Module mapping algorithm for application modules	Efficient utilization of network resources
7	[21]	Task offloading	
8	[22]	Service allocation	Latency reduction
9	[23]	Task scheduling algorithm	Cost reduction on cloud
10	[24]	Resource aware data placement method	Reduce network cost and response time
11	[25]	Task scheduling using load balancing method	Improve resource utilisation
12	[26]	Focused on workload allocation problem	Reducing communication latency
13	[27]	FEMTO	Reduces energy consumption
14	[28]	Energy aware placement method	Reduce end to end latency
15	[29]	Fog node selection	Reduced time delay and improved data rate
16	[30]	<i>K means clustering</i> and PCA fog computing design method	Focuses on fog node selection

three problems have arisen under the successive systems. The solution to these minor problems has been found with the help of existing best practices. Finally, based on extensive simulation and numerical results, they have shown that using fog can significantly improve cloud computing performance by reducing communication latency.

Guowei Zhang et.al in [27] discussed a novel algorithm for service loading called FEMTO. Thus the metrics for justice planning are defined in each fog zone based on reduced power consumption, historical power, and prioritization. The algorithm determines the selected loads of the fog node, the transfer power, and the subtask size that is presented fairly and minimized. In this case the low power consumption is achieved against the cost of proper loading and operation between the many fog areas. The reduced power consumption of this algorithm and performance may have been confirmed with the help of price simulations.

The volatile human-event processing of applications for optimization, was proposed by Instagram Ghosh et.already in [28]. It is designed to provide an analytical, data-flow, by means of a set of frameworks, and cloud resources with the objective of minimizing the pass-through latency on the data.

There were Balevis et.in their work[29], have attempted to increase the average transmission rate and reduce the transmission delay for the purpose of determining the optimum number of nodes that need to be updated to the fog nodes, the additional computational power.

As a result of the study revealed that the optimum number of fog nodes does not increase in direct proportion to the increase in the total number of nodes. The optimum number of fog nodes is reduced when the channel has a high path-loss rates. The results indicate that the proximity to the network with a high degree of loss in the field, the fog nodes may be selected from among those who have the greatest mathematical abilities.

Yuxio, et al. In their paper, [30] stated on the K-means clustering, and data (missing EVENT), which allows for a new, mist service, for the purpose of detection .Proposed construction and machine learning, the design facilitates a node to the selection and exploration in a foggy, company.

V. CONCLUSIONS

In this article, we have tried to explain the importance and advantages of fog computing, in comparison with cloud computing solutions.Overhead on cloud computing in terms of storage,processing ,energy consumption due to increased number of IoT devices can be decreased by using fog computing along with cloud resources. This survey shows how fog computing can be utilized to improve the quality of cloud computing in the areas such as latency,energy consumption ,cost of implementation etc.Further improvements in the fog computing was achieved by using various algorithms as explained in previous sections. This survey can help the researchers to identify the research problems in the field of fog computing which can further improve the performance of IoT applications.

REFERENCES

- [1] Mouradian, Carla, et al. "A comprehensive survey on fog computing: State-of-the-art and research challenges." *IEEE communications surveys & tutorials* 20.1 (2017): 416-464.
- [2] Chen, Yan-Da, Muhammad Zulfan Azhari, and Jenq-ShiouLeu. "Design and implementation of a power consumption management system for smart home over fog-cloud computing." *2018 3rd International Conference on Intelligent Green Building and Smart Grid (IGBSG)*. IEEE, 2018.
- [3] Awaisi, Kamran Sattar, et al. "Towards a fog enabled efficient car parking architecture." *IEEE Access* 7 (2019): 159100-159111.
- [4] Montesdeoca-Mendez, Paul A., et al. "Implementation of fog computing for data pre-processing in an IoT and cloud computing scenario." *2019 IEEE Colombian Conference on Communications and Computing (COLCOM)*. IEEE, 2019.
- [5] Gia, Tuan Nguyen, et al. "Fog computing in healthcare internet of things: A case study on ecg feature extraction." *2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing*. IEEE, 2015.
- [6] Zao, J. K., Gan, T. T., You, C. K., Méndez, S. J. R., Chung, C. E., Te Wang, Y., ... & Jung, T. P. (2014, June). Augmented brain computer interaction based on fog computing and linked data. In *2014 International conference on intelligent environments* (pp. 374-377). IEEE.
- [7] Bhargava, K., Ivanov, S., Kulatunga, C., & Donnelly, W. (2017, January). Fog-enabled WSN system for animal behavior analysis in precision dairy. In *2017 International Conference on Computing, Networking and Communications (ICNC)* (pp. 504-510). IEEE.
- [8] Shafi, Q., Basit, A., Qaisar, S., Koay, A., & Welch, I. (2018). Fog-assisted SDN controlled framework for enduring anomaly detection in an IoT network. *IEEE Access*, 6, 73713-73723.
- [9] H. Gupta, A.V. Dastjerdi, S. Ghosh, R. Buyya, Ifogsim: a toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments, *Softw. Pract. Experience* (2016)
- [10] M.I. Naas, J. Boukhobza, P. Raipin Parvedy, L. Lemarchand, An extension to ifogsim to enable the design of data placement strategies, in: *2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*, pp. 1-8.
- [11] C. Sonmez, A. Ozgovde, C. Ersoy, Edgecloudsim: an environment for performance evaluation of edge computing systems, *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, (2017), pp. 39-44
- [12] PureEdgeSim simulator online: <https://github.com/CharafeddineMechalikh/PureEdgeSim>, accessed in June, 2019.
- [13] T. Qayyum, A. Malik, M.A. Khan, O. Khalid, S.U. Khan, Fognetsim++: A toolkit for modeling and simulation of distributed fog environment, *IEEE Access* PP (2018). 1-1
- [14] M. Seufert, B.K. Kwam, F. Wamser, P. Tran-Gia, Edgenetworkcloudsim: placement of service chains in edge clouds using networkcloudsim, in: *2017 IEEE Conference on Network Softwarization (NetSoft)*, pp. 1-6
- [15] Gill, Sukhpal Singh, Peter Garraghan, and Rajkumar Buyya. "ROUTER: Fog enabled cloud based intelligent resource management approach for smart home IoT devices." *Journal of Systems and Software* 154 (2019): 125-138.
- [16] Bala, Mohammad Irfan, and Mohammad Ahsan Chishti. "Offloading in cloud and fog hybrid infrastructure using iFogSim." *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2020.
- [17] Kumar, Karthik, et al. "A survey of computation offloading for mobile systems." *Mobile networks and Applications* 18.1 (2013): 129-140.
- [18] Mahmoud, Mukhtar ME, et al. "Towards energy-aware fog-enabled cloud of things for healthcare." *Computers & Electrical Engineering* 67 (2018): 58-69.
- [19] Bittencourt, Luiz F., et al. "Mobility-aware application scheduling in fog computing." *IEEE Cloud Computing* 4.2 (2017): 26-35.
- [20] Taneja, Mohit, and Alan Davy. "Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm." *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017.
- [21] Zhu, Qiliang, et al. "Task offloading decision in fog computing system." *China Communications* 14.11 (2017): 59-68.
- [22] Souza, Vitor Barbosa C., et al. "Handling service allocation in combined fog-cloud scenarios." *2016 IEEE international conference on communications (ICC)*. IEEE, 2016.
- [23] Pham, Xuan-Qui, and Eui-Nam Huh. "Towards task scheduling in a cloud-fog computing system." *2016 18th Asia-Pacific network operations and management symposium (APNOMS)*. IEEE, 2016.
- [24] Fang, Yiqiu, Fei Wang, and Junwei Ge. "A task scheduling algorithm based on load balancing in cloud computing." *International Conference on Web Information Systems and Mining*. Springer, Berlin, Heidelberg, 2010.
- [25] Deng, Ruilong, et al. "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing." *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015.
- [26] Taneja, Mohit, and Alan Davy. "Resource aware placement of data analytics platform in fog computing." *Procedia Computer Science* 97 (2016): 153-156.
- [27] Zhang, Guowei, et al. "FEMTO: Fair and energy-minimized task offloading for fog-enabled IoT networks." *IEEE Internet of Things Journal* 6.3 (2018): 4388-4400.
- [28] Ghosh, Rajrup, and Yogesh Simmhan. "Distributed scheduling of event analytics across edge and cloud." *ACM Transactions on Cyber-Physical Systems* 2.4 (2018): 1-28.

- [29] Balevi, Eren, and Richard D. Gitlin. "Optimizing the number of fog nodes for cloud-fog-thing networks." *IEEE Access* 6 (2018): 11173-11183.
- [30] Li, Yuxiao, et al. "Fog Node Selection for Low Latency Communication and Anomaly Detection in Fog Networks." *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*. IEEE, 2019