Resource Allocation in Combined Fog-Cloud Scenarios by Using Artificial Intelligence

Masoud Abedi*, Mohammadreza Pourkiani†
*Thünen Institute of Baltic Sea Fisheries
Rostock, Germany
Email: masoud.abedi@thuenen.de
†Institute of Computer Science
University of Rostock
Rostock, Germany
Email: mohammadreza.pourkiani@uni-rostock.de

***Abstract*—Although both cloud and fog computing technologies provide great on-demand services for the users, but none of them could singly guarantee the Quality of Service for the Internet of Things (IoT) based delay-sensitive applications. Therefore, cooperation between fog and cloud servers is of great importance. In this paper, we discuss about an artificial intelligence (AI) based task distribution algorithm (AITDA), which aims to reduce the response time and the Internet traffic by distribution of the tasks between fog and cloud servers. Our case study is a delay-sensitive application that runs in a situation where the computing capability of fog servers is restricted, and the internet connection is unstable (like vessels on the oceans). The primary trial of the AITDA shows that this method noticeably reduces the response time and internet traffic in comparison to the cloud-based and fog-based approaches.**

***Index Terms*—Task Distribution, Machine Learning, Fog Computing, Cloud Computing, Maritime environments**

I. Introduction

With the advent of the IoT, the number of Internet-connected devices is getting increased. It has been estimated that more than 50 billion things will be connected to the Internet by the end of 2020, which leads to huge communication of data over the Internet [1]. Cloud data centers provide the functionality in order to store, process, and manage the increasing amount of the generated data by the IoT autonomous systems through convenient and on-demand access to a shared pool of capable resources [2]. But, communication of vast amounts of produced data by the IoT systems over the Internet (to be received by the remote servers in cloud), occupies the bandwidth significantly, which causes considerable delay in future communications. Therefore, using the traditional cloud computing is not a suitable approach for processing and storage of the considerable amounts of the generated data by the IoT devices [3]–[6]. Aiming to find a solution for the above-mentioned problems, Cisco presented the "Fog Computing" technology in 2012 [7].

Fog computing is defined as a virtualized platform, built at the edge of the network, which provides computing, storage, and networking services between the end devices and the cloud servers [1]. Reducing the Internet traffic and response time could be mentioned as the most important advantages of using fog computing. The reason of reduction in response time is obvious as in fog computing scenarios, the servers are closer to the users and they are sometimes accessible, even in the local area network. The proximity of fog servers to the users, eliminates data transmission over the Internet and provides services with low latency. Fog computing-based architectures are usually composed of three main components, namely user, fog server, and remote server [1]. The user is the data provider that sends data for processing and receives the response. The fog server is a lightweight server in the proximity of the user, which is responsible to collect, store, process, and analyze the received data from the users, aiming to provide services on-demand. The remote server is a server accessible through the Internet, which is usually used for storage of data and further processing and permanent storage. There might also be an entity called "broker" between the users and servers, which is responsible for receiving the computing tasks from the users and assigning them to the servers (figure 1 shows a combined fog-cloud architecture). However, fog computing has its own disadvantages, and the main one is the limited computing power of the fog servers. The mentioned disadvantage causes a considerable delay when the workloads of fog servers increases [8]. Therefore, in case of higher task arrival rates or requests for compute-intensive tasks, the cloud resources might provide better response times in comparison to fog servers. All in all, on one side there are cloud servers which have powerful computing resources but are located far from the users (with high latency to the users), and on the other side there are fog servers which have limited computing capabilities but are located closer to the users (with low latency to the users). Therefore, an appropriate task distribution algorithm in combined fog-cloud scenarios is required to utilize the fog and cloud resources in the most efficient way, and reduce the response time as much as possible.

This research has been funded by the Thünen Institut of Baltic Sea Fisheries (Johann Heinrich von Thünen-Institut)

978-1-7281-7216-3/20/$31.00 ©2020 IEEE

Nevertheless, where both fog and cloud resources exist in a scenario, the first challenge is that the broker is not able to distinguish the best resource (between the fog and cloud servers) for processing a batch of data. In addition, when the number of available tasks in broker increases, using only one resource (cloud or fog) is not efficient as the best response time could only be achieved when both fog and cloud servers have the least idle time. Therefore, the second problem is that the broker is not able to distribute the available tasks in the most efficient way for reducing the response time and the Internet bandwidth utilization. So, the research questions could be formed as follows: Where should the task processing take place? In fog or cloud servers? And how could it be possible to use the capabilities of both fog and cloud resources efficiently, for reducing the response time and the Internet traffic?

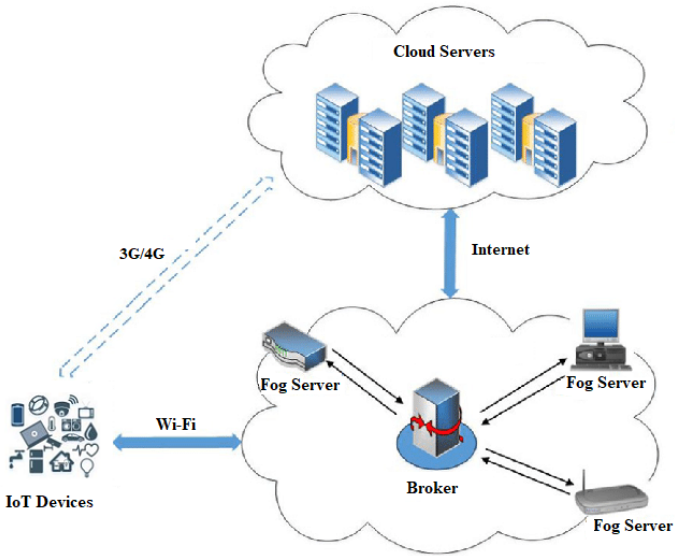

Fig. 1. Combined Fog-Cloud Architecture.

Concerning the above questions, in this paper we propose AITDA, which runs in the broker and aims to reduce the response time and Internet traffic. This goal is of great importance as reducing the Internet traffic, improves data transmission and relieves the bandwidth. Therefore the Internet bandwidth could be reserved for other important users and applications. In addition, providing reduced response time is always required by delay-sensitive applications.

Smart decision making for task distribution between fog and cloud resources becomes more critical in scenarios where the internet connection is unstable and sometimes expensive, and also where the fog resources are limited in terms of computing and storage capabilities. One of the most outstanding examples of such a scenario is the maritime environment, where the computing resources on vessels are restricted, and the accessibility to the Internet is impossible or expensive (when data must be communicated through the satellite). Therefore, in order to check the performance of the AITDA, in this paper we used a delay sensitive application as the case study, which aims to monitor the health status of people in maritime environments.

The structure of the paper is as follows. In section II, we review the previous works and discuss about the existing challenges. In section III, we present the AITDA, and in section IV, we discuss about the case study of this paper. The achieved results by AITDA are presented in section V, before the conclusion in section VI.

II. Previous Works

There are several papers in the literature in which new methods for task distribution between fog and cloud resources have been proposed. For example, in [3] authors proposed a novel mechanism named Gaussian Process Regression for Fog-Cloud allocation (GPRFCA) for implementation in infrastructures that are composed of cooperative fogs and clouds. This mechanism employs a Gaussian Process Regression to predict future demands to avoid blocking of requests, particularly those delay-sensitive ones. The effectiveness of this approach was measured in terms of energy consumption, blocking ratio, and response time. Moreover, in [8], authors proposed designing a fog-based region architecture to provide nearby computing resources. They also investigated efficient scheduling algorithms to distribute tasks among the regions and remote clouds, aiming to minimize the computation and transmission latency of all requests. Their proposed approach reduced the response time in comparison to both fog and cloud-based scenarios. In [9], a hybrid computation offloading approach is proposed where users have multiple and independent tasks that can be processed at fog nodes or a remote server in the cloud, aiming to minimize the total system cost. In addition, in [10] fog-to-cloud (F2C) computing method is presented, which is consisted of a layered management structure that can bring together different heterogeneous cloud/fog layers into a hierarchical architecture. The achieved results after implementation of the F2C show that this method is capable of reducing the response time in comparison to using traditional cloud or fog-based methods. Furthermore, authors in [11] developed a two step resource management approach for fog computing. Their proposed architecture includes the user, a home fog, backup fogs, and cloud. The first step is about the allocation of devices to the fog. Then the performance of both home and backup fogs will be monitored, aiming to choose the better server for providing a suitable response time for the user. In the second step, requests will be distributed to the allocated fogs or to the cloud (when bad performance is detected from the fog servers). This approach improved the resource allocation and provided a slightly better response time in comparison to the fog-based scenario.

A. Current Challenge

As we reviewed, different methods with specific assumptions have been proposed for efficient utilization of the fog and cloud resources, aiming to improve the response time, energy consumption, resource utilization, etc. The current challenge in the process of task distribution between the fog and cloud

servers is the variation of tasks, task arrival rate, delay, task processing time, and fog resources. These variations have not been considered in the previous works and no solution is proposed for predicting the behavior of the mentioned variables. For instance, in [8], the task processing time is considered to be known (which is not true in the real world, as the processing time could vary with regard to the workloads of servers and different types of tasks). Moreover, in [3], the workloads of servers is considered to be constant. In addition, the proposed method in [11] improves the resource allocation after several failures, which might cause an increased response time for several delay-sensitive tasks.

The hypothesis of this paper is that the prediction of the behavior of the mentioned variables would lead to a better task distribution, which finally improves the Quality of Service. For this purpose, we propose the AITDA in the next section.

III. Proposed Method

In order to deal with the above-mentioned challenge, our proposed approach is using a smart broker, which could predict the processing time and the size of the result of a received task (from the user). This prediction process could be performed by using one of the function approximation methods. We selected Artificial Neural Networks (ANN) [12], as they have been widely used in scientific researches and have also provided acceptable results. They are also ready to use and only need to be trained. In the following steps we explain how the ANNs could be implemented in the broker:

1) User sends different tasks to the broker.
2) The broker randomly sends the received tasks to the fog and cloud servers.
3) Fog and cloud servers process the tasks and for each task, they log the received data from the broker, as well as the processing time and the size of the result of the task.
4) Each of the servers train an ANN in which the input is the received data from the broker and the output is the processing time and the size of the result.
5) The ANNs will be sent to the broker (then for each server, the broker has an ANN which makes the broker able to predict the processing time and the size of the result, when a task arrives).
6) The broker distributes the tasks on basis of the estimated response times and sizes, with regard to the pre-defined policies.

Different policies could be considered for the distribution of the available tasks in the broker. The policy that we define in this paper aims to reduce the internet bandwidth utilization and response time. For this purpose, the available tasks in the broker must be sorted with regard to their predicted size of results in ascending order. Then, the smaller tasks will be assigned to the cloud, and larger ones will be sent to the fog servers for processing. In order to reduce the response time as much as possible, fog and cloud servers must process the tasks in parallel so that each of them experiences the least idle time. In order to check the performance of the AITDA, in the next section, we apply it to a real-world application.

IV. Case Study

The importance of using fog and cloud computing in maritime environments has been discussed in [13], [14]. The role of using an intelligent method like AITDA for task distribution between fog and cloud servers becomes more visible in maritime environments as in these types of scenarios, variables like delay, server workload, and request arrival rate change continuously over time. In this paper, our case study is a delay sensitive application which uses the Wireless Body Sensor Networks (WBSN) [15] for data collection and combines the proposed methods in [16], [17] to analyse the collected data by the WBSNs , aiming to predict the reaction of the human body to different environmental factors and working conditions. This application is designed for monitoring the health status of workers in severe weather conditions, such as the weather in maritime environments. As a result, this application informs the workers about their health status. Therefore, specific working schedules for any worker could be prepared, aiming to keep them safe and healthy in their working environment [18].

V. Real-World Experiment and Discussion

In order to check the performance of the AITDA, as depicted in figure 2, we created a network which was consisted of both fog and cloud layers, and we set the computing capability of the cloud layer to be twice that of the capability of the fog layer.

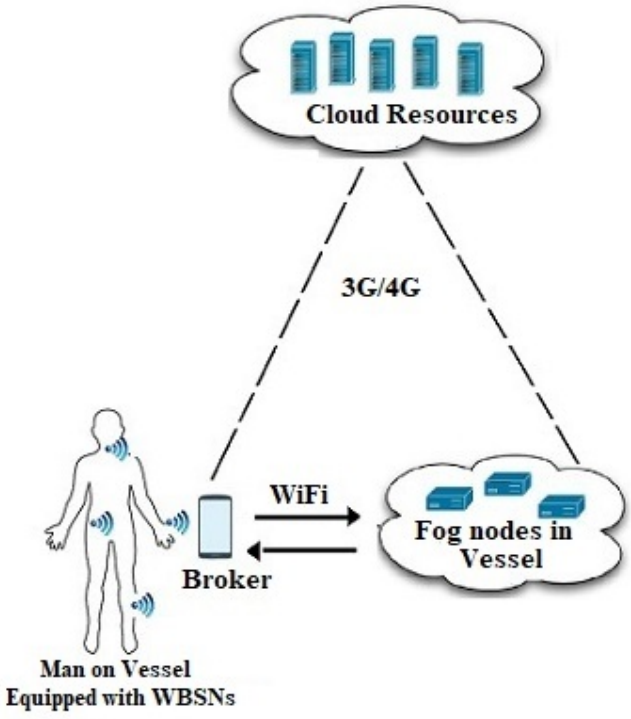

Fig. 2. The network architecture in our case study.

TABLE I
THE IMPLEMENTATION DETAILS OF OUR EXPERIMENT

	Cloud	Fog
Communication Protocol	TCP/IP	TCP/IP
Communication Between User and Server	Through the Internet	Ad-hoc
Location of Server	Internet	LAN
Available Bandwidth	4.2 Mbps	400 Mbps
Distance to User	13 hops	1 hop

In the next step, we randomly generated 1000 different tasks, of which 800 tasks were used for the full batch training process of the ANNs (in fog and cloud servers), and 200 tasks were used for the main experiment. It must be mentioned that we used multi-layer perceptron ANNs in MATLAB, which were consisted of three layers and using the Levenberg-Marquardt training algorithm. In order to check the accuracy of the ANNs, we compared the predicted amounts with the real amounts. The results showed that the ANNs were able to predict the response time and the size of the results with the MSE of 0.09 and 250 respectively. The processing time of fog and cloud servers for 200 different tasks has been depicted in figure 3, which shows that the average processing time of cloud and fog servers are almost 0.5 and 1 seconds, respectively. It must be mentioned that the average size of results is almost 90,000 byte. More details about the experiment can be found in Table I.

Then, we set up the trained ANNs in the broker and made the broker able to estimate the response time of the fog and cloud servers as well as the size of the results. Afterward, the broker distributed the tasks on basis of the discussed policy in section 3. We also assumed that both fog and cloud servers always work properly, and their workloads considered to be equal at the time of task assignment process.

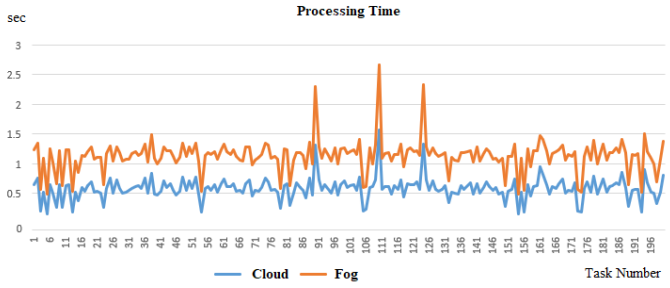

Fig. 3. The processing time of cloud and fog servers for each task.

In addition, the delay considered to be constant (1.2 seconds for the fog and 2.4 seconds for the cloud). It must be mentioned that in maritime environments different types of communication with cloud is possible (such as satellite, 3G, 4G, etc). The mentioned delay is based on using the 3G networks. As it is depicted in figures 4 and 5, we compared the performance of the AITDA with two competitors, namely cloud-based and fog-based approaches, in terms of response time and internet traffic. Figure 4 shows the fact that by increasing the number of the available tasks in the broker, cloud performs better than fog in terms of response time. The reason is the computing capability of the cloud servers, which is more than the capability of the fog servers.

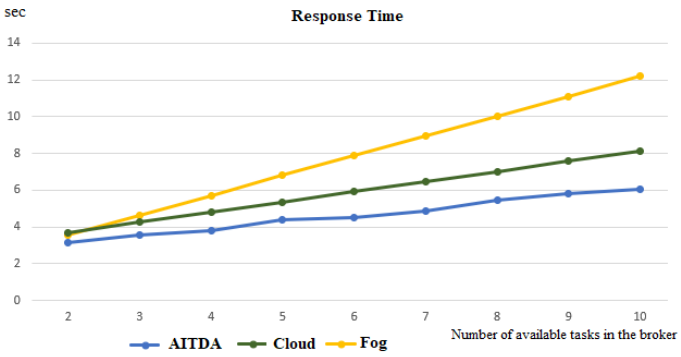

Fig. 4. Performance of different approaches in terms of response time.

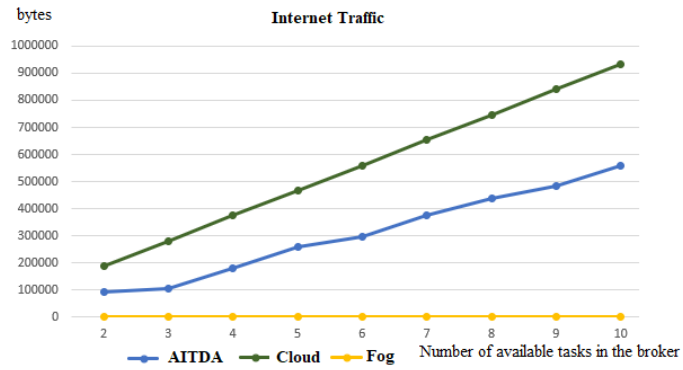

Fig. 5. Performance of different approaches in terms of Internet traffic.

Therefore, when several tasks are sent to both fog and cloud servers, the processing time of the cloud is less than the fog. This difference in processing time is high enough to cover the delay of the cloud servers. As it is illustrated in figure 4, AITDA improves the response time in comparison to the fog-based and cloud-based approaches. The reason is obvious as the AITDA distributes the tasks between the fog and cloud resources. This improvement becomes more significant when the number of available tasks in the broker increases. In addition, as it is shown in figure 5, AITDA performs better than the cloud-based method and worse than the fog-based approach in terms of Internet Bandwidth Utilization, as by using only fog servers (in the local area network) for task processing, no data would be communicated over the Internet.

Currently, because of different assumptions, a quantitative comparison of AITDA with other proposed distribution methods in the literature is not possible. But, from the qualitative point of view, AITDA has performed better than other methods in terms of improving the response time (in comparison to cloud-based and fog-based approaches).

VI. Conclusion

With regard to the increasing number of the Internet-connected devices and the production of vast amounts of data by them, using only fog or cloud computing could not satisfy the requirements of users. Therefore, the cooperation of both fog and cloud servers for providing services is of great importance. In this paper, we proposed using AITDA as an intelligent task distribution algorithm between fog and cloud servers. We used a delay-sensitive application as our case study, which is used in maritime environments for monitoring the health status of people on vessels. The achieved results showed that the AITDA is capable of reducing the response time and Internet traffic noticably, in comparison with fog-based and cloud-based approaches. It must be discussed that the impact of our proposed method becomes clearer when the number of available tasks in broker (which must be sent to the servers for processing) increases. For the future works, we suggest using feature selection and online training approaches for reducing the number of inputs and also improving the accuracy of the function approximation method, respectively. Moreover, we recommend evaluation of the performance of the other function approximation methods for predicting the response time and the size of the results. In addition, the different workloads of fog and cloud servers must be considered in the future works.

Acknowledgment

We would like to thank the Johann Heinrich von Thünen Institute (Federal Research Institute for Rural Areas, Forestry and Fisheries) for funding of this research. In addition, Mohammadreza Pourkiani is supported by Landesgraduiertenförderung Mecklenburg-Vorpommern.

References

[1] F. Andriopoulou, T. Dagiuklas, and T. Orphanoudakis, "Integrating iot and fog computing for healthcare service delivery," in *Components and services for IoT platforms*. Springer, 2017, pp. 213–232.

[2] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," 2011.

[3] R. A. da Silva and N. L. da Fonseca, "Resource allocation mechanism for a fog-cloud infrastructure," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[4] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog computing: Towards minimizing delay in the internet of things," in *2017 IEEE international conference on edge computing (EDGE)*. IEEE, 2017, pp. 17–24.

[5] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 3909–3914.

[6] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *IEEE Network*, vol. 28, no. 4, pp. 46–50, 2014.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

[8] D. Hoang and T. D. Dang, "Fbrc: Optimization of task scheduling in fog-based region and cloud," in *2017 IEEE Trustcom/BigDataSE/ICESS*. IEEE, 2017, pp. 1109–1114.

[9] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. Leung, "Hybrid computation offloading in fog and cloud networks with non-orthogonal multiple access," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 154–159.

[10] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 120–128, 2016.

[11] O. Fadahunsi and M. Maheswaran, "Locality sensitive request distribution for fog and cloud servers," *Service Oriented Computing and Applications*, vol. 13, no. 2, pp. 127–140, 2019.

[12] J. Zou, Y. Han, and S.-S. So, "Overview of artificial neural networks," in *Artificial Neural Networks*. Springer, 2008, pp. 14–22.

[13] T. Yang, R. Wang, Z. Cui, J. Dong, and M. Xia, "Multi-attribute selection of maritime heterogenous networks based on sdn and fog computing architecture," in *2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2018, pp. 1–6.

[14] M. Cankar and S. Stanovnik, "Maritime iot solutions in fog and cloud," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*. IEEE, 2018, pp. 284–289.

[15] K. Hasan, K. Biswas, K. Ahmed, N. S. Nafi, and M. S. Islam, "A comprehensive review of wireless body area network," *Journal of Network and Computer Applications*, vol. 143, pp. 178–198, 2019.

[16] A. Gagge, J. Stolwijk, and B. Saltin, "Comfort and thermal sensations and associated physiological responses during exercise at various ambient temperatures," *Environmental Research*, vol. 2, no. 3, pp. 209–229, 1969.

[17] M. Salloum, N. Ghaddar, and K. Ghali, "A new transient bioheat model of the human body and its integration to clothing models," *International journal of thermal sciences*, vol. 46, no. 4, pp. 371–384, 2007.

[18] M. Pourkiani, M. Abedi, and M. A. Tahavori, "Improving the quality of service in wbsn based healthcare applications by using fog computing," in *2019 International Conference on Information and Communications Technology (ICOIACT)*. IEEE, 2019, pp. 266–270.