IJCSE International Journal of Computer Sciences and Engineering **Open Access**

Research Paper **Vol.-7, Issue-6, June 2019** **E-ISSN: 2347-2693**

Load balancing in Fog-Cloud Environment

T. A. Bhat[1*], J. S. Saini[2]

[1,2]Department of Computer Science and Engineering, Guru Nanak Dev Engineering College, Ludhiana, India

**Corresponding Author: bhat93tawseef@gmail.com, Tel.: +91-7889832729*

DOI: https://doi.org/10.26438/ijcse/v7i6.7177 | Available online at: www.ijcseonline.org

Accepted: 12/Jun/2019, Published: 30/Jun/2019

Abstract— Fog computing is latest addition in the environment of cloud computing which mainly brings cloud resources closer to the client. The main aim of fog computing is to execute the small tasks of smart devices at the edge devices whereas to put away the main intensive and non-sensitive tasks for the remote execution on the cloud. This overcomes the drawback that the cloud had due to the centralised control and problems of executing the small sensitive task at the remote area. In this paper, we provide the algorithm based on the three parameters time, energy consumption, and network usage on the basis of that, scheduling of task can take place between the two, cloud as well as fog, which distributes the load between them. The results we get, show that there is a significant decrease in time approximately 40%, network usage with 40% and significant decrease in energy consumption also on running tasks on fog than cloud . Finally, we assess the achievement of the task through the experimental simulation which shows significant decrease in the parameter values for local tasks at the fog computing.

Keywords— Cloud computing, Fog computing, Internet of things,Task scheduling

I. INTRODUCTION

Due to tremendous increase in the devices of Internet of things from million to billion, now it is predicted that there will be more than 50 billion devices in the market by 2020 with high responsiveness and less delay in the applications due to the fast growing technology in communication and the technologies of hardware [1]. Fog computing comes out the new paradigm in cloud computing to bring down the tiring workload on the traditional cloud data centres and prop up the solution to the widespread geographical area, delay sensitive and the applications which are aware about Quality of Service. It inducts fog computing and its characteristics. Hence fog computing acts as the bridge between the IoT and cloud [2] The substantial rise in the field of technology in cloud computing in the modern world needs to bring its portion of resources closer to the user. This bringing down of resources from cloud to edge devices which pop out the concept of fog computing and this prepares the new blooming in the architectural of computation. The amplification of which provides the beneficial fusion of cloud and fog diversified architecture. By thoroughly going down the ladder in research, the existence of fog computing was found [3]. Fog computing puts the rise in the ability of the end devices, those devices which are located at the networks edge which includes IoT sensors or the things which are wearable- those allow the execution of the tasks closer to the IoT users. This is the way to minimize the total delay in the services time in response, also reduce the consumption of energy and network congestion. It also presents some of the issues in security of the cloud [4,5]. The aim of implementing the fog in cloud is that smart devices can offload their computationally complex task to execute at remote cloud so that resource limited devices like mobile phones can have high battery life[6]. The provocation of using the cloud in smart devices is that to minimize the latency that will occur during the transfer of the data to the cloud and also the process execution of the task. These are the main provocations that come into play during use of cloud.

In cloud and Fog-cloud computing model, scheduling of a task is the main point to look after, that whether the task should run on cloud wholly or on fog environment and what are the advantages of Fog over cloud. In this paper we present the algorithm that schedules the tasks on the basis of few parameters like energy consumption, time and bandwidth. And provide the offloading policy on the basis of minimization of these parameter values used. The results are presented with simulation of the respective parameters.

The rest of the paper is systemized as: Section 2 provides the description of related work. In section 3 formulation of problem is describes, Section 4 presents the proposed work with details of the parameters used and task scheduling algorithm. Section 5 represents the experimental simulation of our work with results, Section 6 concludes the paper.

Rest of the paper is organized as follows, Section II contains the introduction of K-means clustering algorithm and how it is used as dimensionality reduction, Section IV contain the related work of disease prediction and data mining, section V explains the methodology of proposed work with flow chart, Section VI describes results and discussions of classification, and Section VII concludes conclusion of this work.

II. Related Work

Bhumika Paharia and **Kriti Bhushan** [7] provide an architecture which protects us from the malignant traffic produced by the attacks of Distributed Denial of Services(DDoS) from the user up to the cloud by making use of the fog computing. The purpose of using fog acts as refining levels between the client and the remote cloud, which generates the humongous amount of traffic. This work also enhances the overall performance of the network and minimizes the traffic towards the cloud from the client.

Redowan Mahmud *et al.* [8] proposed a management module about the perception of the delay in the application policy that joins the latency of heterogeneous services of delivery and the portion of signals of data that are processed in as per the units of time for heterogeneous applications. The aim of it is to meet the deadlines of service delivery with the guarantee of the quality in service and to utilize the resources to their maximization in the environment of fog.

Ibrahim Takouna *et al.* [9] proposed a placement model of parallel applications that are considering the delay in communication and awareness of energy in the data centers which are virtualized. The need of bandwidth requirement and the features of communication are dynamically identified for the heterogeneous widespread application and the allocation of the application is changed accordingly as if the current allocated place does not meet the criteria. There is a controller on the whole process, which controls the migration of tasks at certain times. The broker orders the instances based on the data traffic flow at the current time and provides the best virtual machine that suites the instance.

Tran Minh Quang *et al.* [10] provides the solid mechanism for the placement of services that utilizes the resources to their maximization on the edge and minimizes the centralization control of cloud. The reduction in the energy consumption, time of response and the monetary cost are the few parameters that improve the performance of the IoT when fog computing is used with the cloud as intermediate. The virtualized resources are optimally used during the fog-cloud architecture. By the decentralization in fog the enhancement in the performance of the location, real time response and also services of the consumption in energy happens. The hierarchical architecture of fog computing comprises of intermediate resource levels, which all are in support for the IoT services and had no centralization control.

Mohit Taneja and **Alan Davy** [11] proposed the presents the mapping of the module which utilizes the resources to their maximization and improves the quality of services in fog-cloud architecture.

III. Problem Formulation

The inclusion of fog computing in the cloud brings the new life in IoT. The limitations that arise during interacting with the cloud that these IoT devices face were increasing day by day as the millions of IoT devices are manufactured yearly. The traffic flow of data, delay in communication, real time response, Consumption of energy were the some of the big issues that cloud had with IoT devices. The collection of data is done by sensors used and then the collected data of the IoT application is processed on cloud or on the fog. The IoT application used comprises of several modules, these modules were totally run on cloud initially. But with the inception of fog with cloud most of the issues were solved. Some of the modules were run on the fog environment now and this increases the overall performance of application and the results are presented on the actuators of the respective application modules. As fog computing bring down the resource of cloud to a limited extent as there is always a hardware constraints in fog nodes, so limited amount of tasks can be executed on it. But these fog nodes provide quick response, consume less energy, have low latency etc. The aforementioned parameters and storage, computational power, bandwidths are some main parameters among which some are considered during the scheduling of task. The calculations of some of these parameters are considered for scheduling between cloud and fog with the cloud. We calculate three parameters like time, energy consumption and bandwidth with their priorities and on that basis we schedule the tasks.

IV. Proposed Work

In this section we present the example used, overview of our system model and how to schedule the task between cloud and fog computing. Firstly we choose the example of surveillance cameras, this application model comprises of five modules and that are cameras acts as sensors, motion detector, object detector, tracking of objects, pan-to-zoom control and interface for clients. Now we define these modules as Motion detector: This module reads the raw data of video taken by cameras to find the objects motion and then sent to object detector for further operations.

Object detector: In this segment of model extraction of the moving object takes place from the video stream and compared with those objects which are already active in that surveillance area. If the object does not match the previous data, tracking of object starts by calculating the coordinates.

Tracking of object: The previous coordinates of the object are taken and the pan-to-zoom (PTZ) configuration is calculated to their optimization for the surveillance cameras which surround the coverage area, so that tracking of object should be done efficiently.

Control of PTZ: It acts as actuator of the application system and on every smart camera it runs. By its control cameras are adjusted in such a way that PTZ parameters are calculated to their optimal value.

Client interface: To represent the tracked object on the output as a portion of video stream is sent for that application presents clients interface.

All these modules are within the smart camera used for surveillance as camera and ptz control acts as sensor and actuator. These all modules were run on cloud before the introduction of fog. Now some of the modules are run on fog devices to enhance the responsiveness to the clients. To schedule the task which module should run on cloud and which on fog we present the algorithm based on few parameters after the hierarchical model, which represents our topology of the module.

Where L_1, L_2, L_3, L_4 and L_5 represent the links between the modules as name of links are camera, video stream motion, object detected, location of object and parameters of PTZ. We give an overview of our hierarchical system that we use, it has four layers which comprises of cloud, proxy server, router and cameras with sensors and actuators connected to them. The data is collected from the sensors to cameras and through to cloud.

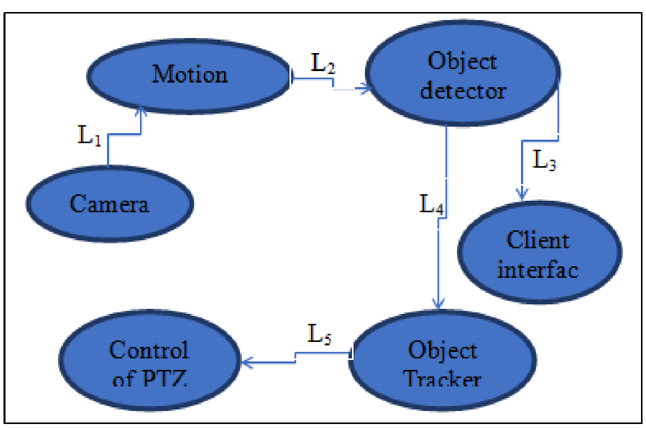

Figure 1. *Surveillance camera module.*

Fig. 2 represents our hierarchical model on which we are executing our data that we collect from the sensors of cameras C1, C2, C3 and C4. The levels between the remote cloud and the smart cameras are considered fog levels. These fog levels bring down the resources of cloud to the some extent like computational power, memory for storage, and have low latency, bandwidth and consume less energy. The devices used in these fog levels are called fog devices and they are configured as well so that they act as kind of mini to micro cloud. As the distance between IoT devices and Fog level is less, it increases the overall performance of the application system. A1, A2, A3 and A4 represent the actuators and S1, S2, S3 and S4 represent the sensors of the cameras through which collection of data happens. The proxy server and router with cameras act as fog levels in our model. Now we have two main levels first cloud level and second fog level. The data collected from sensors are run either totally on cloud level or with fog level in cloud.

By providing resources to these fog nodes some of the tasks are executed there and the congestion of data traffic to cloud mitigates to certain limit and the real time response increases to the IoT users. Because some of the IoT users require immediate response to their tasks like in case of healthcare system, so it becomes important where to execute the task. We take the set of tasks as input from the application module and these set of tasks are run on the cloud wholly and calculate the parameters like time, energy consumption and bandwidth requirement. Also provide the priority values respectively as alpha, beta and gamma whose total sum is considered as one. After executing the task on cloud we execute the same set of tasks with fog computing in cloud. These fog levels pre-process the data and run some portion on its own. Now we calculate the same parameters with it. After executing the data at both the frames, we compare the results.

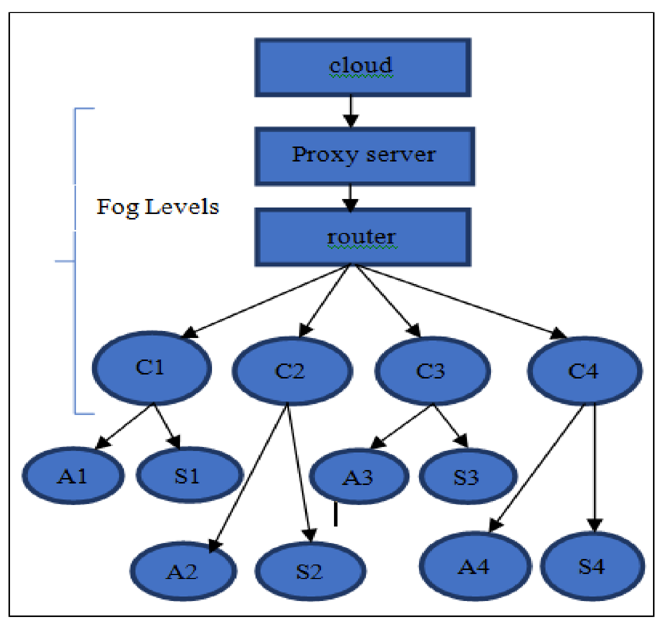

Figure 2. *Hierarchical model of our module*.

A. Description of Parameters

The parameters we discuss in our work are time, consumption of energy and network usage. First we define the symbols used in our work analysis on cloud {T_c, L, I_D, I'_D, B, S_p, C_p, e}. Where T_c represents execution time on cloud, L represents total delay which occurs during transmission of task among the modules during execution, I_D and I'_D represent the input data size and output data size respectively, B total bandwidth used, S_p represent the size of the data to be processed, C_{pc} represents the computational power of cloud and e represents the consumption of energy during transmission of unit data. The few above defined characteristics are same for fog also like I_D, I'_D, B and S_p besides these T_f and C_{pf} represent the execution time on fog and computational power of fog devices. Now the parameters we include in our scheduling algorithm are briefly discussed below.

B. Execution Time

The execution time of a task on fog and cloud is defined as the total time which includes the delays during transmission of data to the cloud directly and with the fog, in addition the time of execution at the cloud and fog respectively. The execution time relay on the computational power of the devices used, which may be fog devices or cloud for the processing of the data size which we consider for the processing. The computational power of cloud is always much greater than the fog devices.

C. Energy Consumption

The consumption of the energy is considered as the amount of energy required to send the certain amount of data size or to receive the data size to the cloud node or the fog nodes. It gives us the total amount of energy consumed by our devices during the run of our module on the cloud or fog. The total amount of energy consumption can be calculated as:

$E = (I_D + I'_D) \times e + E_E$, where E represents the total amount of energy consumed during the sending of I_D size of input data or I_D size of receiving data, e represents the energy consumed during the unit transmission of data and E_E represents the execution energy of task of the device used. The energy consumption is always greater during executing of data compared when the system is not processing data.

D. Network Usage

The usage of the bandwidth by sending and receiving the data through network channels to the cloud or fog. The usage of channel is always greater in cloud compared to the fog as a lot of channels are used to send the data compared to the fog as many modules are executed in the fog environment. The usage of total bandwidth during uplink and downlink of data can be considered as network usage.

$$N = U_b + D_b$$

Where *N* represents the network usage, U_b and D_b represent the uplink and downlink bandwidth consumption during transmission of data to the cloud or fog.

Utilization function: It is the beneficial function that we use in our algorithm, which provides combined values of the parameters that we use in our algorithm by normalizing then and with their respective priority values like alpha for time, beta for energy consumption and gamma for network usage. The function can be defined as:

$$F = \alpha T_N + \beta E_N + \chi N_N$$

Where T_N, E_N and N_N represent the normalizing value of time used, energy consumption and network usage respectively. And F represents the utilization function.

E. Algorithm for Task Scheduling

Input: Task size

Output: Destination for task [fog or cloud]

1: Calculate the time of execution on fog devices

$$T_f = L + (I_D + I'_D)/B + S_p/C_{pf}$$

2: Calculate the consumption of energy on fog devices

$$E_f = (I_D + I'_D) \times E_E$$

3: Calculate network usage during fog

$$N_f = U_b + D_b$$

4: Calculate the time of execution on fog devices:

$$T_c = L + (I_D + I'_D)/B + S_p/C_{pc}$$

5: Calculate the consumption of energy on cloud devices:

$$E_c = (I_D + I'_D) \times e + E_E$$

6: Calculate network usage during cloud:

$$N_c = U_b + D_b$$

7: Normalization of parameters

$$T_{fN} = T_f/(T_f + T_c) : \quad T_{cN} = T_c/(T_f + T_c)$$

$$E_{fN} = E_f/(E_f + E_c) : \quad E_{cN} = E_c/(E_f + E_c)$$

$$N_{cN} = N_c/(N_c + N_f) : \quad N_{fN} = N_f/(N_c + N_f)$$

8: Calculate utilization function for cloud

$$F_c = \alpha T_{cN} + \beta E_{cN} + \chi N_{cN}$$

9: Calculate the utilization function for fog

$$F_f = \alpha T_{fN} + \beta E_{fN} + \chi N_{fN}$$

10: if $(F_f < F_c)$

Processing destination = fog

else

Processing destination = cloud

11: end

The scheduling of the task as we present in our algorithm and the parameters calculated in it. These are all given the weightage as alpha = 0.5, beta = 0.3 and gamma = 0.2 as total sum should be equal to 1. Before providing the weightage we normalize all the parameters and calculate the beneficial utilization function.

V. Result and Discussion

This section consist of two main phases first calculating the parameters at both cloud and on fog environment, second calculating the utilization function for both environments. The task sizes we choose for it are 200,400, 600 and 800 respectively. Our simulation runs for the 1000 milliseconds. The execution time may vary as same task is run multiple times. We take the average values from that as the variation is minimal in execution time.

We calculate the execution time at both the environments as in fog and cloud, and the results show that it always shows positive results for fog compared to cloud for tasks of IoT devices. The process may be taking less time to execute on the cloud than on fog devices due to the high computational power of cloud. But the delay that occurs up to the cloud is much higher than fog devices. The latency links between sensors to camera is 1ms, between camera to router and router to proxy server is 2ms each. But from proxy server to cloud there is a huge delay of about 100ms which comes into play when we calculate the total time of task on cloud and fog. The sum of latency up to cloud brings the huge difference in the total time to execute the task. As we see from the results there is a huge difference between times taken to execute the task between fog and cloud as shown in Fig. 3 and Fig. 4, this all is due to the latency values. As we see from the two figures that fog takes very lesser time than cloud to execute the tasks. The average values are several times less than cloud.

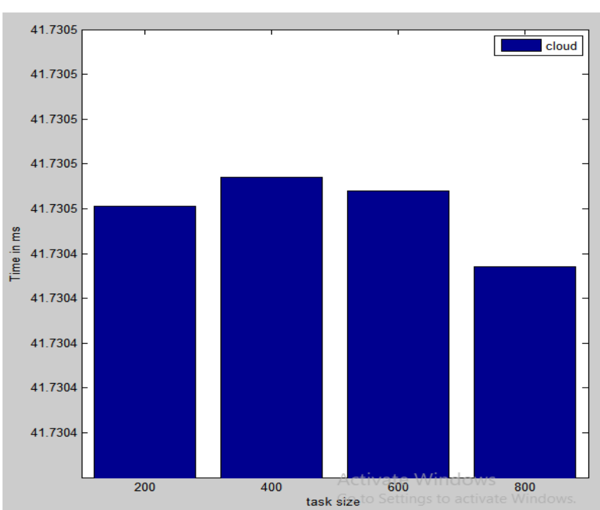

Figure 3. *Total time of execution on cloud.*

We calculate energy consumption at the two environments also and compare the energy. The energy used is always greater for busy state compared to idle state. These represent the state when the task is executing on the cloud or fog and when no process is executing on these two environments respectively. So during the busy state energy consumption is the sum of energy consumption of idle state to the energy required by the specific environment to execute the task.

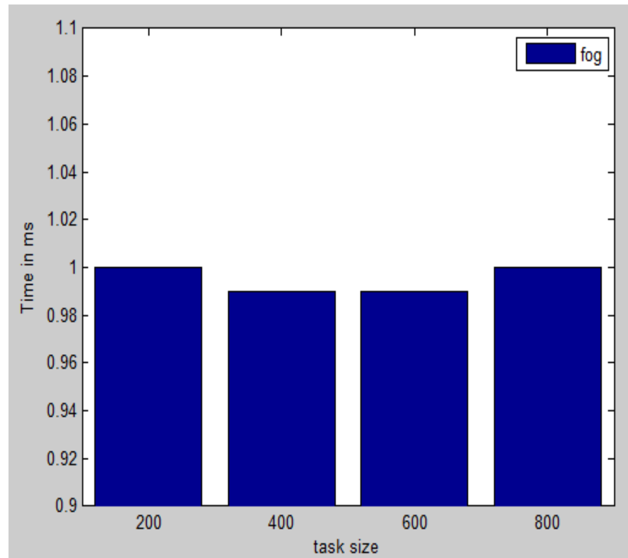

Figure 4. *Total time of execution on fog.*

Also we see there is not so much difference between energy consumptions between cloud and fog as it was in time. Because the small difference that comes here is due to the large energy consumption of idle states, as there is a huge setup of cloud so that requires huge energy also during the idle state. The difference in results is shown in Fig. 5.

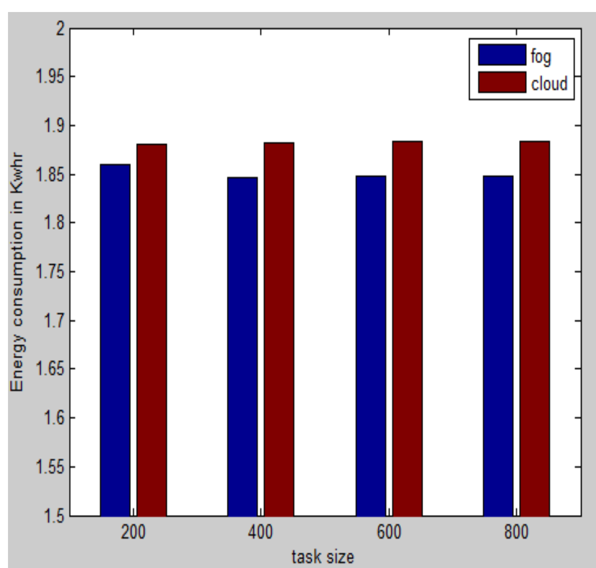

Figure 5. *Energy consumption on cloud and fog.*

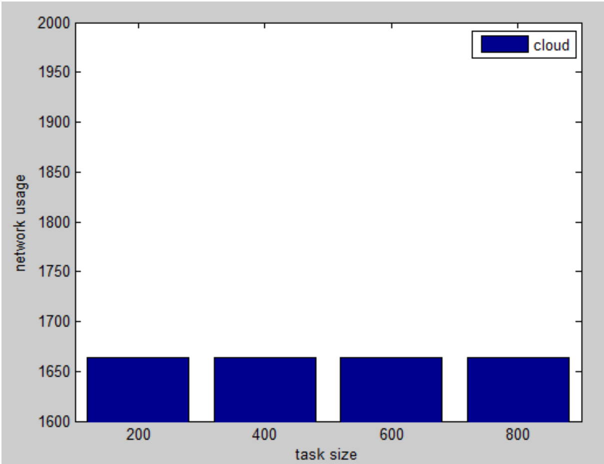

Figure 6. *Network usage on cloud.*

From the results we see that cloud always consume more energy than fog either it be on busy state or with idle state. The values are high for the cloud during the each of the task we run on it wholly. So, it is always better to run IoT tasks with fog than on cloud. Fig. 6 and Fig. 7 represent the network usage in cloud and fog. We can see that network usage remain same almost for all the tasks which we run because the capability and capacity of cloud is very high so for the small tasks the variations in networks usage is almost negligible. The consumption of network usage still high because of the channels that we use means we still have to traverse through all the fog level when we have to go for cloud. So lot of bandwidth channels are used for small tasks also. But once we see the graph of Fig. 7 of fog the network usage continuously change and these usages are still small compared to the usage of cloud, because we don't need to traverse so much for executing the task on the cloud.

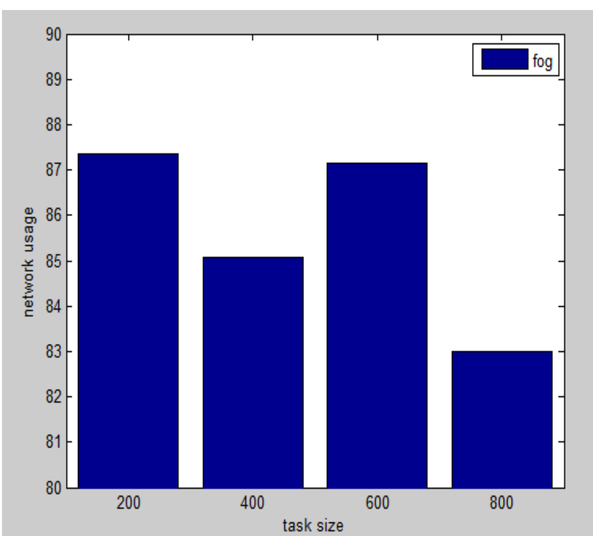

Figure 7. *Network usage on fog.*

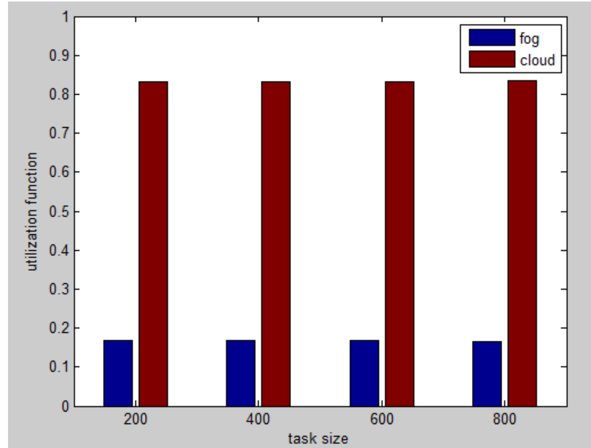

Figure 8. *Comparison of utilization function.*

After calculating the parameters individually and their comparison with each other on cloud and fog respectively, we make a general utilization function which combines all these parameters together and normalize them with their respective weightages and provide the values for each task. The values of cloud and fog which we got from the utilization functions are compared, the one which have minimal value task should be placed there. From the results it is clear to use fog than to use cloud wholly because results are better on fog. The utilization function consists of time, energy consumption and network usage, and alpha= .5, beta = .3 and gamma = .2 are weightages respectively. The results we get are shown in Fig. 8.

VI. Conclusion

We conclude the paper by saying it's better to use fog with cloud to optimise the quality of service and consume the resources to their maximum value. Fog computing provides the support to the cloud computing to the great extent, as local tasks are executed nearer to the respective IoT devices. By this a lot of traffic flow of data congestion to cloud also reduces and also overall improves the responsiveness of the real time task. There are lot of other parameters on the basis of which we can schedule the task between fog and cloud. Those need to be discussed in future work by keeping in mind when and where we need to trade-off among those parameters.

References

[1] T.Kleiner, " Internet of Things – *From Research and Innovation to Market Deployment*",River Publishers, pp. **355,2014**.

[2] M. Ketel, "*Fog-Cloud Services for IoT*", In the Proceedings of the 2017 SouthEast Conference on ACM SE, **USA**, pp.**262-264, 2017.**

[3] M.Kaur, A.M.Aslam, "*Big Data Analytics on IoT:Challenges,Open Research Issues and Tools*", Isroset-Journal (IJSRCSE)**,** Vol.**6,** *No.***3**, pp. **81–85, 2018**.

[4] W. Ramirez, X. M. Bruin, E.M. Tordera, B.B.C. Souza, and A.jukan, "*Evaluating the benefits of combined and continuous Fog-to-Cloud architectures*", Computer Communication, Vol.**113,** pp.**43-52**, **2017**.

[5] M.Patinder, P.Bansal, *"Cloud Forensics:An Overall Research Perspective",* IJSRNSC journal, Vol.**6,** No.**2,** pp.**5-10,2018.**

[6] Q. Zhu, B. Si, F. Yang, Y. Ma, "*Task offloading decision in fog computing system*", China Communicatins, Vol.**14**, Issue.**11**, pp.**59-68**, **2017**.

[7] B. Paharia, K. Bhushan, **"***Fog Computing as a defensive approach against Distributed Denial of Service (DDoS): a proposed architecture",* 9th International Conference on Computer Communication and Network Technology, **India**, pp.**1-7**, **2018.**

[8] R. Mahmud, K. Ramamohanarao, R. Buyya, "*Latency-Aware Application Module Management for Fog Computing Environments*", ACM Transanction on Internet Technology, Vol.**19**, No.**1**, pp.**1–21**, **2018**.

[9] I. Takouna, R. Rojas-Cessa, K. Sachs, and C. Meinel, "*Communication-aware and energy-efficient scheduling for parallel applications in virtualized data centers*", In the Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility Cloud Computing UCC, **Germany**, pp.**251–255**, **2013**.

[10] M.Q. Tran, D.T. Nguyen, V.A. Le, D.H. Nguyen T.V. Pham, "*Task Placement on Fog Computing Made Efficient for IoT Application Provision",* Wireless Communication and Mobile Computing, Vol. **2019**, pp.**1–1**, **2019**.

[11] M. Taneja, A. Davy, "*Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm*", In the Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Netwwotk and Service Management, **Portugal** pp.**1222–1228**, **2017**.

Authors Profile

T. A. Bhat received Bachelor degree in Computer Science and Engineering in 2016 from Islamic University of Science and Technology, Awantipora (Jammu & Kashmir) and is currently pursuing Master of Technology in Computer Science and Engineering from Guru Nanak Dev Engineering College, Ludhiana (Punjab). His main research areas are fog and cloud computing.

J. S. Saini received a Bachelor degree in Computer Science Engineering in 1998 from PTU, Jalandhar and a Master degree in Computer Science & Engineering in 2003 from Punjabi University, Patiala. He is currently pursuing Ph.D. He is with Guru Nanak Dev Engineering College, Ludhiana Associate Professor, in CSE department since 2006- 2019. His research areas are computer networks and networks security.

