International Journal on Emerging Technologies **10**(2): 122-128(2019)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

Agent based Dynamic Resource Allocation in Sensor Cloud using Fog Computing

Prashant Sangulagi[1], Ashok Sutagundar[2]
[1]*Bheemanna Khandre Institute of Technology, Bhalki, INDIA*
[2]*Basaveshwar Engineering College, Bagalkot, INDIA*

(Corresponding author: Prashant Sangulagi, psangulgi@gmail.com)
(Received 03 April 2019, Revised 25 June 2019 Accepted 08 July 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Managing the sensory data is a tedious task in sensor cloud. Usually sensor nodes produce multiple data and have heterogeneity character. Fog computing is a new paradigm to remove the latency problem and improves the system accuracy. Fog computing is a middleware between end devices and cloud server. As number of users increase, the resource allocation becomes very difficult in sensor cloud. In this paper, we are proposing a methodology for resource provision and pricing model for sensor cloud. The resource allocation is completed on priority basis as requested by the user. Initially 'N' number of tasks is requested by the user to cloud or fog. The tasks are sent to sensor cloud server and they are categorized as fog tasks and cloud tasks. The fog related tasks are forwarded to fog admin and cloud related tasks are forwarded to cloud admin. Pricing is fixed for fog tasks dynamic for cloud. Allocation of tasks to fog and cloud is done using mobile agents. The results show that the fairness and accuracy has improved by considering separate tasks for fog and cloud server.

Keywords: Cloud Computing, Resource Management, Pricing, Fixed Pricing, Dynamic Pricing, Fog computing, Sensor Cloud, Agents.

I. INTRODUCTION

In the present era cloud computing technology has attracted many corporate and business industries and received lot of attention among the people for improving their business skills [3, 17, 21]. As the number of users are improving day by day the request and it's fulfilling becoming a tedious work for the controller. The sensors used by the user/tablet are generating more data and cloud receiving huge resource and its becoming overhead for the cloud server to manage such a huge data [24]. To overcome, a new paradigm is brought forward called fog computing. Fog computing is the extended version of cloud computing which helps the cloud computing to overcome from resource allocation and its management. The number of servers and size of database is high in cloud server than compare to fog servers. To provide resources in better way the some strategies are adapted. The strategy involves allocation of resources to only those who are authorized with proper pricing strategies. The priority level is also given for those whose request level is very high. The pricing level decides priority to users. The pricing representations deem both user and contributor profit [5]. The main motivation is to provide the resources to needy user with less time and minimum delay. The cloud computing consists of three models private cloud, public cloud and hybrid cloud [14, 15]. In these types of clouds the prices are varied, like in public the pricing is dynamic, in private the pricing is fixed and in hybrid some of the services are fixed pricing and some are dynamic pricing. The fog architecture is another idea of the cloud at the edge of the system, is viewed as the fitting stage for some web of things [1] administrations and application. The primary point of fog registering is to put the information near end client. As per CISCO, because of its wide topographical circulation, it is appropriate for constant investigation and enormous information and gives an element of the area get to [4]. The fundamental task of fog is to convey information to desired user/client situated at an area which is at the edge of system [16].

Here the term edge refers to various hubs to which the end client is associated. Fog includes some of the parameters like Time, Perception (Cognition), Proficiency (Efficiency), and Dexterity (agility). Fogging process can call in some of the condition like, (1). Data is gathered at the extreme edge (2). Thousands or a huge number of sensors are located over a vast geographic region, are creating information

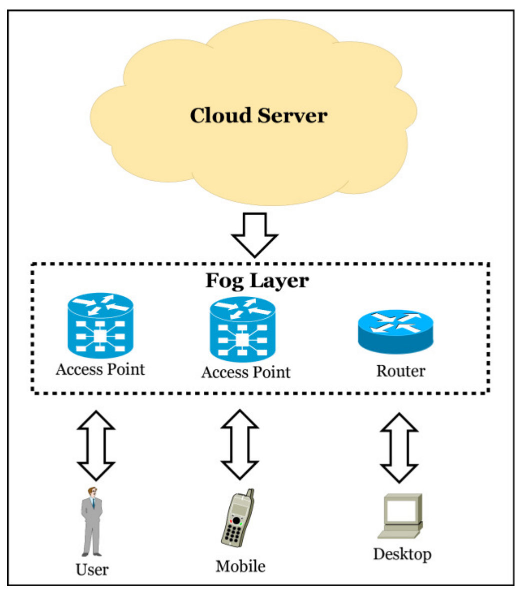

Fig. 1. Fog Architecture.

(3). It is important to break down and follow up on the information in under a second. The fog architecture is explained below as shown in figure 1.

Sensors, users, mobile and gateway as shown in figure 1. The bottom layer also contains apps that can be established in the quit gadgets (end devices) to provide their functionality [22, 23]. Some of the parts from bottom layer use the subsequent layer for communicating with cloud. The subsequent layer includes cloud offerings and resources that support aid management. The cloud layer is a top layer which lays resource management software program that manipulate the substructure and allow pleasant of Services to Fog applications [11]. Finally, the top (cloud) layer consists of the applications that leverage fog computing to deliver progressive and intelligent application program to quit users.

Looking at internal side of the software-defined based resource management layer, it establish many middleware-like military services to optimize the usage of cloud and fog sources on behalf of the applications. The intention of these services is to reduce the price of the use of the cloud at the same time that overall performance of applications reach suited

ranges latency through pushing task executions to fog devices [15].

Some of the characteristics of fog computing are low latency where fog underpins endpoints with best administrations at the rim (edge) of the system, mobility is fundamental for some applications of fog to discuss straightforward with cell phones and consequently help mobility methods, real time interaction requires ongoing communications for fast administrations (services) and heterogeneity where Fog hubs come in various frame factors and conveyed in a wide assortment of situations.

The rest of this paper is organized as follows. Section II gives related work, section III describes the resource allocation and pricing model, section IV depicts the simulation procedure, section V explains results of the proposed work and at last the section VI gives the conclusion to the paper.

II. RELATED WORK

Savani *et al.* has described that, day to day basis there is an increase in the usage of cloud computing and there is a requirement of effective and efficient usage of resources [2]. Author has proposed a priority-based resource allocation which can be used for proper allocation of resources and better utilization. Batches are prepared according to the user's request which consist of type of resource required by the user, amount of processor requirement and time taken to complete the task. Based on the availability of the resources, the priority is given. If multiple users request for same data, then first come first serve will be applied.

Pawar, *et al*. has presented a virtualization technique for fog computing and elastic resource allocation for cloud computing [3]. Authors have described how to minimize the response time and to increase throughput. When a user requests a task, it is sent to fog. The fog manager verify for the situation. Further the task will be in processing mode, next the appeal is sent to cloud server later sent to user giving utmost retort time.

Shanhe Yi *et al.* have proposed 2 algorithms, Meta task scheduling algorithm and priority based performance enhancement algorithm [4]. The proposed algorithm is based on min-min & max-min algorithm. In this the author has concentrated on priority of the user along with execution time.

Aazam *et al.* have discussed how to manage the bandwidth allotment among servers [5]. Author has proposed a queuing algorithm. The bandwidth allotment is done by applying the delay. The priority is also included for giving band with access to urgent needy applicants.

Niyato *et al*. have proposed an algorithm called task based allocation [6]. The aim is to get lower make span and provide better resource utilization. The architecture of the algorithm includes physical machine, task model and virtual machine. This algorithm assigns higher priority to the larger size-based tasks. When tasks are accepted, they are forwarded to queue then tasks are sent to VMs scheduler. The VMs scheduler checks for availability of resources. Then assigning VMs to task and execute the task.

Mulla *et al.* have proposed scheduling task using wel known scheduling algorithm which intern improves the accuracy [7]. The purpose of this paper is to provide server to the users in better way through cloud. The research has execution & completion time of the task.

Amalarethinam *et al.* has surveyed criteria's that are related to cloud computing technology [8]. Some challenges present in network environment. The author mentioned some of the terms like execution time, auction, utility function etc. The author also mentioned some of advantages and disadvantages of resource allocation.

Qadeer *et al.* has literature work of a variety of pricing strategy [9]. They have presented a few models like absorption pricing, high-low pricing, premium pricing, time-based pricing etc. some other pricing techniques also includes the overcrowding pricing.

Mishra *et al.* proposed nonlinear optimization approach for public transport which is based on user behaviour [10]. The author described structures, operation of scenario's and optimization. The main aim of this project is to have a maximum on demand access, maximum user benefit and to provide better social welfare constraints. This project provides good elasticity and cost. The future work of this project is to have better structure for real cost and transfer time.

Yadav *et al.* surveyed some of pricing models related to cloud computing [11]. The author mentioned some of pricing models like pay per user, pricing for subscription and hybrid pricing. This paper provides some of the common factors which affect the pricing. The author also describes about some of pricing schemes in cloud computing which includes description, features and implementation. The future work of this paper is to provide good quality of service between owner and customer.

The link has discussed about strategies and managing of resources in cloud networks [13]. Here they proposed a pricing mechanism called dynamic pricing based on availability of resources. The major intend of this project is to give priority-based model and considering usage of CPU. It provides better resource utilization and throughput. The future work of this project is to provide scalability and dynamicity in cloud environment.

Vinothina *et al.* has proposed a concept of cloud computing with pricing schemes [15]. Here they have discussed different pricing models for both provider and consumer (users). The major plan of this scheme is to have improved revenue for providers and a good quality of service for end users. The future work of this project is to get negligible risk factor for pricing strategies in cloud com-

puting.
Waheetha *et al.* has discussed about fog computing and its characteristics, issues and applications of fog [16]. The author mentioned some of the characteristics of fog like location awareness, mobility, low latency etc. and author also mentioned about fog structure and its benefits and applications.
Sangulagi *et al.* have discussed some of the resource management methods [18]. Author has mentioned that the resource allocator should be chosen in such a means that it should finish the task within the time and provision should be errorless and price optimized. Some of the methods of resource management are, adding data priority wise, saving data on types of cloud, saving data based on organizational modes.
Sangulagi *et al.* have described agent-based resource management in sensor cloud [19]. The sensor cloud is used to collect the resources from the sources and agents are included to further ease the energy utilization and improving the overall efficiency of the system. Now instead of reaching the data directly to cloud, agents will accept the data first and if data is same as before then such redundant data will be discarded. Here author has considered two models namely utility model and resource allocation model.

III. PROPOSED WORK

The proposed work explains pricing strategy for given quantity of users. Different pricing schemes have been provided for users, say fixed and dynamic pricing. Depending upon the selectivity server (Sensor Cloud Server or Fog Server) pricing and resources are allocated to user.

A. Network Environment

Proposed work system scenario is shown in figure 2 consisting of sensor nodes like tablet, mobile and laptop forming the sensor network and communicates between them and sends their data to sink node. The system environment also consist of fog server has fog nodes and database. The fog server stores nodes data and it also gives service to cloud server. The system environment also consist of cloud server at the top has many servers to store the sensory data. The stored data may be public or private depends on the controller. The sensor nodes sense the data and send to the sink node. The sink node checks the authentication of data and sends to cloud administrator where fog server and cloud server are located through the gateway. The admin will check the type of data and request from the user and accordingly the data is either stored in the fog server or cloud server. The prioritized data are stored in the fog server and pricing value is high in this. If the data is less prioritized and pricing value is less then that kind of data is stored in the cloud server. Time delay is applied in this whereas time delay is less in fog based servers.

B. Mathematical Model

The 'n' number of requests is generated from N number of users within the network. The requests are forwarded to cloud or for. The type of request may be some files, images or uploading of pictures and videos etc.
Nomenclature: 'N' represents user number, 'n' represents tasks number at a time, T_{tot}: total time, T_s: start time, T_f: finish time, P_{fixed}: fixed pricing, P_{dyn}: dynamic pricing

Dissimilar users have different pricing format. In this two types of pricing format are considered i.e fixed and dynamic pricing. Fixed charge scheme is only for fog service desires and similar way dynamic charge scheme for cloud service desired users. Price allocations depend upon the type of resources and its availability along with completion time.
Set of tasks are considered where the tasks may be related to fog or cloud. Say 'S' is the set of services and T1, T2, T3_ _ _ _ T_n are number of tasks.
For some set of users, there is a fixed price and for others the pricing is dynamic. This is based on regularity and their usage of resources. In this proposed work, three models are defined namely premium cost, prime cost and affordable cost. The premium cost comes with fixed pricing schemes and prime cost and affordable cost schemes comes with dynamic pricing.
The costs are characterized based on user demand, dispensation time and finishing point time. Premium cost is set, based on short time and cost. Prime or Affordable cost is set based on finishing time and cost. Here 'N' number of users requesting tasks to the fog.

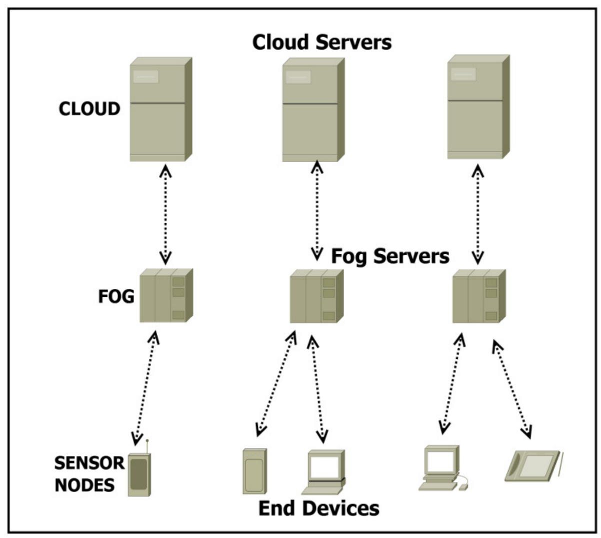

Fig. 2. Network environment.

Let r1,r2,r3. . . . . rn are the available resources and Rj is total resources

$$R_j = \sum_{i=1}^{n} r_i \quad \text{where } 0 < R_j < R_{tot} \quad (1)$$

Presume there are 15 resource demands, files are of different size. Let us assume, r1 =5mb, r2 =8mb, r3 =15mb. . . . . r10 =50mb. . . . r15=150mb.
The priorities are assigned based on the file size. Lowest file size will get highest priority and highest file size are given with lower priorities. Resource availabilities are also considered for selection of fog or cloud server. R_{req} is the resources requested by the user, R_{ava} is resource available; R_{tot} is total resource accessible in cloud or fog. The administrator can also set premium cost to its premium users by paying extra.
Let t1, t2, t3, t4... t_{tot} be the time taken by the particular task for allocating resource to the user. t_{tot} is the total time taken by the any of the task. It can be represented by

$$T_j = \sum_{i=1}^{n} t_i \quad \text{where } 0 < T_j < T_{tot} \quad (2)$$

$T_j = \sum_{i=1}^{n} t_i \quad \text{where } 0 < T_j < T_{tot}$

Fixed pricing and dynamic pricing strategies are adopted when there a more demand of resources. The pricing depends on 'Sf' scaling factor and its variable.

The cloud and fog have N number of servers and whenever there is a request from the user side then either fog or cloud server are assigned. Based on the assignment pricing are assigned. The running tasks require some memory and newly tasks are assigned based on the available resource and memory.

C. Flow for pricing model

1. For 'N' numbers of users in the system, each user can request 'n' number of tasks to the resource request processing.
2. The resource request processing checks for type of the user, whether the user is fixed pricing or dynamic pricing.
3. If the user is of fixed pricing, then tasks are forwarded to the fog servers. If the kind of user is dynamic, then tasks are sent to cloud server.
4. If the availability of resources is less in fog, the requested tasks by the users will be directly forwarded to the cloud.

D. Sequence of steps

The sequence of steps is finely presented in Fig. 3. Initially Sensor cloud admin receives request from the user for resource. Based on the type of resource and pricing strategy the resources are assigned by the admin. The admin will assign the task request either to fog or cloud server and it will be fulfilled. The work of RPA is to receive the task from user and assign them to either for server or cloud server. The following steps are used for execution.

1. 'n' tasks are requested by 'N' users to the sensor cloud server.
2. The tasks are categorized as fog tasks and cloud tasks by agents.
3. The cloud admin and fog admin receives tasks from respective say cloud tasks and fog tasks.
4. Sensor cloud administrator sends the task to sensor cloud server.
5. Sensor cloud services can be divided into three types namely, fog services, private cloud services, public cloud services.
6. The user request/tasks can be categorized into two types: prime tasks (high cost function) and normal tasks (low cost function)
7. Fog devices continuously update their resource into the cloud server.
8. Resource Processing Agent confirms the type of task (prime or normal).
9. Resources in the fog devices are checked if the task is prime type. Later availability is checked and if it is there then it is informed to fog admin. Intern admin assign one of the servers to prime task.
10. If fog servers are busy then all prime tasks are sent to cloud servers which consumes less time to execute i.e private clouds are considered.
11. The processed tasks from the private and public clouds are sent to individual user by RPA agents.
12. The completed tasks from the fog devices are given to the relevant user by the agents.

E. Agent Flow

The agent flow paradigm is as shown in figure 4. The agent is independent, autonomous, decision making and act upon the environmental condition [20]. The agents can be classified into two type, static agents and mobile agents. In this proposed work, we have considered the agent named, RPA (Resource Processing Agent) a mobile agent with consist of supervisor and database named Agent Black Board (ABB). The work of RPA is to check user's requirements and accordingly the task will be given to either sensor cloud server or fog server. The agent has been programmed with two conditions, premium users and normal users. If there is a request from the premium user, then task will be given to fog server and if user is normal then task will be given to sensor cloud server. Upon completion of task the RPA will provide the price list and required resource to users. After completion of task RPA updates the ABB.

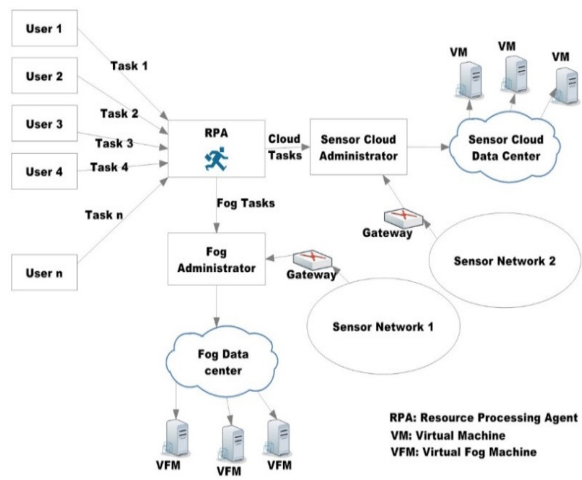

Fig. 3. Proposed block diagram.

F. Algorithm for proposed scheme

Nomenclature: N: number of users, n: number of requests or tasks, r: resources, RPA: resource processing agent, CSA: sensor cloud administrator, FA: fog administrator, Vm: cloud virtual machine, Vfm: fog virtual machine, Sf: Scaling Factor, Rreq: Resource requested by User, Rtot: Total resources available

1. Begin
2. Network initialization
3. User request for resources
4. Send r to RPA
5. RPA checks the user request
6. if (Rreq < Rtot) then
    RPA forwards task to FA
    else
    RPA forward task to SCA
    end
7. FA forwards to Vfm
8. SCA forwards to Vm
9. If User is Pfixed then
    Pfixed = Ttot * Sf
    else
    Pdyn = Ttot * Sf
10. if (Pfixed or Pdyn and Rreq) is done then

RPA will forward information to user
else
RPA check for next task.
11. End process

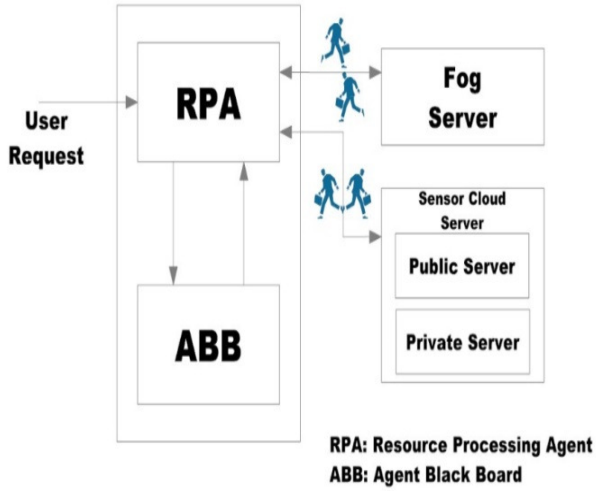

Fig. 4. Agent Activity.

IV. SIMULATION

The proposed work has been simulated using Cloudsim toolkit in the eclipse software and the program is executed in java language.

A. Simulation Model

The simulation of proposed model is done using Cloudsim toolkit. In this proposed model, we have considered number of instruction (MIPS) in the system. Then allocating available resources and pricing for these instruction in network environment.

B. Simulation Process

This subsection describes the process of simulation
-Accomplish the sensor cloud and fog environment
-Configure the sensor network, set the agents and inputs
-Apply the proposed plan and compute the performance parameters.

C. Configurations

Table 1: System Specification.

Sensor nodes	150
Virtual Machines	Xen (6 no's)
Agents	Mobile and Static
Bandwidth	10MBPS
No. of Servers	3
Processing Units	9
Storage	1TB
Cost	0.1Rupees/MB of Data
Operating System	Linux

D. Performance parameters

Here some of the performance parameters are written below:
-Fixed pricing: It is one of the parameters that estimate as total time taken by the task to complete and a scaling factor. Measured in rupees (Rs).
-Dynamic pricing: It is defined as total time taken by the task to complete it and scaling factor. Measured in rupees (Rs).
-Time: It is defined as difference between the start time and finish time. Measured in milliseconds (ms).

V. RESULTS

The proposed model is tested across some of the performance parameters. Figure 5 depicts amount of resource request required in percentage Vs pricing. As number of requested resources increases (in percentage), the pricing (Rs) increases. Based on the usage of CPU, bandwidth and storage, we are comparing with different pricing schemes like, Fixed, Premium, and Prime pricing. For fixed pricing, the number of CPU is 1, bandwidth is 2000MB and storage is 512MB. For premium pricing, the number of CPU is 3, bandwidth is 4000MB and storage is 1024MB. For prime pricing, the number of CPU is 5, bandwidth is 6000MB, and storage is 2048MB.

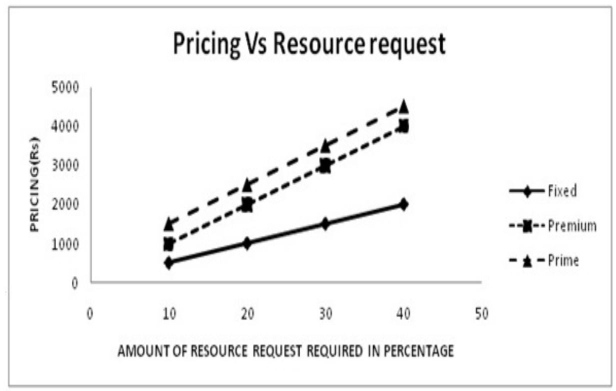

Fig. 5. Pricing Vs Amount of Resource Request Required in Percentage.

Fig. 6 depicts resource scheduling time Vs number of requests at time 't'. As number of requests increases in cloud and fog, the resource scheduling time also increases. Compared to fog, the cloud takes more time to process the tasks. In cloud, more number of resources are available and the data center is larger because it has more resources in it and takes more time to compute the task. In fog, the data center is smaller compared to cloud and it has limited resources in it, so it takes minimum time to compute the tasks.
Fig. 7 depicts number of virtual machines Vs processing request size in MIPS. As there is increase in the number of requests size, the creation of virtual machines also increases. The cloud generates more number of virtual machines compared to fog. If there is more demand for resources in cloud, the cloud creates more number of virtual machines as there is more number of resources is available. Whereas in fog, the availability of resources is minimum so that, fog creates minimum number of virtual machines. As there is more number of virtual machines in cloud, it processes more number of requests with less time. As there is less number of virtual machines in fog, it takes lots of time to compute the tasks.
Fig. 8 shows price Vs number of prime users. As the number of prime user's increases, the price for cloud and fog also increases.

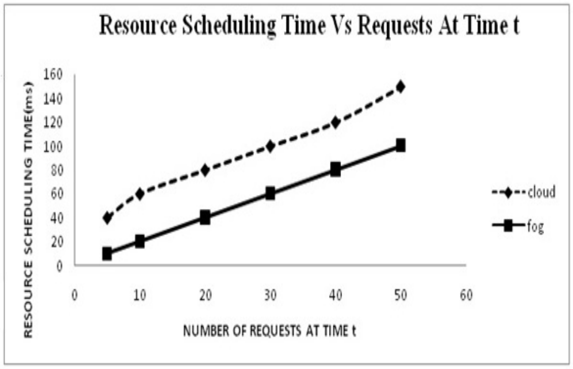

Fig. 6. Resource Scheduling Time Vs Requests at Time.

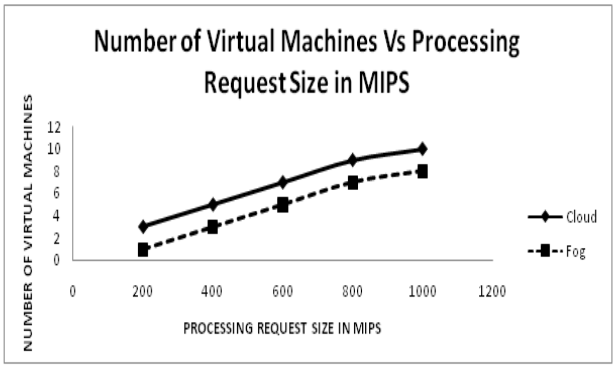

Fig. 7. Number of Virtual Machine Vs Processing Request Size in MIPS.

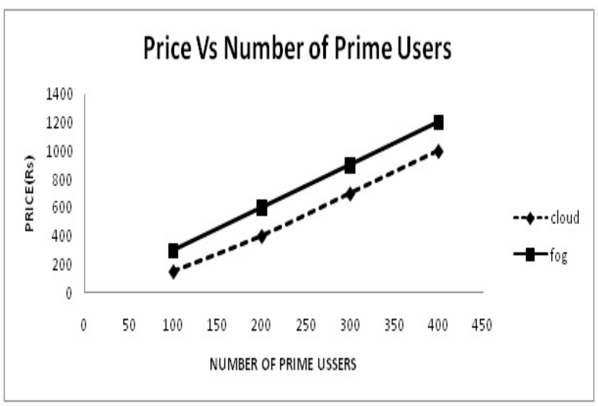

Fig. 8. Price Vs Number of Prime Users.

For prime users, based on the availability of virtual machine, CPU, bandwidth and storage, the access to the resources is given to the fog. If the availability of virtual machine, CPU, bandwidth and storage is not sufficient in fog, then the task is forwarded to cloud. For other users, the access to the resources is given to cloud. If the resources are available in fog then the access to requests is given to fog because fog consists of smaller data centre and minimum resources so it takes minimum time to compute the task and price for that request is also more. If the resources are busy in the fog then the access to request is given to the cloud because it consists of more number of resources and larger data centre compared to fog so it takes maximum time to compute the task and pricing for that request is minimum compare to fog.

Fig. 9 shows processing time Vs pricing. As the price increases (in rupees), the processing time also increases. Based on the different file size, we are comparing processing time. For the file size (in MB) and price (in Rs) 2000, the processing time is minimum compared to the file size (in MB) and price (in Rs) 6000. For the file size=2000MB, the availability of resources is checked in fog. If the resources are available then request is forwarded to fog where it computes the task with minimum time and allocates the price based on the file size. Similarly for the file size=4000MB and file size=6000MB. If the resources are busy in fog, then the request is forwarded to the cloud. Where it takes more time to complete the task so that the price is minimum.

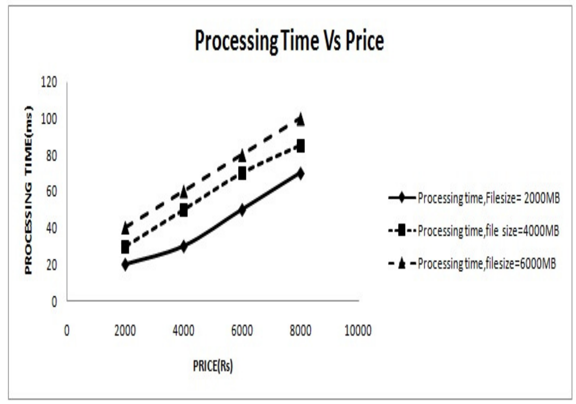

Fig. 9. Processing Time Vs Price.

VI. CONCLUSION

The proposed model proves to be good with the parameters like amount of resource request required in percentage, resource scheduling time, creation of more number of virtual machines, number of prime users, processing time and desired pricing for sensor cloud and fog. Agent paradigm helps in finding the appropriate user and their requirements within short time and it also help in providing the resources to the requested user without losing the accuracy. With suitable pricing and allotment of resources improves the services there by dipping the latency and lumber for clouds. The results show there is an improvement in fast resource allocation with minimum delay.

CONFLICT OF INTEREST: No conflict of interest.

ACKNOWLEDGEMENT

We would like to thank Basaveshwar Engineering College, Bagalkot for providing necessary resources in completing the research.

REFERENCES

[1]. Abirami, S.P. and Ramanathan, S. (2012). Linear Scheduling Strategy for Resource Allocation in Cloud Environment. *International Journal on Cloud Computing: Services and Architecture (IJCCSA),* **2**(1): 9-17.

[2]. Savani Nirav, M. and Amar, Buchade (2014). Priority Based Resource Allocation in Cloud Computing. *International Journal of Engineering Research & Technology (IJERT),* **3**(5): 855-857.

[3]. Pawar, C.S. Rajnikant, B.W. (2013). Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue. *International Conference on Intelligent Systems and Signal Processing (ISSP), Gujarat, India,* 311-316.
[4]. Shanhe, Yi, Zhengrui Qin, and Qun, Li (2017). Security and Privacy Issues of Fog Computing: A Survey. *International journal on IoT*, **5**, 9293-19304.
[5]. Mohammad, Aazam, Eui-Nam, Huh (2015). Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT. *IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangiu, South Korea,* 687-694.
[6]. Dusit, Niyato, Dinh, Thai Hoang, Nguyen, Cong Luong, Ping Wang, Dong In Kim, and Zhu Han (2016). Smart Data Pricing Models for Internet-of-Things (IoT): A Bundling Strategy Approach. *Access IEEE,* **30**(2), 18-25.
[7]. Minaz, Mulla, Malanbi Satabache and Netravati Purohit (2017). An Efficient Architecture for Resource Provisioning in Fog Computing. *International Journal of Science and Research (IJSR),* **6**(1), 2888-2895.
[8]. D.I. George Amalarethinam and S. Kavitha (2017). Priority based Performance Improved Algorithm for Meta-task Scheduling in Cloud Environment. *IEEE Second International Conference on Computing and Communications Technologies (ICCCT)*, 69-73.
[9]. Mohammed, A.Q., Habib, S. and A. Yazdan Javaid (2008). Differential Allocation of Bandwidth to Services based on Priority. IEEE 4th IEEE/IFIP International Conference, Uzbekistan, 1-5.
[10]. Mishra, S.K., Khan, Md Akram, Bibhudatta Sahoo and Jena, S.K. (2017), Time Efficient Task Allocation in Cloud Computing Environment. 2nd International Conference for Convergence in Technology (I2CT), 715-720.
[11]. Yadav, A, and Rathod, S.B. (2016). Priority based task scheduling by mapping conflict-free resources and Optimized workload utilization in cloud computing. *Computing Communication Control and automation (ICCUBEA) International conference*, Pune, India, 1-6.
[12]. Vinothina, V. Sridaran, R. and Padmavathi Ganapathi (2012). A Survey on Resource Allocation Strategies in Cloud Computing. *International Journal of Advanced Computer Science and Applications (IJACSA),* **3**(6), 1-7.
[13]. https://en.wikipedia.org/wiki/Pricing_strategies, 1/24/2018.
[14]. Ralf Borndörfer, Marika Karbstein, Marc E. Pfetsch, (2012). Models for fare planning in public transport. *International journal on Discrete Applied Mathematics,* **160**(18): 2591-2605.
[15]. Prathimasarma, Indeverchana, Anjubala (2017). A survey on different pricing schemes in cloud computing. *International journal of research in computer science,* **5**, 19293-19304.
[16]. R. Waheetha, Sowmya Fernandez (2016). Fog computing and its applications. *International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST),* **2**(19): 56-62.
[17]. Marek Moravþík, Pavel Segeþ, Jozef Papán, Jakub Hrabovský (2016). Overview of cloud computing and portability problems. *15th International Conference on Emerging eLearning Technologies and Applications (ICETA), Slovakia,* 1-6.
[18]. Prashant Sangulagi, Ashok V. Sutagundar (2017). Resource Management in Sensor Cloud. Advancing Cloud Database Systems and Capacity Planning with Dynamic Applications, Edited by Narendra Kumar Kamila and published by IGI-Global. DOI: 10.4018/978-1-5225-2013-9 (Chapter 7).
[19]. Prashant Sangulagi, Ashok V. Sutagundar (2018). Agent based resource management in Sensor Cloud. *International Journal of Emerging Technologies and Innovative Research (JETIR),* **5**(8), 331-336.
[20]. Sutagundar, A.V. and Manvi, S.S. (2008). Agent based Information Fusion in Wireless Sensor Networks. Proc. IEEE TENCON, Hyderabad, India.
[21]. Khan, Sheeba and Sharma, S. (2019). Analysis of Cloud Computing for Security Issues and Approaches". *International Journal on Emerging Technologies,* **10**(1): 68-73.
[22]. Gomathi, M., Sivanesan, P., Sekar, K.R., Manikandan, R. and Soundararajan, R. (2019). An Optimized Energy Efficient Clustering and Load Balancing Approach for Multi-Hop Wireless Sensor Network. *International Journal on Emerging Technologies,* **10**(1): 121-127.
[23]. Niharika, Himanshu Yadav and Chetan Agrawal (2019). A Review of Enhancing Energy Efficiency of Wireless Sensor Network. *International Journal of Electrical, Electronics and Computer Engineering*, **8**(1): 46-49.
[24]. Akanksha Pandey and Sanjeev Sharma (2017). Hybrid Encryption Technique for Security of Cloud Data. *International Journal of Theoretical & Applied Sciences*, **9**(2): 283-287.

How to cite this article: Sangulagi, P. and Sutagundar, A. (2019). Agent based Dynamic Resource Allocation in Sensor Cloud using Fog Computing. *International Journal on Emerging Technologies*, **10**(2): 122-128.