To Offload or Not? An Analysis of Big Data Offloading Strategies from Edge to Cloud

Singh, R., Kovacs, J. and Kiss, T.

https://doi.org/10.1109/AIIoT54504.2022.9817276

To Offload or Not? An Analysis of Big Data Offloading Strategies from Edge to Cloud

Raghubir Singh[†], Jozsef Kovacs[*†], Tamas Kiss[†]
Centre for Parallel Computing
Department of Computer Science and Engineering
University of Westminster, London, UK
[*†] ELKH SZTAKI, Budapest, Hungry

***Abstract*—Major research efforts have been recently made to develop resource orchestration solutions to flexibly link edge nodes with centralised cloud resources so as to maximise the efficiency with which such a continuum of resources can be accessed by users. In this context, we consider the case of Big Data analytics in which total task completion time reductions can be achieved by routing tasks initially to edge servers and subsequently to cloud resources. We demonstrate that the task complexity of the computational jobs, the Wide Area Network (WAN) speed and the potential overload of edge servers (as reflected by CPU workloads) are crucial for achieving total task completion time reductions by offloading from edge to cloud resources. The edge-cloud orchestrators are situated in the edge nodes and, therefore, require continuous access to the parameters of WAN speeds (and their fluctuations), edge server CPU workload and the task complexities in Big Data analytics requirements to make accurate edge-to-cloud offloading decisions. With favourable values for these three parameters, large reductions in completion times can result from transfer of large-scale data from edge nodes to cloud resources, which can reduce the completion times by up to 97% and meet client deadlines for computational tasks with responsive and agile solutions.**

***Keywords*—Application-level orchestration, Cloud-to-Edge continuum, Big Data analytics, WLAN, WAN, Computational complexity, Server workload**

I. Introduction

THE relatively close proximity of Edge Computing nodes to users in comparison with the distance to consolidated data centres in Cloud Computing has generated considerable recent interest in understanding how Internet of Things (IoT) devices could benefit from accessing edge servers rather than relying solely on cloud resources to process data [1]–[8].

With Big Data applications, however, the advantages of using edge nodes for computation are less clear. The relatively small computing resources of edge nodes compare negatively with the much greater cloud resources [9], [10]; on the other hand, data transmission times to may be prohibitively lengthy to those superior cloud computing resources while job queuing and other delays may also adversely affect total task completion times.

In this paper, we directly analyse task completion time advantages when Big Data requests are alternately processed by edge nodes or by transfer from edge nodes to cloud resources (Figure 1). We focus on calculations of absolute times for data transmission and computation to identify the conditions for shorter total task completions using the alternatives of edge-only and edge-to-cloud modes (Figure 1).

Furthermore, we consider parameters of data transmission speeds, task computational complexity and edge server overload (as measured by CPU workload) and analyse how time advantages of using cloud resources for Big Data applications can be eroded or negated. Additionally, we are also exploring the complexity of the interactions of these parameters with numerical simulations of the practical scenarios where large data sets are processed off-site is represented by Figure 1.

II. Related Work and Contributions

EdgeCloudSim simulator tool has been proposed as a means of supporting experimentation and finding optimal solutions for tasks when both sets of resources are available [9], [10]. Using EdgeCloudSim, an edge-cloud orchestration decision maker could handle up to 250 individual edge devices such as mobile phones with only short queuing delays; however, no clear advantage of using cloud resources for task completion times was demonstrated [9]. Handling up to 750 devices was considered in [10] but the focus was on task failures and network delays rather than on task completion times.

Other studies have modelled edge-cloud interactions for specific use-cases on cloud servers, has been proposed. An edge-cloud computing system to detect non-mask wearers in urban settings, using the relatively poor computing resources of edge servers to reduce operating demands on cloud servers has been proposed in [11]. A sequential use of, firstly, edge servers and, subsequently, cloud resources is considered to be beneficial for healthcare diagnostics in [12]. For Big Data

applications, a model was proposed in which decision makers were introduced into both the edge servers and cloud resources to manage traffic from smartphones, laptops and tablet computers as well as sensors in IoT in [13]; again, the focus of this study was on minimising latency in how the system responded to requests from end users in [13].Any form of data migration from edge nodes to cloud resources potentially introduces latencies and delays and this has been a concern for modelling studies.

In recent years, considerable research effort has been given to devising orchestration solutions for multi-layered edge-cloud (or cloud-to-edge) environments [14]. The most recent of these is MiCADO-Edge, which has a wider range of attributes than previous solutions, and which has been evaluated in video processing and secure healthcare data analysis applications [14].

A. Contributions

In this paper, we propose a computational offloading model wherein we seek to minimise the completion time of all jobs in a multi-user multi-Edge-Cloud set-up minimise the total task completion time for large datasets in either edge or edge-cloud scenarios (Figure 1). The contributions of this paper are twofold:

- A mathematical model is proposed, which is suitable for distributed deployment at the edge-cloud network, and that uses local knowledge to handle each individual task to investigate better solutions for job allocations in Edge or Cloud servers.
- Different link speeds are investigated to determine whether or not to offload each job, and selection mechanisms for edge servers to which jobs are initial offloaded are proposed and evaluated.

The remainder of the text is organised as follows. Section II states the problem formulation and the methodology used. Section III presents quantitative outcomes from data simulations with ranges of data transmission speeds, computational complexities and edge server CPU workloads. Section IV draws conclusions and outlines possible future work in this field.

III. Problem Formulation

We consider the scenario where a service provider orchestrates transfer of data files from edge servers to cloud servers for time-limited processing and when a time advantage for processing exists by such edge-to-cloud data transfer. A typical multi-edge server network with a set of tasks, each with a given number of job requests, is shown in Fig. 1.

Let $\mathcal{J}_e$ be the set of jobs on edge server (ES) e, and E be the set of edge servers in the network. Let $u_{j,c}$ be a binary variable that models the offloading decision of a job j on the cloud c. Mathematically, this is modelled as follows:

$$u_{j,c} = \begin{cases} 1 & \text{job } j \text{ offloaded to cloud } c \\ 0 & \text{compute locally on an edge server} \end{cases}$$

Let $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{C}$ be the set of links that connect edge servers to the cloud. Let $\mathcal{N}$ be the number of links to cloud c and let $\mathcal{C}$ be the cloud c in the network.

Table I: Notations used in the paper.

Symbol	Definition
D_j^e	the data size of a job j on the Edge server e
λ^e	the application complexity on the ES e in bits/instructions
α_e	The computing capability of ES i in instructions/sec
β_c	The computing capability of Cloud c in instructions/sec
T_e^{ES}	Total computational time to execute job j on ES e in sec
$T_{i,c,j}$	Transmission time of offloading job j from edge server i to cloud c in sec
T_c^{C}	Total computational time to execute jobs on Cloud c in sec
Π	Proportion of data size reduction after processing on Cloud c
η_c	the application complexity on a cloud c in bits per seconds.
$T_{e,c}$	Transmission time of offloading job j from edge server e to cloud c in sec
γ^{DL}	Downlink speed to e and c in bits/second
γ^{UL}	Uplink speed to c and e in bits/second
T^{Total}	The total completion time to process jobs j
$\mathcal{N}$	The number of links to cloud c
$\mathcal{C}$	The cloud c in the network
E	The set of edge ES servers in the network
$\mathcal{J}_e$	The set of jobs on edge server e

A. Edge Server Computational Time

Let D_j^e be the data size of a job j on the Edge server e. Let λ_e be the complexity of the application that is being executed on the ES e. The authors of [15] used a similar approach to determine the job completion time on edge nodes. Thus, we used a comparable method to calculate the processing time of edge servers. Let α_e be the on-board processor speed of ES

(in instructions per second). The total local computational time T_e^{ES} is defined as follows:

$$T_e^{\text{ES}} = \sum_{j \in J_e} \frac{D_j^e (1 - \sum_{e \in E} u_{j,c})}{\alpha_e \lambda_e} \tag{1}$$

Note that the maximum value that the inner summation can take is 1 and that is when a job j is offloaded to cloud c, otherwise the summation is zero. Following two constraints ensure that a job is solved by the edge-cloud system.

$$\sum_{e \in \mathcal{E}} u_{j,c} \leq 1 \quad \forall\ j \in \mathcal{J} \tag{2a}$$

$$\sum_{j \in \mathcal{J}} u_{j,c} \leq 1 \quad \forall\ c \in \mathcal{C} \tag{2b}$$

B. Cloud Computational Time

If a job j is offloaded to cloud c then there are three periods of time that we need to consider: time for offloading, time for processing a job on the Cloud server and time to send the result back to the source. Let β_c be the on-board processor speed of cloud c (in instructions per second). Let λ_c denote as the application complexity in bits per seconds. The total processing time to compute offloaded jobs on a Cloud server c is given as follows:

$$T_c^{\text{C}} = \frac{\sum_{j \in J} D_j^e U_{j,c}}{\beta_c \lambda^c} \tag{3}$$

Let γ^{UL} be the up-link speed in bits/second. The following equation gives the time required to transfer the job from edge to the cloud:

$$T_{i,c,j} = \frac{\sum_{j \in J} U_{j,c} D_j^e}{\gamma^{\text{UL}}} \tag{4}$$

Let Π cloud c. Let γ^{DL} be the downlink speed in bits/second. The time required to transfer the processed data from cloud to edge can be calculated as follows:

$$T_{c,i,j} = \frac{\sum_{j \in J} \Pi D_j^e}{\gamma^{\text{DL}}} \tag{5}$$

Where eqs. 3, 4 and 5 can be represented as: Π be the proportion of data size reduction after processing on Edge e, γ^{DL} and γ^{UL} are the downlink and uplink speed, respectively.

$$T^{\text{Total}} = \max \left\{ \underbrace{\max_{e \in E} \{T_e^{\text{ES}}\}}_{\text{Edge Time}}, \underbrace{\max_{c \in C} \left\{ T_c^{\text{C}} + \sum_{e \in \mathcal{ES}} T_{e,c} \right\}}_{\text{edge-cloud maximum Time}} \right\} \tag{6}$$

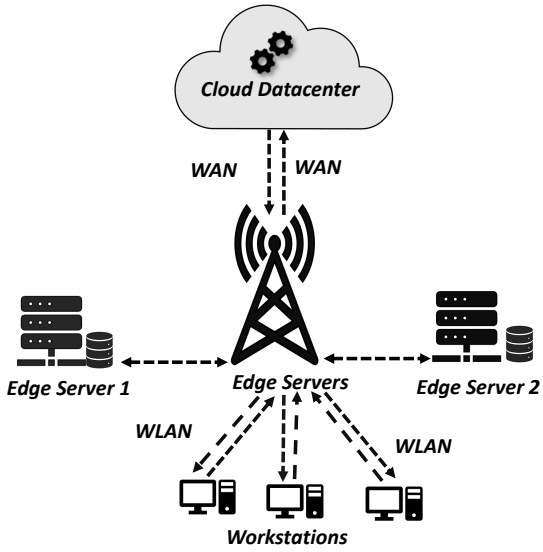

Figure 1: Model of edge computation and edge-to-cloud offloading for Big Data applications

There are two components in Equation 6: the first component defines the edge computation time; the second component defines the cloud server computational time which includes the time for transmission and receiving.

The client can be and enterprise which uploads datasets initially to an edge node where an assessment is made of achievable time advantages to be gained by further transfer to cloud resources. The edge network involves a Wireless Local Area Network (WLAN) in and through which the enterprise's workstations transfer data to edge servers; beyond the WLAN, edge to cloud transfer is performed by a Wide Area Network (WAN) connecting the edge servers to physically distant remote consolidated data centres which house cloud resources.

Based on knowledge accessible to the edge resource controller, the total task completion time (per GB of data) is computed as the sum of sequential computed times [16]:

1) The time required to transfer data from enterprise's work stations the to the edge server(s);
2) The time required for processing by one or more edge servers;
3) The time required to transfer data back from the edge server(s) to the enterprise's work stations; quantitatively, the returned data load is assumed to be 10% of the data uploaded.
4) The time required to transfer from the edge node to cloud resources.
5) The time required for processing in the cloud.
6) The time required to transfer data from the cloud to the edge node.

Table II: Parameters used for numerical simulations

Entity	Parameter	Value	Unit
ES	α_e	1.36×10^{11}	IPS
CS	β_c	2.72×10^{12}	IPS
App	λ	0.0000529 - 0.00227	bpi
Network	WAN	5-50	Mbps
	WLAN	50	Mbps

The total time for task completion using the edge node is the sum of (1), (2) and (3). The total task completion time using edge-cloud-client transfer, i.e., uploaded data from the enterprise's work stations to the edge servers, onward to cloud resources and finally returned (as fully processed results) to the enterprise's work stations is the sum of tasks (1), (4), (5), 6), (3). If this sum is less than the sum of tasks (1), (2), (3), the service provider may opt to use cloud resources; if the client has imposed deadline times for data processing (for example seconds per GB of data) in the Service Level agreement reduced time using edge to cloud transfer may prove beneficial.

An orchestration software that makes the offloading decision is, therefore, assumed to be based in edge nodes and can assess the times required for processes (1) – (6) from knowledge of processor and data transfer times (per GB) accessible in the edge node, cloud and client.

IV. Numerical Simulations

Using the mathematical methodology described in Section II, numerical simulations were performed to identify the conditions under which task completion time advantages could be gained by migrating tasks from edge nodes to cloud resources. The set of parameters used in these simulations is listed in Table II. These values and ranges were used to investigate the influence of the three factors (data transfer speed, computational complexity and edge server overload) on potential task completion time advantages from edge-to-cloud task transfer. The computational complexity values were taken for nine scientific apps from [15]. The nine programs identified were scientific programs of varying complexity suitable for modelling Big Data analytics. The computational complexity is an arithmetical means of converting bits (from bytes of data size per file) to instructions for a computational program: dividing bits by bits per instruction (bpi). Using this process, bits are converted into numbers of instructions [15]. With chosen values of instructions per second as computing speeds [15], the computation times for tasks can be calculated [15]–[17]. In general, the smaller the bpi value of an application, the more computationally complex is the task.

The WAN speeds indicated in Table II are mean speeds

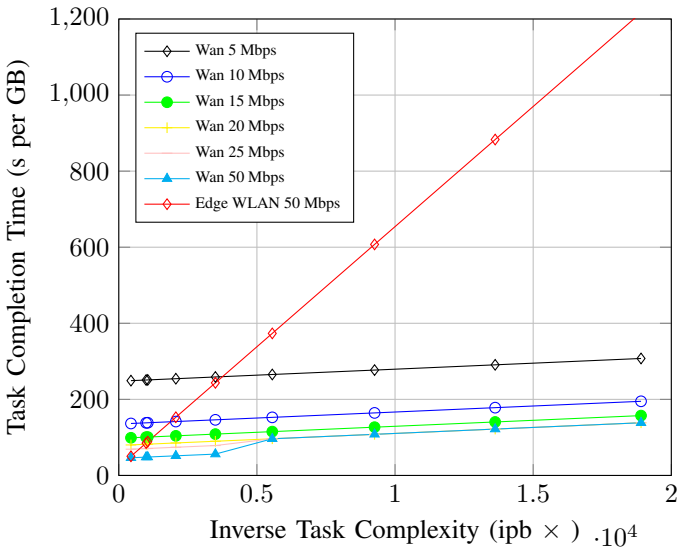

Figure 2: Effect of task computational complexity on total task completion times at an edge node (WLAN 50 Mbps) or by edge-to-cloud data transfer (WAN 5-50 Mbps).

throughout the time required for total task completion in either edge-only or edge-cloud models. While WAN speeds might be variable on short time scales, we have used mean values for completion times which might exceed 300s at the lowest WAN speed considered. When operational problems reduce WAN speeds for extended periods, this could reduce the benefits of edge-cloud transfer but would be automatically assessed as part of the edge-cloud decision making process

A. Relations between computational complexity and WAN speed

The results are summarised in Figure 2 for a WLAN of 50 Mbps. As the task complexity increases – using inverse bpi values to generate a left-to-right x-axis – the task completion time lengthens, driven by the longer processing times required. At the lowest computational complexity, cloud processing is only faster at a WAN speed of 50 Mbps (Table III) . At the highest four computational complexities, edge-to-cloud data transfer is advantageous even at the lowest WAN speed (5 Mbps); edge processing is between 1.4 and 4 times longer (the sum of data transfers and computational processing times). With progressively lower computational complexities, edge processing becomes competitive even with progressively increasing WAN speeds and eventually is advantageous compared with edge-to-cloud data transfer at all but the fastest considered WAN speed (50 Mbps) (Table III).

Very similar results were obtained with WLAN speeds across the range of 5 – 300 Mbps included in this study; in all instances, edge-to-cloud data transfer was advantageous at a WLAN of 50 Mbps. Thus, it can be observed that the WLAN speed to the edge node is not an important parameter because

Table III: Effect of WAN speed and task complexity on total task completion time

Task Complexity (bpi)	WAN 5 Mbps Edge-Cloud (% of edge time)	WAN 10 Mbps Edge-Cloud (% of edge time)	WAN 20 Mbps Edge-Cloud (% of edge time)	WAN 50 Mbps Edge-Cloud (% of edge time)
0.002270	495	271	159	92
0.001010	295	163	96	57
0.000953	283	156	93	54
0.000484	166	93	56	34
0.000286	106	60	37	23
0.000180	71	41	26	26
0.000108	46	27	18	18
0.000073	33	20	14	14
0.000053	25	16	11	11

client-to-edge transfer times are also incorporated into both edge and edge-cloud hybrid processing. Moreover, decreasing the data returned to 1% of that transmitted also had no effect on the minimum WAN speeds required for shorter task completion times by edge-to-cloud data transfer, although total completion times decreased.

If a client sets a maximum acceptable time, this deadline time is a function of both computational complexity and the WAN speed. For example, a deadline time of $<$300 (s) per GB for the most complex task would be met by all WAN speeds. For a task with a lower complexity and with a deadline time of $<$100 (s) per GB, a minimum WAN speed for redirection to the cloud would be required to meet the requirements.

In general, the computationally simplest tasks benefit from the "proximity" of Edge Computing – part of the vision of moving away from distant consolidated cloud data centres. With more complex apps, the superior computing power of cloud resources become apparent. Formally, the decision-making process for edge-to-cloud transfer is:

$$\sum T_1 - T_3 > \sum T_1, T_3 - T_6 \tag{7}$$

In the inequality represented by equation (7), the left-hand side represents the total task completion time using the edge node only while the right-hand side represents the total task completion time using edge-cloud transfer.

An orchestration mechanism would take the information and redirect tasks from edge to cloud (or vice versa); in practice, variable WLAN and WAN speeds may occur - especially the WLAN speed if the edge node becomes overloaded - and decision making would need to be both flexible and responsive.

B. Edge CPU workload

The effects of edge node congestion can be readily modelled by increased edge server CPU workload. Following the procedure of [9], the cloud servers are assigned zero CPU workload and edge-to-cloud data transfer incurs negligible server CPU workload.

At a parameter choice of a WLAN speed of 50 Mbps and a WAN speed of 15 Mbps, tasks with high complexities use edge-to-cloud data transfer for faster total processing (Figure 3). If the edge server CPU workload is then increased in the range 10% to 65%, at an edge server CPU workload of 65%, edge-to-cloud data transfer yields shorter total completion times at all computational complexities, i.e., even with tasks with the lowest complexities considered (Table IV).

Table IV: Effect of edge server CPU workload on task completion time (WLAN 50 Mbps, WAN 15 Mpbs)

Task Complexity (bpi)	Edge-Cloud CPU 10%(% of edge time)	Edge-Cloud CPU 20%(% of edge time)	Edge-Cloud CPU 50%(% of edge time)	Edge-Cloud CPU 65%(% of edge time)
0.002270	185	173	127	97
0.001010	109	100	68	50
0.000953	105	96	65	48
0.000484	62	56	37	26
0.000286	41	36	23	17
0.000180	28	25	16	11
0.000108	19	17	11	7
0.000073	14	13	8	6
0.000053	12	10	7	5

If the WAN speed is increased to 25 Mbps, only the least complex tasks do not benefit from edge-to-cloud data transfer (Figure 4); however, tasks of all computational complexities are processed faster by edge-to-cloud data transfer at an edge server CPU workload of 45% (Table V).

Conversely, at a much slower WAN speed of 5 Mbps, only the most computationally complex tasks benefit from edge-to-cloud data transfer (Figure 5); very high (88%) edge server CPU workloads were found to be necessary for task of all computational complexities to have shorter total task completion times by edge-to-cloud data transfer (Table VI).

C. WAN propagation delay and job queuing

Our analysis assumes uninterrupted data transmission with no delays or task queuing. With large quantities of data for processing, edge-cloud transfer $>$ 1200 (s) and cloud processing times $>$ 300 (s) for the tasks with the greatest computational complexity render WAN propagation delays of 100 ms [9] as minor. Only if severe network congestion occurs

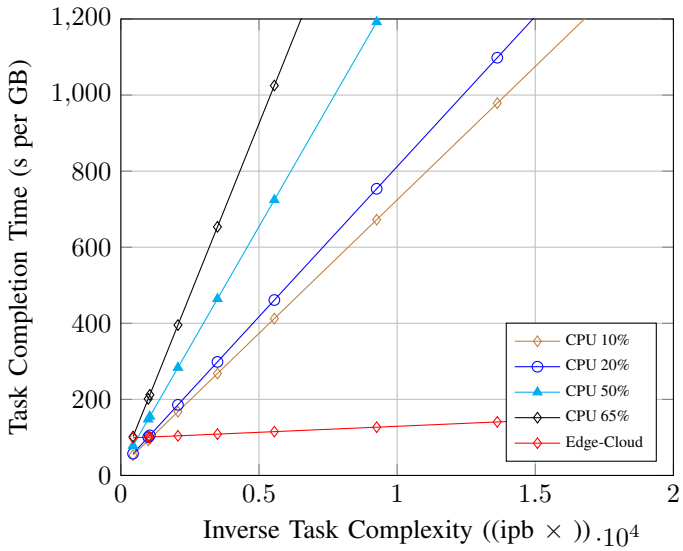

Figure 3: Effect of edge server CPU workload on total task completion times with a WLAN speed of 50 Mbps) and edge-to-cloud data transfer (WAN) of 15 Mbps.

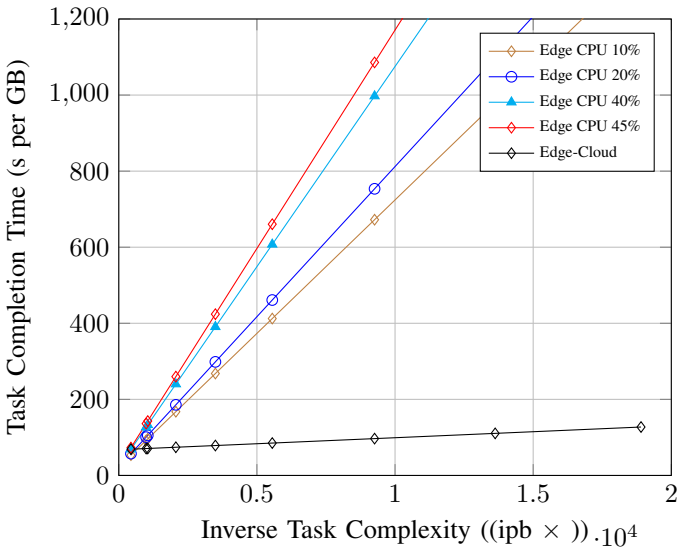

Figure 4: Effect of edge server CPU workload on total task completion times with a WLAN speed of 50 Mbps) and edge-to-cloud data transfer (WAN) of 5 Mbps.

Table V: Effect of edge server CPU workload on task completion time (WLAN 50 Mbps, WAN 25 Mpbs)

Task Complexity (bpi)	Edge-Cloud CPU 10%(% of edge time)	Edge-Cloud CPU 20%(% of edge time)	Edge-Cloud CPU 50%(% of edge time)	Edge-Cloud CPU 65%(% of edge time)
0.002270	129	120	100	94
0.001010	77	70	56	52
0.000953	74	67	53	50
0.000484	44	40	31	29
0.000286	29	26	20	19
0.000180	21	18	14	13
0.000108	14	13	10	9
0.000073	11	10	8	7
0.000053	9	8	6	6

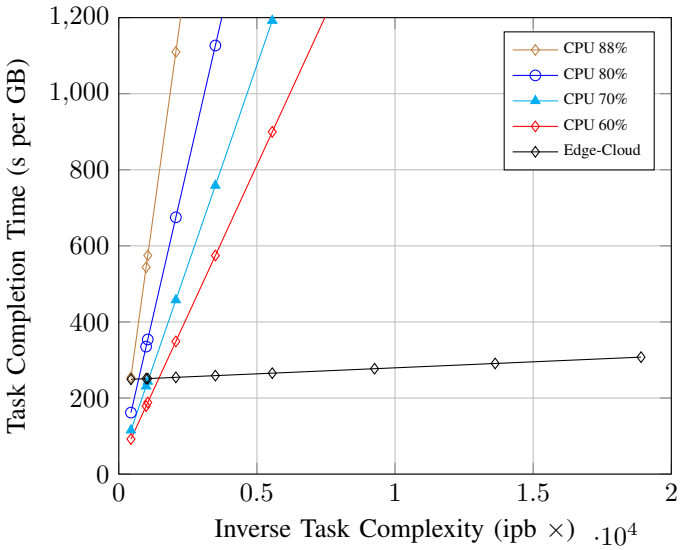

Figure 5: Effect of edge server CPU workload on total task completion times with a WLAN speed of 50 Mbps) and edge-to-cloud data transfer (WAN) of 25 Mbps.

would WAN propagation from edge nodes be incompatible with total task time reductions.

Considering tasks with low computational complexity, however, the total tasks completion times using cloud resources become $<$ 5 (s) and any job queuing of this magnitude becomes important, i.e., with $<$ 5 (s) completion times. Total queuing times in cloud computing environments of $<$1 (s) have been claimed [18] and would not influence our findings. Only in case of seriously impaired or badly functioning edge nodes, networks and/or cloud servers would result that propagation delays and job queuing become relevant factors.

V. Conclusions and Future Work

Our results show that the problem of whether or not to migrate data from edge nodes to cloud resources is soluble if sufficient parametric data are available to the decision maker: transfer speeds, processor speeds, task complexity and the congestion status (CPU workload) of the edge node. Computationally, the offload decision is taken when the total task completion time is shorter when using cloud resources, i.e., equation (7).

Table VI: Effect of edge server CPU workload on task completion time (WLAN 50 Mbps, WAN 5 Mpbs)

Task Complexity (bpi)	Edge-Cloud CPU 60%(% of edge time)	Edge-Cloud CPU 70%(% of edge time)	Edge-Cloud CPU 80%(% of edge time)	Edge-Cloud CPU 88%(% of edge time)
0.002270	271	216	154	98
0.001010	140	109	75	46
0.000953	133	103	71	44
0.000484	73	56	38	23
0.000286	45	34	23	14
0.000180	29	22	15	9
0.000108	19	14	9	6
0.000073	13	10	7	4
0.000053	10	8	5	3

For the large data files required in Big Data applications, edge nodes offer time advantages for processing within the edge WLAN network for tasks of low computational complexity because data transfer (especially at 5G speeds) is much shorter than with WAN transfer to distant consolidated processing centres. This conclusion is in line with the widely promoted features of the various forms of Edge Computing [19]

As computational complexity increases, however, our analysis shows that edge-to-cloud data transfer becomes increasingly attractive to reduce total task completion time and meet any stipulated job deadline times. This requires edge-cloud and cloud-edge data transfer times to be sufficiently short not to introduce delays that would eliminate any advantages of the much greater computational capacity (as reflected in job processing times) of cloud resources. With high task complexities and high WAN speeds, edge-cloud synergy can result in total task completion times approaching 10% of those required by processing in the edge node only (Table 3). The magnitude of these time reductions would be a major factor in efficient task processing in Big Data analysis.

Our analysis shows that the WAN speed is critical to the decision-making process: slow WAN speeds (in the 5 Mbps range) render edge-cloud orchestration unsuccessful in achieving shorter total task completion times for low-complexity tasks while much faster WAN speeds can give total task time reductions even with the tasks of the lowest computational complexity. Any orchestration mechanism must have full access to WAN speeds, especially in periods of WAN speed reduction, when edge solutions may prove superior.

Conversely, our analysis shows that as the edge node becomes congested, data transfer to the cloud for processing becomes increasingly beneficial to achieve shorter task completion times. Depending on the precise combination of task complexity, WAN speed and edge server CPU workload, total task completion times could be reached which were considerably less than 10% of those required by processing in the edge node only (Tables 4-6). Again, this increased efficiency would be valuable for Big Data applications.

Edge-to-cloud data transfer is likely to be a functioning mechanism to avoid service delays but this is critically dependent on three factors: task complexity, WLAN speed and WAN speed. While the task complexity is set by the client's requirements, network speeds and edge node congestion will be expected to be variable and an efficient decision-making system relies on full and continuous access to all relevant parameters.

Future work will focus on applying the findings of this paper in practice when developing an offloading strategy for an edge-to-cloud orchestration solution. To achieve this, various offloading policies will be developed for the MiCADO-Edge orchestrator [14] to support various application scenarios on multiple heterogeneous edge-cloud networks for automated scalability and, subsequently, incorporating the price cost of using an edge-cloud service for users of Big Data analytics.

REFERENCES

[1] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[2] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa, "Fogflow: Easy programming of iot services over cloud and edges for smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 696–707, 2017.

[3] S. Ghosh, A. Mukherjee, S. K. Ghosh, and R. Buyya, "Mobi-iost: mobility-aware cloud-fog-edge-iot collaborative framework for time-critical applications," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2271–2285, 2019.

[4] T. Jing, T. Jia, J. Yutong, Z. Ning *et al.*, "Application of cloud edge collaboration architecture in power iot," in *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, vol. 1. IEEE, 2020, pp. 18–22.

[5] N. Waranugraha and M. Suryanegara, "The development of iot-smart basket: Performance comparison between edge computing and cloud computing system," in *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*. IEEE, 2020, pp. 410–414.

[6] M. Abbasi, E. Mohammadi-Pasand, and M. R. Khosravi, "Intelligent workload allocation in iot–fog–cloud architecture towards mobile edge computing," *Computer Communications*, vol. 169, pp. 71–80, 2021.

[7] J. Wu, G. Zhang, J. Nie, Y. Peng, and Y. Zhang, "Deep reinforcement learning for scheduling in an edge computing-based industrial internet of things," *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[8] D. Wu, X. Huang, X. Xie, X. Nie, L. Bao, and Z. Qin, "Ledge: Leveraging edge computing for resilient access management of mobile iot," *IEEE Transactions on Mobile Computing*, 2019.

[9] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3493, 2018.

[10] T. A. Zaitoun, M. B. Issa, S. Banat, and W. Mardini, "Evaluation and enhancement of the edgecloudsim using poisson interarrival time

and load capacity," in *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, 2018, pp. 7–12.

[11] D. Yang, T. Yang, F. Gao, P. Shi, and S. Liang, "The application of the edge-cloud computing system based on reinforcement learning in large-scale mask recognition," in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*. IEEE, 2020, pp. 1756–1759.

[12] G. Sirisha, A. M. Reddy *et al.*, "Smart healthcare analysis and therapy for voice disorder using cloud and edge computing," in *2018 4th International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. IEEE, 2018, pp. 103–106.

[13] M. S. Thangam and M. Vijayalakshmi, "Data-intensive computation offloading using fog and cloud computing for mobile devices applications," in *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 2018, pp. 547–550.

[14] A. Ullah, H. Dagdeviren, R. C. Ariyattu, J. DesLauriers, T. Kiss, and J. Bowden, "Micado-edge: Towards an application-level orchestrator for the cloud-to-edge computing continuum," *Journal of Grid Computing*, vol. 19, no. 4, pp. 1–28, 2021.

[15] S. Melendez and M. P. McGarry, "Computation offloading decisions for reducing completion time," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2017, pp. 160–164.

[16] R. Singh, S. Armour, A. Khan, M. Sooriyabandara, and G. Oikonomou, "Identification of the key parameters for computational offloading in multi-access edge computing," in *2020 IEEE Cloud Summit*, 2020, pp. 131–136.

[17] R. Singh, S. Armour, A. Khan, M. Sooriyabandara *et al.*, "The advantage of computation offloading in multi-access edge computing," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2019, pp. 289–294.

[18] T. S. Sowjanya, D. Praveen, K. Satish, and A. Rahiman, "The queueing theory in cloud computing to reduce the waiting time." *International Journal of Computer Science Engineering & Technology*, vol. 1, no. 3, 2011.

[19] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Computer Networks*, vol. 130, pp. 94–120, 2018.