Response to Editor and Reviewers' Comments

Manuscript ID: CLCO-D-20-00276
Journal: Journal of Cloud Computing: Advances, Systems and Applications (JoCCASA)
Title: **A Novel Approach for IoT Tasks Offloading in Edge-Cloud Environments**
Authors: Jaber Almutairi, and Mohammad Aldossary

Dear JoCCASA Editor,

First and foremost, the authors would like to register their profound gratitude to the handling editor and reviewers for the timely, meticulous, and professional assessment of our manuscript.

The revisions made in the revised manuscript accompanying this report are highlighted in **red**.

Meanwhile, in this report, all contents that are quoted from the revised manuscript are presented in italics and, where required, certain parts are underlined for emphasis.

While satisfied and elated with the enhanced version of the manuscript, the authors unequivocally acknowledge and credit the erudite reviewers who through their thorough criticisms guided us on the improvements therein.

Sincerely Yours,

Mohammad Aldossary

1. Comments by Editor

1.1 Comment 1

"The reviewers all agree with the novelty of the proposed work. However, some efforts are required to improve the work as suggested by the reviewers: 1) more details and motivations of the proposed algorithms should be provided. 2) The writings should be significantly improved. More comments can be found from the reviewer reports.".

Response/Action taken: The authors would like to start by appreciating the time, effort, and diligence invested by the esteemed editor not only in selecting experts that have helped guide the revisions undertaken on our manuscript but also making time to provide his/her technical expertise to improve our study.

While apologizing for the poor comprehensibility in our previous effort to explain the motivations of our study. Prompted by this comment from the erudite editor, we revised relevant sections of the manuscript to clarify the motivations of the proposed algorithms with more details. In addition, the proposed algorithms are rewritten in more detail to clarify their motivations. These revisions traverse the entire revised manuscript but a few of which are quoted below for the kind consideration of the distinguished editor.

Section 4, Page 10:

"First, the edge controller receives the IoT devices information summary from edge computing nodes including the number of connected IoT devices, task length, the required number of cycles for task, and deadline requirement to complete task. In addition, the status of computation resources at each edge node is monitored periodically. Subsequently, the edge controller computes and decides the optimal strategy for scheduling and assigning the computation tasks to the best server (i.e., one of the edge computing server nodes or the cloud server) for execution based on the gathered information through Fuzzy logic and task scheduling algorithms. Later, we present the proposed algorithms in detail."

Section 3, Page 8:

"In practice, the edge computing nodes are connected through an intermediate layer (for example, backbone router) that serves as a central control manager to monitor them. In addition, software-defined network (SDN) technology can be utilized at this layer to monitor and manage the application services between the edge computing nodes depending on data gathered, in which where SDN has a global view of the network and is capable of making more efficient and precise decisions."

Section 5, Page 15:

"Algorithm 2 shows the detailed processes to assign each computation task to the appropriate computational resources. The process for assigning computation tasks is as follows. Firstly, the set of application, computation tasks, and the available computational resources at edge node are gathered. Then, as shown in line 1, the computation tasks are sorted in descending order regrading their CPU requirements, in which the heavy tasks come first and the lightweight tasks come last. In addition, as shown in line 2, the available computational resources (VMs) of edge node are sorted in descending order regrading their CPU capabilities (i.e, number of cores), where the most powerful VMs comes first. After that, the algorithm iterates over the ordered tasks and assigns each application task to the appropriate computational resources, where the heavy tasks could to be assigned to the powerful VMs, shown in line 3-11. This ensures that heavy tasks have the

priority to be assigned to a powerful VM, thereby will produce less processing time.”

2. Comments by Reviewer 1

2.1 Opening remarks

“A very interesting and well organised manuscript introducing a novel approach for offloading tasks on Edge-Cloud environments. Only the following minor comments should be considered by the authors..”

***Response/Action taken**:* The authors are humbled by the kind comments from the reviewer. Guided by his/her feedback we have invested a lot of effort to improve the previous submission.

2.2 Comment 1

“- on 1st page, line 54, change "enables" to "enabling"
- on page ii, line 16 "uploading/download" to "upload/download"
- on page ii, lines 19-21, the first sentence is not completed. It should be edited.
- on page v, line 58, "dose not addressed" to "is not addressed"
- on page vii, line 38 "have connected" to "are connected"
- on page viii, line 10 "layer responsible for managing and assign" to "layer is responsible for managing and assigning"
- on page xviii, lines 50-51, change "should consider" to "should be considered"
- on page xix, lines 52 and 53, "failures due to" to "failures are due to".
- on page xxi, lines 57-60, the second sentence of the second paragraph should be paraphrased to read correctly.
- on page xxii, lines 38 and 49, omit "that".”

***Response/Action taken**:* The vigilance and attention to detail of the reviewer are truly appreciated. In addition, the authors apologize for the poor quality and formatting of the manuscript. The paper has been carefully revised by a professional language editing service to improve grammar and readability. Further, all the mentioned issues were addressed.

2.3 Comment 2

“On page xi, line 12, clarify what is VM Utilisation? is it a combined utilisation for all the VM's resources (processor, memory etc)?”

***Response/Action taken**:* The erudite reviewer’s comment on the choice of VM Utilization for our approach is truly appreciated. To clarify, in this study, the VM utilization gives us information on the residual computational capacity (processor) of the edge server. However, the other metrics (i.e., memory, disk, etc) will be combined and considered in future work.

2.4 Comment 3

“Figure 4 on page xii, what is the x value for the Task Delay Sensitivity (d). Milliseconds?”

***Response/Action taken**:* The lack of clarity regarding the execution of load balancing is regretted. In this study, the delay sensitivity indicates the tolerance of the task to take a longer time to execute either due to network conditions or due to server utilization levels. In our model, this value is a ratio between (0-1) that is assumed to be determined by the application profile. However, the x value for the Task Delay Sensitivity (d) can be (%) and it added in Figure 4.

2.5 Comment 4

“Section 6.2 "Results and Discussion", it mostly provides observations but further justifications for each of the observed results should be added..”

Response/Action taken: The authors acknowledge the presentation issue of the performance results emanating from the study. Guided by this perceptive comment from the reviewer, in the revised version, we have clarified the performance results of our paper. In addition, large areas of subsection 6.2 were rewritten with a more detailed discussion of the performance contribution of this study in comparison with other benchmark solutions. We have added clarifications to convey this intuition in the revised manuscript as quoted below for the consideration of the esteemed reviewer.

Section 6.2, Page 17:

“The main performance metric is the service time, since the end-to-end service time of an offloading task is most significant for IoT latency-sensitive application. Figure 8 shows the average service time for the 4 different approaches versus the number of IoT devices, in which the service time is composed of processing and network time. The purpose of the experiments was to enhance the resources management in Edge-Cloud system in order to reduce latency for IoT applications. It is seen from the figure that all the approaches have nearly the same performance when the system is unloaded. However, with increasing the number of IoT devices, the service time of the proposed approach remains steady compared to the other approaches. Moreover, the service time for Flores algorithm increases rapidly after the number of IoT devices exceeds 1400. Whereas, Utilization-Based and Sonmez algorithms nearly have the same performance. This is attributed to the usability of VM utilization in task scheduling policy which will avoid processing delays and then produce less service time.

The average network time of all approaches related to different numbers of IoT devices is shown in Figure 9. It is observed from the figure that all the approaches have the same network time when the system is stable, whereas, the results are differentiated after the system overloaded. Also, the utilization-based approach can provide the lowest network time compared to the other approaches. This is due to the demands associated with increased time regarding the processes of communication when the task might be sent to the cloud for execution, particularly with the larger number of IoT devices.”

Section 6.2, Page 18:

“Similarly, Figure 10 assesses the processing time of the four approaches versus different numbers of IoT devices. It is seen from the figure that when the number of IoT devices is less than 1400, all the approaches approximately have the same processing time. However, as the number of IoT devices further increases, the performance of the Flores approach decorates in comparison with Sonmez and Utilization-based approaches while our proposed model remains steady and outperforms the others. This is traced to the shorter processing time as the computation tasks are assigned to the appropriate resources. Whereas, Flores

approach offloads the task to the edge whenever possible without considering whether the resource is overloaded, thereby leads to an increase in the processing time.

In the EdgeCloudSim, the task failure can be happened due to different reasons such as the lack of computational resources at the VM (e.g., overloaded VM) and congested network. Therefore, task failure is considered an important performance metric in order to evaluate the offloading approach. In the following, the evaluation of task failures for our approach will be divided into two parts, system stable and system overloaded.

First, in the case of system stability, the proposed approach has the lowest percentage while the other approaches have nearly the same performance, and around 0.5% of tasks will fail, shown in Figure 11. This is due to that our approach considers the required amount of data to be uploaded and downloaded. On the other hand, when the system load is high (the second part), it can be seen that the lowest task failure average is for the proposed approach, as shown in Figure 12. Interestingly, there were slight differences between Utilization-Based and Flores for all number of IoT devices. When the system load is low, most of the tasks failures are due to network issues such as losses of the packet [9], but when the system is overloaded, task failures are due to the lack of computation (e.g., unsuccessful completion task) as well as network issues. In comparison, the proposed approach has the lowest task failures because it assigns the heavy tasks to the powerful VM as well as considers other factors (e.g., VM utilization, network demand and delay sensitivity).

Finally, the preliminary analysis of the average VM utilization at edge servers versus a different number of IoT devices is shown in Figure 13, where the system server IoT devices less than 1000. It can be seen that the utilization level of all the approaches at 200 devices is similar, while its value is changed when the number of devices increased. In addition, the proposed approach is keeping the utilization level relatively low comparing to other approaches when the number of devices increased.

On the other hand, as shown in Figure 14, when the system load is high, the proposed approach was the lowest compared to other algorithms. This is because it trades utilization for reduced service time. Also, it can be seen that Sonmez and the proposed approach were relatively similar and lower than the others. Flores was the highest and we can link that with the results of failed tasks because it assigns the tasks to a highly utilized VM (overloaded VM). Moreover, the proposed approach succeeded in avoiding reaching the exponential deterioration when the computational resources reach their limit comparing to other existing approaches. When the resources reach their limit, this will increase the overall service time and task failure due to insufficient computational resources."

3. Comments by Reviewer 2

3.1 Comment 1

"There are still some typos or grammar errors in this paper, such as:
Page 1 line 53-54: by enables (enabling) on-demand access
Page 1 line 57: (is) required to be
Page 2 line 33: lead to select (selecting)
Page 4 line 32: computational (task)
Page 4 line 51: had aim(aimed)
Page 5 line 58: dose (is) not addressed."

Response/Action taken: The vigilance and attention to detail of the reviewer are truly appreciated. In addition, the authors apologize for the poor quality and formatting of the manuscript. The paper has been carefully revised by a professional language editing service to improve grammar and readability. Further, all the mentioned issues were addressed.

3.2 Comment 2

"*The Introduction section should* be reorganized to present the context, existing solutions of this research in a clear brief manner."

Response/Action taken: The authors apologize for the poor organization of Intordution section in the previous version of the manuscript. In the revised version we have reorganized the introduction section where the problem, main motivations and contributions for our work have been clarified. The esteemed reviewer is respectfully invited to assess the revised section.

3.3 Comment 3

"All the results were carried out using simulation and there is no empirical evaluation of the solutions in real edge/cloud environment. This makes it hard to justify the applicability. During performance evaluation, the time cost of the task scheduling and offloading process should also be taken into consideration."

Response/Action taken: The authors thank the reviewer for pointing out the issue regarding the simulation and real-world hardware setting. In this study, we used the simulation due to its low cost and this is considered as the first step in this work. In addition, we plan to apply this work to real-world hardware as future work. Finally, our simulation applied different scenarios with different parameter settings and the simulation is executed and the average value for each scenario is computed to measure the difference between our work and the other benchmark solutions.

4. Comments by Reviewer 3

4.1 Opening remarks

"*This work adopts fuzzy logic algorithms to minimize the overall service time for latency-sensitive applications. It* considers considering application characteristics, resource utilization and resource heterogeneity. The simulation experiments show that the proposed approach outperforms other related approaches.

List of main strengths of the paper
- The paper adopts fuzzy logic algorithm as a new solution to offload the tasks.
- The evaluation includes multiple works.."

***Response/Action taken**:* The authors would like to start by thanking the reviewer for his positive feedback.. Furthermore, we thank him/her for the time, effort and diligence invested to guide the revision of our study. The authors note and appreciate the concerns raised by the learned reviewer, which we invested a lot of effort to address as enumerated in the responses below as well as responses to comments from the other learned reviewers.

4.2 Comment 1

"The logic of the paper is not clear. Recommend the author to rewrite the Intro part and describe the problem clearly."

***Response/Action taken**:* The authors apologize for the poor organization of Introduction section in the previous version of the manuscript. In the revised version we have reorganized the introduction section where the problem, main motivations and contributions for our work have been clarified. The esteemed reviewer is respectfully invited to assess the revised section.

4.3 Comment 2

"The writing is poor. Authors need to revise the paper carefully."

***Response/Action taken**:* The vigilance and attention to detail of the reviewer are truly appreciated. In addition, the authors apologize for the poor quality and formatting of the manuscript. The paper has been carefully revised by a professional language editing service to improve grammar and readability.

4.4 Comment 3

"The scheduling problem described by authors in edge-cloud system is the same as the cloud system. Many previous works in cloud system have revealed how to use fuzzy logic algorithm for tasks scheduling. In other words, even though the authors claimed that they deployed fuzzy logic algorithm in edge-cloud system early, the main contribution of the paper only verifies that algorithm also works well in edge-cloud system.."

***Response/Action taken**:* The poor clarity regarding describe the scheduling problem is truly regretted. Actually, in our approach, the computation tasks can be scheduled based on MAPE (Monitoring, Analyzing, Planning and Executing) method, in which edge controller entity can perform this operation as follows. First, the edge controller receives the IoT devices' information summary from edge computing nodes including the number of connected IoT devices, task length, the required number of cycles for task, and deadline requirement to complete task. In addition, the status of computation resources at each edge node is monitored periodically. Subsequently, the edge controller computes and decides the optimal strategy for scheduling and assigning the computation tasks to the best server (i.e. one of the edge computing server nodes or the cloud server) for execution based on the gathered information through Fuzzy logic algorithm. A few of the revisions related to this comment are quoted below for the attention of the learned reviewer.

Section 4, Page 10:

"First, the edge controller receives the IoT devices information summary from edge computing nodes including the number of connected IoT devices, task length, the required number of cycles for task, and deadline requirement to complete task. In addition, the status of computation resources at each edge node is monitored periodically. Subsequently, the edge controller computes and decides the optimal strategy for scheduling and assigning the computation tasks to the best server (i.e., one of the edge computing server nodes or the cloud server) for execution based on the gathered information through Fuzzy logic and task scheduling algorithms. Later, we present the proposed algorithms in detail."

4.5 Comment 4

"In the Intro, the authors emphasized the overhead of multi-objective optimization like "Thus, solving this problem with traditional methods at the edge nodes could add extra overhead, which ...", and drew a conclusion in the following paragraph directly using "Therefore, Fuzzy Logic is considered to be among the most feasible solutions...". I suggest authors should introduce fuzzy logic first before drawing that conclusion."

Response/Action taken: The lack of clarity regarding using Fuzzy Logic is regretted. The authors appreciate the opportunity provided by this important observation by the distinguished reviewer to improve it. In the revised version, we have added a new paragraph to introduce the fuzzy logic.

Section 1, Page 3:

"Fuzzy Logic (FL) is a method of reasoning that seems closer to the way our brains work. The concept of fuzzy logic is to abstract the problem complexity to a level that can be understood. It helps to model imprecision and uncertainty of the system, where it can define the imprecise information in a more logical and meaningful fashion. It can also handle system uncertainty by dealing with many input and output variables and can represent the problem with simple if-then rules. Many researchers in the field of the distributed systems use fuzzy logic to deal with the challenges caused by vagueness, uncertainty and the dynamicity of the environment. Therefore, in this study, Fuzzy Logic is considered to be among the most feasible solutions for a multi-objective optimization problem when the activity of multiple parameters is significant. It can be easily adapted to the dynamicity of computational resources and application parameters as well as providing scalability within the context of the system. It also averts computational complexity and can provide decisions very quickly. As a consequence, fuzzy logic has been adopted in this research to determine where to offload the tasks based on application and system parameters. To the best of our knowledge, this is one of the early attempts to design and implement such a system with regards to application's demands, edge-cloud resource utilization and resource heterogeneity by adopting fuzzy logic."

4.6 Comment 5

"The design of the system architecture confuses readers. The edge controller (EC) which makes the offloading decision can be deployed in any layers between edge and cloud. In the Intro, the authors claimed that the offloading management is heavy in edge node. It is not clear why not deploy EC in the cloud directly? Please give more details.."

Response/Action taken*:* The lack of clarity regarding the design of system architecture is regretted. The authors appreciate the opportunity provided by this important observation by the distinguished reviewer to improve it. We note that, in practice, the edge computing nodes are connected through an intermediate layer (for example, backbone router) that serves as a central control manager to monitor them. In addition, software-defined network (SDN) technology can be utilized at this layer to monitor and manage the application services between the edge computing nodes depending on the data gathered. We have added a new paragraph (in Section 3) to clarify this perception in the revised version. We are truly appreciative of this guidance.

Section 3, Page 8:

"In practice, the edge computing nodes are connected through an intermediate layer (for example, backbone router) that serves as a central control manager to monitor them. In addition, software-defined network (SDN) technology can be utilized at this layer to monitor and manage the application services between the edge computing nodes depending on data gathered, in which where SDN has a global view of the network and is capable of making more efficient and precise decisions."

4.7 Comment 6

"The evaluation released that the network becomes an overhead when the number of IoT devices increase. However, it only accounts for less than 10% of scheduling time. Maybe authors should consider other evaluations, such as how the complexity of multi-objective problem affect the performance of algorithms. Some more experiments can be conducted to make the experiments more comprehensive."

Response/Action taken*:* I would like to thank the reviewer for the comment.
The authors totally agreed that considering other evaluations, such as "how the complexity of multi-objective problem affect the performance of algorithms" will make the experiments more comprehensive.

Unfortunately, the author who is responsible to do that is currently out of the country and I had a hard time contacting him. Thus, we are unable to run more experiments at this stage.

4.8 Comment 7

"Moreover, "Prposed Approach" in Figure 8, 9 and 10 should be corrected as "Proposed Approach"."

Response/Action taken*:* The authors apologize for the mistaike of the Figs. 8, 9 and 10. All figures in the revised manuscript have been carefully revised and the "proposed" word has been corrected. We appreciate and credit the distinguished reviewer for these improvements.

Comments by Reviewer 4

5.1. Opening remarks

“This paper deals with IoT offloading tasks in edge-cloud computing. The authors propoesd an algorithm for task offloading combining with fuzzy logic. The approach contains three parts: the first part is to monitor information of server and applications; the second part is to decide where to allocate those offloading tasks considering multiple resources. The decision is made by fuzzy logic system. It seems that fuzzy logic is good at solving this kind of problem; the last one is to assign tasks to appropriate resources.”

Response/Action taken*:* The authors would like to start by thanking the reviewer for his positive feedback.. Furthermore, we thank him/her for the time, effort and diligence invested to guide the revision of our study. The authors note and appreciate the concerns raised by the learned reviewer, which we invested a lot of efforts to address as enumerated in the responses below as well as responses to comments from the other reviewers.

5.2. Comment 1

“Why the authors only considering the CPU? How about ram or bandwith?.”
Response/Action taken*:* The erudite reviewer’s comment on the choice of VM Utilization for our approach is truly appreciated. To clarify, in this study, the VM utilization gives us information on the residual computational capacity (processor) of the edge server which is considered the most important metric. However, the other metrics (i.e., memory, disk, bandwidth etc.) will be combined and considered in future work.

5.3. Comment 2

“Besides, how to guarantee the resource utilization of servers in the assigning progress.”
Response/Action taken*:* The authors are humbled by the kind comments from the reviewer. In our work, the resource utilization of edge and cloud servers are guaranteed, in which the status of computation resources is monitored periodically and then used in task assignment as follows. First, the computation tasks are sorted in descending order regrading their CPU requirements, in which the heavy tasks come first and the lightweight tasks come last. In addition, the available computational resources (VMs) of edge node are sorted in descending order regrading their CPU capabilities, where the most powerful VMs comes first. After that, the ordered tasks are iterated and assigned to the appropriate computational resources, where the heavy tasks could be assigned to the powerful VMs. This ensures that heavy tasks have the priority to be assigned to a powerful VM, thereby will produce less processing time.

5.4. Comment 3

“The task scheduling algorithm is too simple and needed to modify.”
Response/Action taken*:* The poor completeness regarding the task scheduling algorithm is truly regretted. In the revised version, we have modified it and rewrote it with more details. We are truly appreciative of this guidance!

Section 5, Page 15:
“Algorithm 2 shows the detailed processes to assign each computation task to the appropriate computational resources. The process for assigning computation tasks is as follows. Firstly, the set of application, computation tasks, and the

available computational resources at the edge node are gathered. Then, as shown in line 1, the computation tasks are sorted in descending order regrading their CPU requirements, in which the heavy tasks come first and the lightweight tasks come last. In addition, as shown in line 2, the available computational resources (VMs) of edge node are sorted in descending order regrading their CPU capabilities (i.e., number of cores), where the most powerful VMs comes first. After that, the algorithm iterates over the ordered tasks and assigns each application task to the appropriate computational resources, where the heavy tasks could be assigned to the powerful VMs, shown in lines 3-11. This ensures that heavy tasks have the priority to be assigned to a powerful VM, thereby will produce less processing time."

"The computational complexity for this algorithm is $O(n^2)$, where n denotes the number of computation tasks. This can be analyzed as follow. Firstly, the computa- tion tasks and computation resources are sorted, in which $O(n^2)$ is the worst-case time complexity of the sorting. Then, the computation tasks of IoT applications are iterated to be assigned to computational resources, thereby $O(n^2)$ is the time complexity. Consequently, the overall time complexity is $(O(n^2)+ O(n^2))$ which is $O(n^2)$, where the constant is removed."