PRE-Fog: IoT Trace Based Probabilistic Resource Estimation at Fog

Mohammad Aazam, Marc St-Hilaire, Chung-Horng Lung, Ioannis Lambadaris
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada
aazam@ieee.org, marc_st_hilaire@carleton.ca, chlung@sce.carleton.ca, ioannis@sce.carleton.ca

Abstract— **Lately, pervasive and ubiquitous computing services have been under focus of not only the research community, but developers as well. Different devices generate different types of data with different frequencies. Emergency, healthcare, and latency sensitive services require real-time responses. Also, it is necessary to decide what type of data has to be uploaded to the cloud, without burdening the core network and the cloud. For this purpose, the cloud on the edge of the network, known as Fog or Micro Datacenter (MDC), plays an important role. Fog resides between the underlying Internet of Things (IoTs) and the mega datacenter cloud. Its purpose is to manage resources, perform data filtration, preprocessing, and security measures. To achieve this, Fog requires an effective and efficient resource management framework, which we propose in this paper. Fog has to deal with mobile nodes and IoTs, which involves objects and devices of different types having a fluctuating connectivity behavior. All such types of service customers have an unpredictable relinquish probability, since any object or device can stop using resources at any moment. In our proposed methodology for resource estimation and management through Fog computing, we take into account these factors and formulate resource management on the basis of fluctuating relinquish probability of the customer, service type, service price, and variance of the relinquish probability. With the intent of showing practical implications of our method, we implemented it on Crawdad real trace and Amazon EC2 pricing. Based on various services, differentiated through Amazon's price plans and historical record of Cloud Service Customers (CSCs), the model determines the amount of resources to be allocated. More loyal CSCs get better services, while for the contrary case, the provider reserves resources cautiously.**

Index Terms—**IoT; Cloud of Things; Fog computing; Edge computing; Micro Datacenter (MDC); resource management.**

I. Introduction

Connectivity has been revolutionized with the rapid development of Wireless Sensor Networks (WSNs), healthcare related services, smart phones, and other pervasive means. With the advent of IoT; devices, services, and people are ubiquitously connected almost all the time, generating a tremendous amount of data. The objective of IoT is to provide a network infrastructure with interoperable communication protocols and softwares to allow interaction and integration of physical/virtual sensors, computers, smart devices, vehicles, and dumb objects like fridge, dishwasher, microwave oven, food items, medicines, etc. [1].

The backbone of IoT communications is Machine-to-Machine (M2M), although, not limited to it. In M2M communications, two or more machines communicate with each other directly, without human interventions. IoT enables non-communicating devices to become part of the Internet and communicate through data communications means such as bar-code readers, RFID, etc. With the advancements in smartphone technology, many objects would be able to be part of IoT through various smartphone sensors. This way, non-intelligent nodes, known as "things", become communicating and data generating objects of IoT.

IoT-based services are gaining importance rapidly. Since 2011, the number of connected devices has already exceeded the number of people on Earth. Already, the number of connected devices have reached 9 billion and is expected to grow more rapidly and reach 24 billion by 2020 [2]. With the increasing number of heterogeneous devices connected to IoT and generating data, it is no more possible for a standalone IoT to perform power and bandwidth constrained tasks efficiently. IoT and cloud computing amalgamation is becoming very important [3]. There comes a situation when the cloud is connected with IoT that generates multimedia data. Visual Sensor Network or CCTV connected to cloud are examples of such a scenario. Since multimedia content consumes more processing power, storage space, and scheduling resources, it becomes important to manage them effectively to perform efficient resource management in the cloud. Specially, with mobile devices and other IoT nodes which do not have a reliable connectivity behavior cost considerably when it comes to resource allocation. Since resources are reserved but due to mobility or fluctuating behavior, if resources are given up, the datacenter has to manage the underutilization scenarios. In addition, mission critical and latency sensitive IoT services require very quick responses and processing. In that case, it is not feasible to communicate through the distant cloud, over the Internet. Fog computing plays a very vital role in this regard [3]. Fog computing refers to bringing networking resources close to the underlying networks. It is a network between the underlying network(s) and the cloud(s). Fog computing extends the traditional cloud computing paradigm to

978-1-4673-9292-1/16/$31.00 ©2016 IEEE

the edge of the network, enabling the creation of refined and context-aware applications or services [4]. Fog is an Edge computing, MDC paradigm for IoTs and WSNs.

In our previous work [13], we proposed a basic mathematical model for resource estimation for Fog. In this paper, we first extend the model to propose a customer's Probabilistic Resource Estimation (PRE) model for Fogs, which can help in efficient, effective, and fair management of resources for IoTs. Our work is mainly focused on customer type based resource estimation. We have considered different traits and characteristics of customers in this regard, which makes our model more flexible and scalable. For most of the mobile devices, like smartphones, laptops, tablet computers, where most of the time video contents would be played or cloud storage services would be used, efficient resource management is required. As a second contribution, the model presented in this paper has been validated with real traces. Our model is implemented based on the Amazon EC2[1] data storage pricing to get realistic outcome. Netflix, Dropbox, NASDAQ, iCloud, Bitcasa, etc. all use Amazon's service for cloud storage. For customer's characteristics, one of the main components is its service relinquish probability. The relinquish probability is based on the overall record of all the services a customer has been consuming as well as the record for a particular service. In this way, resources are reserved based on that record, which helps minimize resource underutilization. For relinquish probability, Crawdad's trace [15] has been utilized partially in this paper.

II. RELATED WORK

Research on Fog computing is at its beginning, therefore, no standard architecture is available regarding managing resources in the Fog. So far, most studies simply focus on resource management in the clouds. The scenario of Fog computing or Cloud of Things (CoT) has not been considered by most of the prior works.

Cubo et al. [5] discuss the integration of heterogeneous devices which are stored and accessed via the cloud. The presented work, however, lacks discussion on the key issue of management of resources for such devices in the cloud. Abu-Elkheir et al. [6] discuss management of data in IoT. The authors mention how distinctive design parameters for management of the data work. But how that data and IoT nodes are going to be handled at the cloud layer and how resources are to be managed for the generated data is not part of this study. Ning and Wang discuss in [7] the potentials of IoTs and the amount of data it is going to generate. The authors also emphasize on efficient management of resources for the future Internet, in which heterogeneous IoTs would be an essential part. Sammarco and Iera [8] analyze Constrained Application Protocol (CoAP) for IoTs and discuss service management and method for exploiting resources of IoT nodes. Chatterjee and Misra [9] provide a mapping of sensors to their respective targets through a sensor-cloud infrastructure. But how every node or sensor is allocated with resources in a dynamic fashion is not part of this study. Tei and Gurgen [10] emphasize on the importance of the integration of cloud-IoT. They discuss preliminary outcomes of a project in this domain. Distefano et al. [11] contribute in presenting a framework for the integration of the underlying IoT nodes with the cloud. However, the challenge of dynamic and node-based resource management is not a focus of this study. In [12], Rakpong et al. consider resource allocation in mobile cloud computing environment. They discuss communication/radio resources and computing resources, but their work only focuses on decision making for coalition of resources to increase service provider's revenue. Bonomi et al. [4] present a basic architecture for Fog computing, which does not include its practical implications and resource management for IoTs. Similarly, Stolfo et al. [14] present data protection through Fog computing, but do not discuss resource management and related matters.

III. FOG COMPUTING

Fog computing is a newly introduced paradigm, which extends the standard cloud computing to the edge. Therefore, it is also called Edge computing.

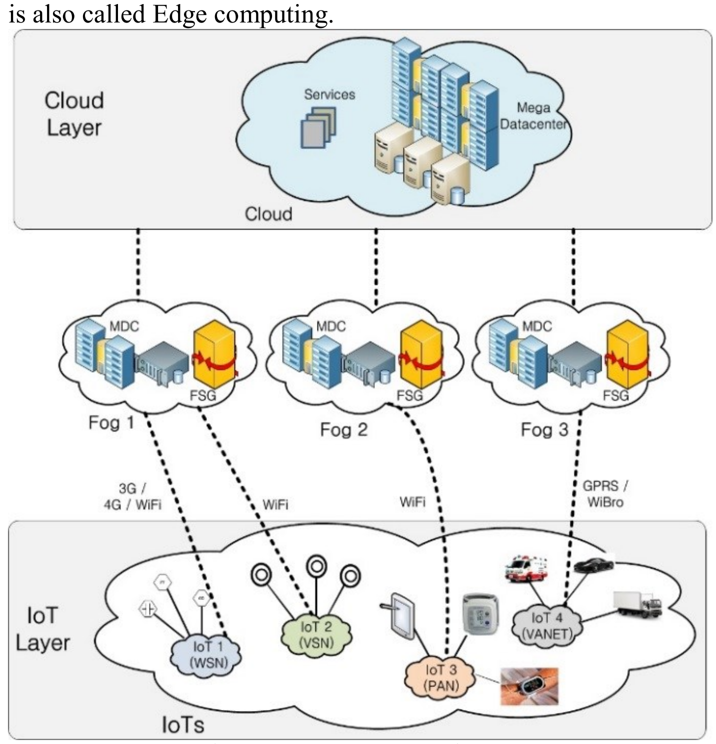

Figure 1. Fog MDC supported IoTs.

Fog is an MDC, highly virtualized platform, responsible for providing computation, storage, and networking services between the end nodes in an IoT and traditional clouds [4]. In contrast to the standard cloud, which is more centralized, Fog computing is targeted for widely distributed applications.

[1] http://aws.amazon.com/ec2

Figure 1 presents an overall architecture of Fog computing, where dedicated Fogs will be able to provide resources near the underlying networks or IoTs. As examples, WSN, Virtual Sensor Network (VSN), Personal Area Network (PAN), and Vehicular Ad hoc Network (VANET) are shown in the figure.

Fogs would be able to provide low latency and high quality streaming to mobile nodes and moving vehicles through proxies and access points positioned accordingly, like along highways and tracks. Similarly, resource and power constrained WSNs and VSNs would be able to take advantage from the presence of Fogs. Because of being localized, i.e., residing closer to the underlying IoTs, Fogs suit applications with low latency requirements, emergency and healthcare related services, video streaming, gaming, augmented reality, etc. For smart communications, Fogs are going to play an important role. Fogs are composed of MDCs, where processing, memory, virtual machines, and storage resources are available. Besides, Fogs also contain gateway(s) and are able to handle data communications in a smarter way on the basis of the requirement of the higher level applications and constraints of the underlying nodes. Such type of gateway is termed as Fog-Smart Gateway (FSG) [3]. In the Cloud-Fog-IoTs architecture presented above, the underlying nodes and networks are not always physical. Virtual sensors and VSNs are also required for various services. Similarly, temporary storage of data, preprocessing, data security and privacy, content delivery services and other such tasks can be done easily and more efficiently in the presence of a Fog. Based on the feedback from the application and depending upon the constraints of the node generating data, the FSG decides the timing and type of data to be processed in the Fog and then sent to the cloud. FSG helps in providing a better utilization of network and cloud resources.

Since Fog is localized, it provides low latency communication and more context awareness. Fog computing allows real-time delivery of data, especially for delay sensitive and healthcare related services. It can perform preprocessing and notify the cloud, after which cloud could further adapt that data into enhanced services. With heterogeneous nodes, heterogeneous types of data would be collected. In this case, interoperability and transcoding become an issue. Fogs play a very vital role in this regard. Likewise, IoT and WSN federation, in which two or more IoTs or WSNs can be federated at one point, can be made possible through the Fog. This will allow creation of rich services.

IV. PROBABILISTIC RESOURCE MANAGEMENT MODEL

Sensors, IoT nodes, devices, and CSCs contact a Fog to acquire the required service(s) at best price. CSCs perform the negotiation and SLA tasks with a Fog. Once a contract is agreed upon, the service is provided to the customer. In this regard, Fog not only provides services on an ad hoc basis, but also it has to estimate the consumption of resources, so that they can be allocated in advance. Resource estimation allows more efficiency and fairness at the time of consumption. As mentioned, the requests can be made from objects or nodes as well as devices operated by people. Therefore, prediction and pre-allocation of resources also depend upon the user's behavior and its probability of using those resources in the future. For this purpose, we formulate the estimation of required resources as:

$$\mathfrak{R} = \sum_{i=0}^{n} \sum_{k=0}^{x} \begin{cases} \left(\mho_i * (P_L - \sigma^2)\right) * (\Omega_L), if\ n = 0 \\ \left(\mho_i * (P_L - \sigma^2)\right) * (1 - \Omega_i), if\ x = 0 \\ \left(\mho_i * \left((1 - \bar{x}\,(P_i(L|H)_s)\right) - \sigma^2\right) * (1 - \Omega_i) \end{cases} \quad (1)$$

$$\mathfrak{R} \in \{CPU, storage, memory, bandwidth\}$$

$$P_i(L|H)_s = \begin{cases} \bar{x}\,(\sum_{s=0}^{n} P(L|H)_s) & if\ n > 0, \\ 0.3 & if\ n = 0 \end{cases} \quad (2)$$

$$\sigma^2 = \frac{1}{n-1} * \sum_{i=1}^{n} (x_i - \bar{x})^2 \quad (3)$$

Where $\mathfrak{R}$ represents the required resources, $\mho_i$ is the basic price of the requested service. In most of the cases, $\mho_i$ is decided at the time the contract is being negotiated. P_L is low relinquish probability, which is used when a customer is new and its historical record is not available. In such a case, it is anticipated that the customer will be somewhat reliable. $\bar{x}\,(P_i(L|H)_s)$ is the average of the Service Oriented relinquish Probabilities (SOP) of a particular customer of giving up the same resource that is being currently requested. In case the customer is requesting this service for the first time, the default value set for $\bar{x}\,(P_i(L|H)_s)$ is 0.3. Because, the average of low relinquish probability (0.1 to 0.5, from the complete range of 0.1 to 0.9) is 0.3. Requesting customer can have any probability value between 0 to 1 as low (L) or high (H) giving up probability.

$$0 < L \leq 0.5\ ,\ 0.5 < H \leq 1 \quad (4)$$

$$\Omega_i = \begin{cases} \bar{x}\,(\bar{x}\,(\sum_{k=0}^{n} P(L|H)_k)), P(L|H)_{last}) & if\ n > 0, \\ 0.3 & if\ n = 0 \end{cases} \quad (5)$$

σ^2 is the variance of the SOP. CSCs, especially mobile users, can have a fluctuating behavior in utilizing resources, which may lead to deception, while making decision about resource allocation. That is why, in our model, we have taken into account variance of relinquish probabilities, which helps determining the actual behavior of each customer.

Ω represents the overall history i.e., the Average Overall relinquish Probability (AOP). Here, it should be noted that $P_i(L|H)_s$ determines the probability of that particular service that the customer is currently requesting, while Ω is the probability which includes all the activities a particular customer has been doing. The last activity of the user in this regard tells about its most recent probability. That is why, it is given more importance and the average is taken again, by adding the last relinquish probability. In case of a new customer, when there is no historical data for that user, its default value is set to a low relinquish probability of 0.3.

V. SETUP, IMPLICATIONS, AND OUTCOME

In this section, we present the implementation results of our service model, along with the discussion on each result. We defined our service model through an algorithm to evaluate the effectiveness in CoT business. Our main objective is to observe the influence of the performance factors on the system and test the feasibility of our method on the basis of actual Crawdad trace [15] which was partially applicable for relinquish probability through the link strength and quality parameters such as the Received Signal Strength Indicator (RSSI) and the Link Quality Indicator (LQI). RSSI and LQI are typical terms in wireless communications. RSSI is a measure of Radio Frequency (RF) power of the channel, coming from WiFi, Bluetooth, or other IEEE 802.15.4 transmitters. LQI determines the quality of the link. It is a cumulative value used in multi-hop networks. Specially for PANs and WSNs, such measure is deemed important where the user is mobile and link quality fluctuates. Degradation in quality becomes a reason for service relinquishing. Crawdad's trace is a packet delivery performance over an 802.15.4 link under different stack parameter configurations for more than 6 months. "The data set consists of measurements of the data delivery performance of a WSN link. Four major performance metrics, i.e., energy, throughput, delay and loss, have been measured over 6 months for around 50 thousand parameter configurations of 7 key stack parameters. Overall, nearly 200 million packets are included in the data set" [15].

We have considered different parameters to estimate the required resources for different types of users. TABLE 1 shows the setting for the basic parameters. Since implementation on real test-beds limits the extent to the scalability, which consequently makes it difficult to reproduce the result and analyze in varied scenarios, we chose simulation instead.

TABLE 1: KEY PARAMETERS' SETTING FOR EVALUATION

Parameters	**Range**
Default SOP	0.3
Default AOP	0.3
Relinquish probability (P)	0.1 ~ 0.9
Service Price (U_i)	USD 85.3 ~ USD 1000
Default User Characteristic	$0 < L \leq 0.5$, $0.5 < H \leq 1$
Variance range	0 ~ 0.16
Minimum VRV	2.56

A. *Resource estimation for new customers*

When CSCs having different traits are requesting for a particular service, the Fog layer has to decide the amount of resources that have to be allocated for that service, based on the type of customer. For low relinquish probability CSCs, priority in resource allocation is given. For new customers (i.e. the Fog has no past record for them), the default relinquish probability value is used (0.3). In other words, the default case assumes that new customer will be "somewhat" loyal as a perfectly loyal customer would be having a probability of 0.1. Since cloud resources are precious and it is not advisable to take risks, thence, instead of assigning 0.1 probability value, we have assigned 0.3, which is the average probability of low relinquish, as explained earlier with the model. Figure 2 shows the unit of resources, we call it Virtual Resource Value (VRV), being estimated for new customers, for different types of registered services. This unit is then mapped to actual resources (memory, CPU, storage space, etc.), according to the type of service being offered and policies of a particular Cloud Service Provider (CSP). For example, a USD 85.3 cloud storage collaboration service is more I/O intensive. It requires more CPU as well as storage space. The CSP will map 8 to level one of its resource allocation actual mappings. In case the USD 85.3 service is related to database queries, then only I/O is intensive, not storage, because it requires read-only process. The CSP will perform mappings accordingly. This is how different units of resources are mapped to actual resources, based on the type of service. Similarly, for a USD 682 service, 62 units of resources are reserved.

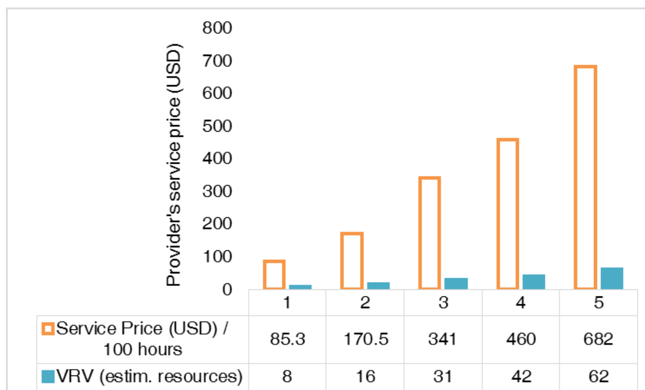

Figure 2. Resource estimation for new CSCs, for different Amazon EC2 storage services.

Figure 3 shows an illustrative scenario of how mapping can be performed by the CSP, according to its resource pool and the type of service being provided. For a Video on Demand (VoD) service, S1, VRV 8 is mapped to corresponding Resource Pool Level (RPL).

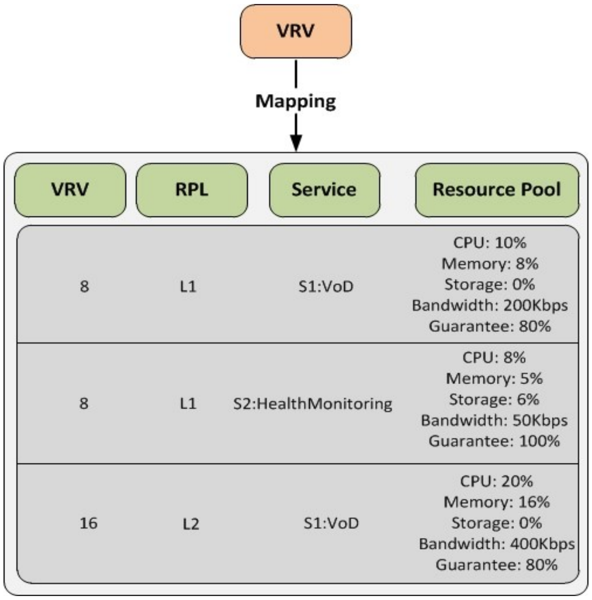

Figure 3. Illustrative scenario of mapping of virtual resource value to the resource pool, according to the type of service.

Then according to the type of service being provided, the mapping is performed to the actual resource pool. Among the available resources for service 1, CSP allocates 10% of CPU, 8% of memory, and data rate of 200Kbps. Storage is not required for this service, therefore, it is 0%. The guarantee of allocation of these resources is 80%, which means that at least 80% of the resources from the mapping are guaranteed. This is only an example. This mapping would vary according to the type of service and available resource pool of CSP.

B. *Resource estimation for an existing customer, requesting service S (Amazon EC2 USD 85.3, 1X800 SSD) for the first time*

In the scenario where a CSC has already been a customer of a CSP before, but requested a particular service S for the first time, resources are estimated differently. In this case, the CSP knows the overall behavior (i.e. AOP) of the CSC but has no historical data for service S. Therefore, the Fog layer allocates resources keeping in view the available record, but assuming that the CSC is going to be somewhat loyal in utilizing the current service S. The main idea is to incorporate the available historical data as much as possible so that the CSC is dealt with fairness while minimizing the risks for the CSP and Fog layer.

Figure 4 shows that resources are predicted on the basis of available AOP, keeping SOP to 0.3 (somewhat loyal). In case of CSC 1, when AOP is 0.1, maximum possible resource units are allocated. For this case, 23.03 resources are allocated. Resources are decreased as the relinquish probability increases. For a CSC having 0.9 AOP, 2.56 units of resources are reserved. By doing this, the Fog layer makes sure that CSCs are treated according to their reliability and Fog itself and the CSP are not deprived of the profit they deserve. Also, chances of resource underutilization are minimized.

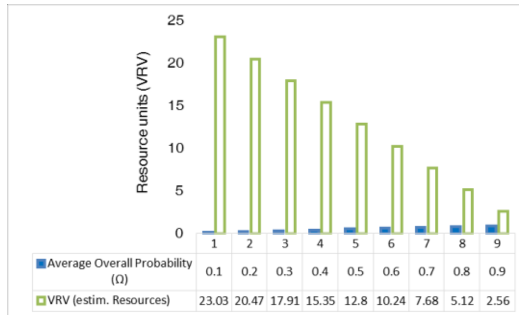

Figure 4. Res. estim. for existing CSC, requesting service S for the first time.

C. *Resource estimation for an existing customer*

For the returning/existing customers, the Fog layer already has a historical record of its past activities and probabilities (i.e. AOP and SOP). When characteristics of a particular customer are known, it is more reasonable and rational to determine and allocate resources accordingly. This way, the Fog layer and the CSP will be able to reserve the right amount of resources while minimizing their chances to lose profit. Figure 5 shows five different types of CSCs, having different SOPs and AOPs, requesting a particular service S. In this example, the result is presented for the Amazon EC2 1X800 SSD storage service, priced at USD 85.3 per 100 hours. The unit is greater for loyal (L) customers, while it is smaller for not loyal (H) customers, because of their behavior. Since there are more chances of an H customer to relinquish the service, hence, more priority and quality is provided to the more loyal customer, having L probability.

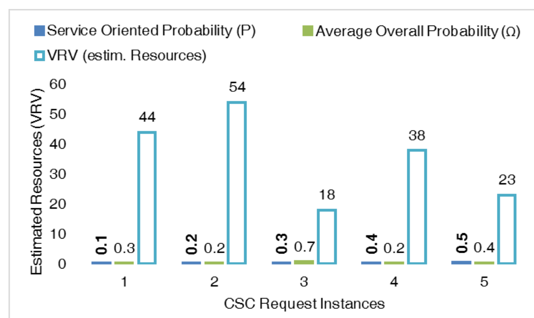

Figure 5. Res. estim. for different CSCs, for Amazon EC2 $85.3 service.

In case of CSC 1, having SOP = 0.1 (bold font in the figure) and AOP = 0.3, 44 units (VRV) of resources are reserved. In case of CSC 2, SOP = 0.2 and AOP = 0.2, 54 unit of resources are reserved. Comparing CSC 2 with CSC 4, both have the same AOP but CSC 4 has an SOP = 0.4, therefore, it gets less resources (38). This shows that both types of probabilities have their impact on estimating the amount of resources that will be allocated. This makes sure that a CSC who has generally been loyal, but not so in the case of a particular service, or vice versa, gets treated accordingly.

D. Resource estimation with variable AOP variance

As mentioned earlier, the service relinquish probability is very fluctuating in the case of IoT devices and mobile nodes. Due to this, the variance in AOP is also included in the user characteristics while determining resources. This section presents the effect of the variability in AOP variance. In this part, we fixed the SOP to 0.3 and the service price to USD 85.3 to assess the effect of AOP and its variance. Figure 6 shows that for case 1, when AOP is 0.4 and variance in AOP (shown in bold font) is 0.16, resource estimated for are 27 VRV. Cases 4 and 5 have the same AOP=0.1 but the effect of the variance is evident. For case 4, the variance is 0.04 and the estimated resources are 51 VRV. Case 5 has a variance of 0.05 and the amount of resources is decreased with the same ratio the variance increases. In this case, the estimated resources are 50 VRV.

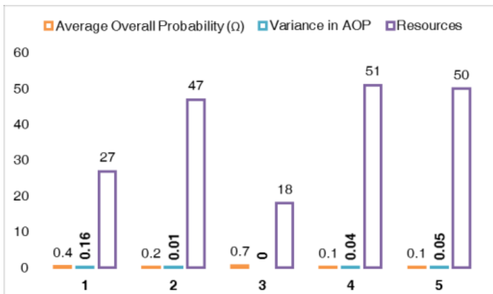

Figure 6. Effect of variance on overall resource estimation.

VI. Conclusion and Future Work

Rapidly increasing IoT-based services have triggered the need for more sophisticated ways to handle heterogeneous devices, fluctuating connectivity and data generating behaviors. Energy and resource constrained IoT nodes require their computation tasks to be offloaded. Furthermore, healthcare, emergency, and multimedia services require quick response with minimum latency. With IoT-Cloud communications, it becomes difficult to achieve that, having cloud reachable through a shared, unreliable core network. Resources are to be brought up closer to the users. Fog computing provides the solution by bringing cloud resources to the edge of the underlying IoTs and other end nodes. However, with heterogeneous devices being part of IoTs, it is difficult to predict how much resources will be consumed and whether the requesting nodes, devices, or sensors are going to fully utilize the resources they have requested. Due to this uncertainty, the probability of resource utilization, known as relinquish probability in our model, is incorporated while performing resource estimation. Our model presents user characteristic based resource management for Fogs, taking into account the type of service, the overall service relinquish probability, and the service oriented relinquish probability. This methodology helps determine the right amount of resources required, by avoiding resource underutilization and profit-cut for the CSP as well as the Fog itself. Every involved entity is treated rationally. To simulate our model in a real world environment, we have implemented it on Crawdad trace with various Amazon EC2 services, specially keeping in view the fact that many of the services a mobile user would be using, like Netflix, Bitcasa, Dropbox, iCloud, etc., they all are using Amazon for cloud storage.

For future work, we are looking at extending our model for varied scenarios, considering monetary matters, according to the types of CSCs.

References

[1] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthi, “Smart Objects and Building Blocks of Internet of Things”, IEEE Internet Computing Journal, volume 14, issue 1, pp. 44-51, Jan.-Feb., 2010.

[2] J. Gubbi, R. Buyya, Slaven Marusic, and Marimuthu Palaniswami, “Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions”, Technical Report CLOUDS-TR-2012-2, July 2012.

[3] M. Aazam, E-N. Huh, “Fog Computing and Smart Gateway Based Communication for Cloud of Things”, Proceedings of IEEE Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, pp. 27-29 August, 2014.

[4] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, “Fog Computing and Its Role in the Internet of Things”, Proceedings of ACM SIGCOMM, August 17, 2012, Helsinki, Finland.

[5] J. Cubo, A. Nieto, and E. Pimentel. "A Cloud-based Internet of Things Platform for Ambient Assisted Living", Sensors, vol. 14.8 2014, pp. 14070-14105.

[6] Abu-Elkheir, Mervat, Mohammad Hayajneh, and Najah Abu Ali. "Data management for the internet of things: Design primitives and solution." Sensors 13.11 (2013): 15582-15612.

[7] Ning, Huansheng, and Ziou Wang. "Future Internet of things architecture: like mankind neural system or social organization framework?" Communications Letters, IEEE 15.4 (2011): 461-463.

[8] Sammarco, Chiara, and Antonio Iera. "Improving service management in the internet of things." Sensors 12.9 (2012): 11888-11909.

[9] Chatterjee, Saptarshi, and Sudip Misra. "Target tracking using sensor-cloud: Sensor-target mapping in presence of overlapping coverage." Communications Letters, IEEE 18.8 (2014): 1435-1438.

[10] Tei, Kenji, and Levent Gurgen. "ClouT: Cloud of things for empowering the citizen clout in smart cities." IEEE World Forum on Internet of Things (WF-IoT), 2014.

[11] Distefano, Salvatore, Giovanni Merlino, and Antonio Puliafito. “Enabling the cloud of things”, 6th IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012.

[12] Rakpong Kaewpuang, Dusit Niyato, Ping Wang, and Ekram Hossain, “A Framework for Cooperative Resource Management in Mobile Cloud Computing”, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Vol. 31, No. 12, December 2013.

[13] M. Aazam, and E-N. Huh. "Dynamic resource provisioning through Fog micro datacenter." 13th IEEE International Conference on Pervasive Computing and Communication, 23-27 March, 2015, St. Louis, MO, USA.

[14] Salvatore J. Stolfo, Malek Ben Salem, Angelos D. Keromytis, "Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud", IEEE Symposium on Security and Privacy Workshops (SPW), 2012.

[15] Songwei Fu, Yan Zhang, CRAWDAD dataset due/packet-delivery (v. 2015-04-01), traceset: packet-metadata, downloaded from http://crawdad.org/due/packet-delivery/20150401/packet-metadata, doi:10.15783/C7NP4Z, Apr 2015.