Edge-Cloud Synergy: Unleashing the Potential of Parallel Processing for Big Data Analytics

Singh, R. and Kiss, T.

Edge-Cloud Synergy: Unleashing the Potential of Parallel Processing for Big Data Analytics

Raghubir Singh, Tamas Kiss
Centre for Parallel Computing
Department of Computer Science and Engineering
University of Westminster, London, UK

***Abstract*—If an edge-node orchestrator can partition Big Data tasks of variable computational complexity between the edge and cloud resources, major reductions in total task completion times can be achieved even at low Wide Area Network (WAN) speeds. The percentage time savings are greater with increasing task computational complexity and higher WAN speeds are required for low-complexity tasks. We demonstrate from numerical simulations that low-complexity tasks can benefit either by task partitioning between an edge node and multiple cloud servers. The orchestrator can also achieve greater time benefits by rerouting Big Data tasks directly to a single cloud resource if the balance of parameters (WAN speed and the ratio between edge and cloud processing speeds) is favourable.**

Keywords—Big Data, Edge Computing, Cloud Computing, edge-to-cloud orchestration, Wide Area Network, Wireless Local Area Network, computational complexity

I. Introduction

THE synergy of combining edge computing nodes with cloud computing resources is being increasingly studied in networks with a wide range of motives, ranging from that of increasing the performance of resource-constrained mobile devices [1] to optimisation tasks in global e-commerce [2] Edge-cloud orchestration has been explored with a wide variety of computational and IT scenarios: real-time vehicle route management [3], wearable device communication and Internet of Things (IoT) data processing [4], geolocated deployment of edge computing services [5], large-scale mobile IoT applications [6], [7], edge node resource management [8], [9], IoT device and application deployment [10]–[12] and video processing and secure healthcare data analysis applications [13].

Recently, we have demonstrated the great reductions in task processing times if Big Data analytics can be flexibly moved from edge nodes to cloud resources if the combination of task complexity, processing powers, data transfer rates and edge node congestion are recognised [14]. In this paper, we explore how the parallel processing abilities of edge nodes and cloud servers can be combined to optimise computing performance, especially when data transfer rates are major constraints.

978-1-6654-6316-4/22/$31.00 ©2022 IEEE

A. Motivation and Related Work

Rather than visualising edge and cloud resources as alternatives, they can be explored as parallel tracks in a spatially large computing network. Specifically, different quantities of a Big Data analytics tasks could be partitioned into allocate portions and our analysis is focused on establishing under what circumstances total task completion times could be minimised and what effects physical parameters such as data transfer rates could have an orchestration decision making. Task partitioning is a topic that has been explored in Edge Computing where offloading efficiency is optimisable by, for example, the use of Artificial Intelligence [15], [16] or where multiple mobile devices and edge servers are combined [17], [18].

Edge-cloud orchestration aims to balance demand from an end user with the supply of appropriate services (in this case, computational power and capacity for Big Data analytics) by matching service deployment and service delivery in terms of a Service Level Agreement (SLA) [19]. We focus on minimising total task completion time in accordance with presumed SLA requirements and we base calculations to a per GB base (from which all conclusions can be scaled up to actual end user demands). We assume further that the SLA for an enterprise client will place restrictions on edge and cloud resource use for security reasons [20]: rather than operating with multiple edge servers and cloud data centres [21], we restrict the simulation analysis to one edge server and up to three cloud resource centres identified and specified by a SLA (Figure 1). Finally, we do not consider the end user demand to include real-time manufacturing systems to avoid latency issues and the construction of smart monitoring-analysis-planning-execution in closed loops [22] and the SLA prohibits any but transient data storage off site to follow specified security protocols [23].

B. Contributions

Following these principles, our main contributions in this paper are:

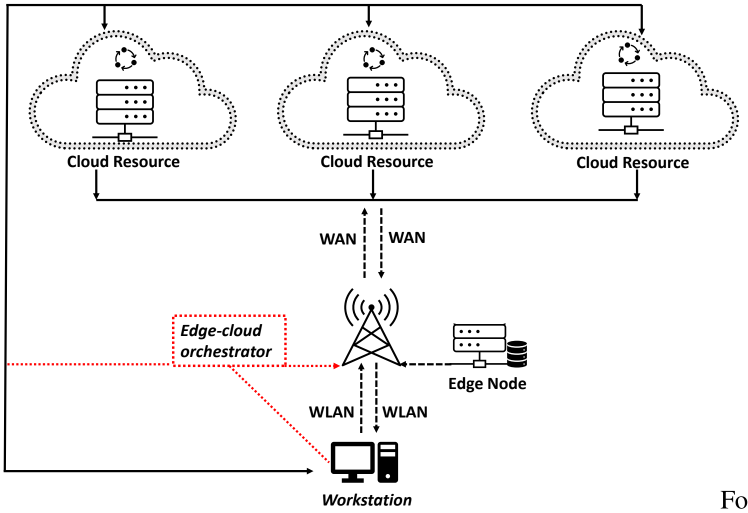

Figure 1: Edge-cloud orchestrator embedded in an edge node directing data traffic to and from edge and/or cloud servers.

- With high, intermediate and low computational task complexities, optimum partitioning by an edge-located orchestrator results in minimal total task completion times at different ranges of Wide Area Network (WAN) speeds for edge-to-cloud data transfer.
- With increasing WAN speeds, the reductions in total task completion times are larger and occur with a greater proportion of the task sent for processing in cloud resources.

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 WAN speeds and different task computational complexities. Section IV draws conclusions and outlines possible strategies for further optimising edge-cloud synergies for Big Data analytics.

II. Problem Formulation

An orchestrator embedded in the proximal edge node decides the transfer of data files from edge servers to cloud servers for time-limited processing when an advantage for processing exists by edge-to-cloud data transfer [13].

Following the mathematical treatment proposed by [14], a task processing time can be represented by

$$T^T = T^{ES}(1-\theta) + T^C_\theta \tag{1}$$

where T^{ES} is the total task processing time in the edge server, T^C is total task processing time using cloud resources and θ (max = 1) is the proportion of the data forwarded to the cloud from the edge node.

Table I: Parameters used for numerical simulations

Parameter	Numerical Value/Range	Unit
α_e	1.36×10^{11}	IPS
β_c	2.72×10^{12}	IPS
λ	0.0000529 - 0.00227	bpi
WLAN	50	Mbps
WAN	0.5-100	Mbps

Each of T^{ES} and T^C is composed of multiple sub-times. For T^C there are (sequentially) a data transfer time from the end user via a Wireless Local Area Network (WLAN) to the edge node, a processing time in the edge node and a reduced (× 0.1) data transfer back to the end user. For Tc there are additionally (and sequentially) a data transfer time from the edge node via a Wide Area Network (WAN), a processing time in the cloud and reduced (× 0.1) data transfer back to the edge node.

The processing times are directly proportional to the data size (in GB) and inversely proportional to the server processing speed and to the computational complexity (in bits per instruction) of the task [24]. The data transfer times assume a constant WLAN speed but variable WAN speeds and the total time when utilising cloud resources is critically dependant on the WAN speed: high WAN speeds favour edge-to-cloud transfer while low WAN speeds favour edge node processing [14].

Based on knowledge accessible by the edge node orchestrator, a value of θ (in the range 0-1) is selected to minimise T^T at the WAN speed then applicable and which is assumed to be constant for the time represented by edge-to-cloud data transfer.

III. Numerical Simulations

Numerical simulations were performed to identify possible optimal minimum total task processing times by migrating proportions of tasks from edge nodes to cloud resources. The numerical values of parameters used in these simulations are listed in Table I. A cloud:edge processing speed. upscaling of 20:1 was used [24].

Where α_e is the computing capability of edge server in instructions/sec, β_c is the computing capability of Cloud in instructions/sec and λ is the application complexity on the ES in bits/instructions.

A. Task computational complexity parameter choices

Task computational complexity values were taken for scientific apps which represent scientific programs of varying complexity suitable for modelling Big Data analytics [24].

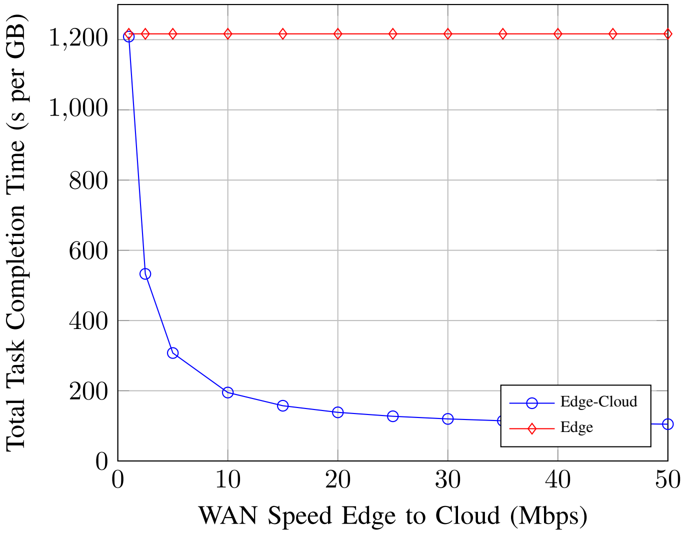

Figure 2: Effect of WAN speed on total task completion time for the highest task computational complexity (0.000059 bpi).

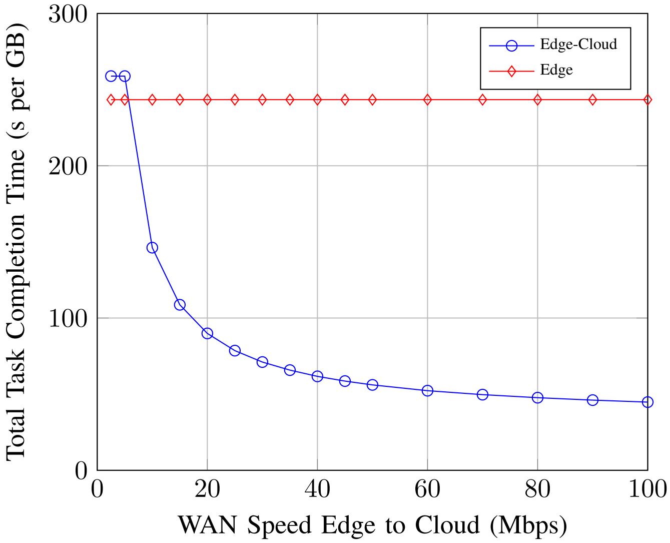

Figure 3: Effect of WAN speed on total task completion time for an intermediate task computational complexity (0.000286 bpi).

At the highest computational complexity (0.000059 bpi, Figure 2), edge-to-cloud data transfer has a clear total task processing time advantage at a WAN speed of 2.5 Mbps and such transfer results in time reductions of >90% at WAN speeds greater than 40 Mbps.

At an intermediate computational complexity (0.000286 bpi, Figure 3), a WAN speed in excess of 5Mbps is required for any time advantage of edge-to-cloud transfer to be evident. Above WAN speeds of 70 Mbps, such transfer results in time reductions of >80%.

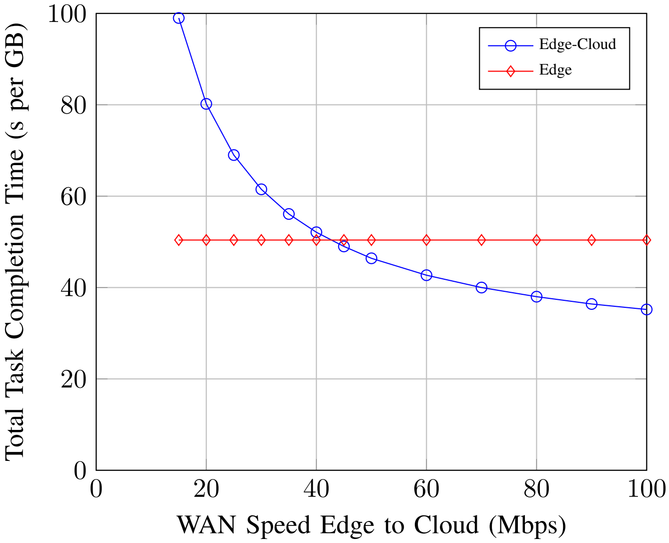

Figure 4: Effect of WAN speed on total task completion time for the lowest task computational complexity (0.00277 bpi)

In contrast, at the lowest computational complexity (0.00277 bpi, Figure 4) a minimum WAN speed of 40 Mbps is necessary before time advantages can be achieved and, even at 100 Mbps, the time reduction over edge node processing is only 30%.

B. *Task partitioning between edge and cloud resources*

At low WAN speeds (0.5-5 Mbps), the highest complexity tasks showed optima for task partitioning to result in reduced total task completion times (Figure 5). At the slowest WAN speed (0.5 Mbps), increasing the partitioning of the task to cloud resources to >0.4 resulted in increased total task completion times relative to edge-only processing.

At higher WAN speeds ($\geq$ 10 Mbps), task time reductions increased progressively as the WAN speed was increased.

At the intermediate task computational complexity, a similar pattern of optimisation occurred but at a higher range of WAN speeds, 5-25 Mbps (Figure 6). Again, at higher WAN speeds ($\geq$ 30 Mbps), task time reductions increased progressively as the WAN speed was increased with partitioning optima.

At the lowest task computational complexity, an optimised total task completion time represented a 37% time saving over edge-only processing and this used a partitioning of 0.7 using cloud resources but required a WAN speed of 100 Mbps (Figure 7). Total partitioning of the task to cloud resources at 100 Mbps achieved only a 30% time saving over edge-only processing.

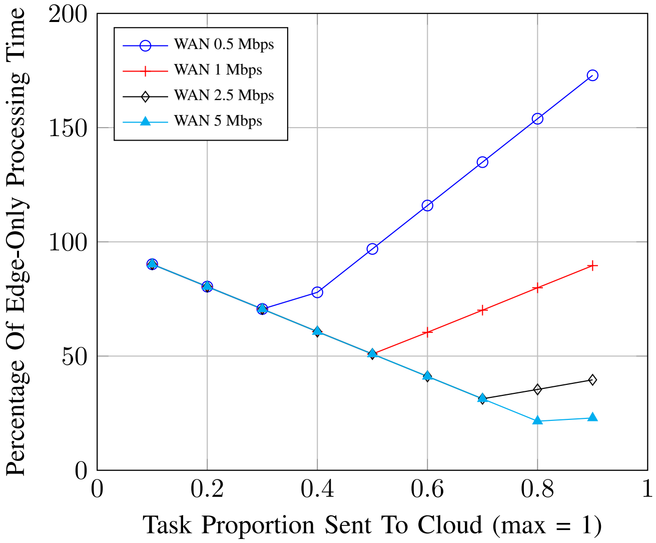

Figure 5: Effect of increasing the partitioning of a high-complexity (0.000059 bpi) task from edge nodes to cloud resources at relatively low WAN speeds.

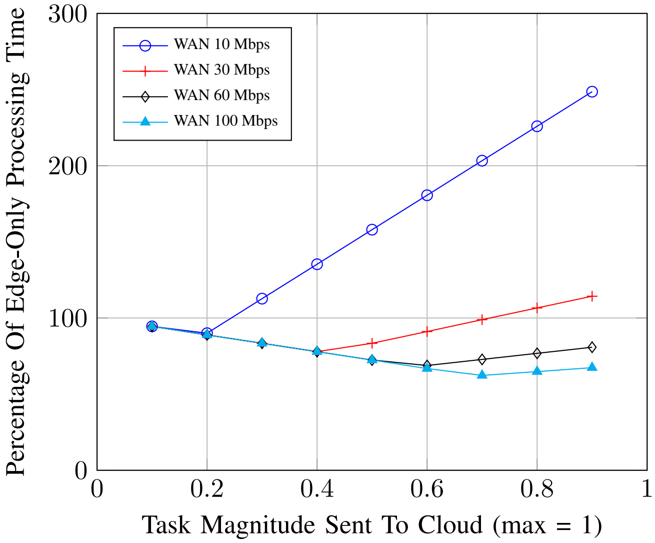

Figure 7: Effect of increasing the partitioning of the lowest-complexity (0.00277 bpi) task from edge nodes to cloud resources.

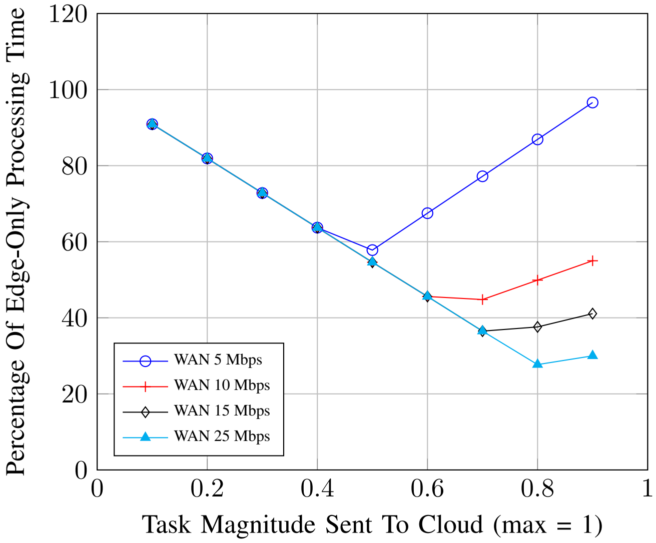

Figure 6: Effect of increasing the partitioning of an intermediate-complexity (0.000286 bpi) task from edge nodes to cloud resources.

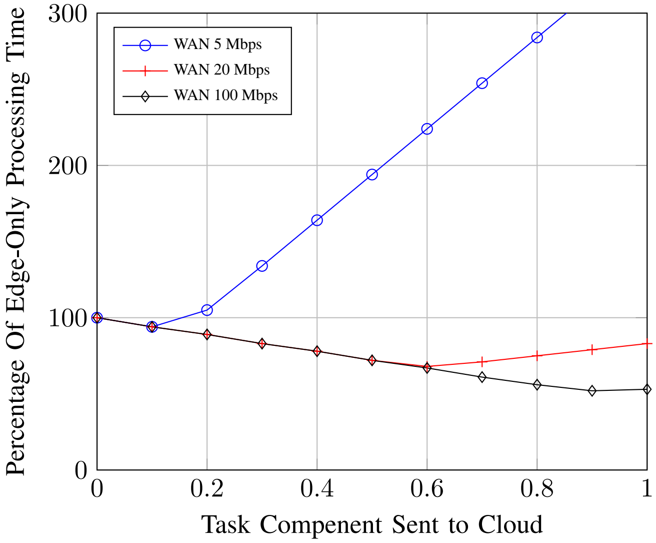

Figure 8: Parallel processing of the lowest-complexity (0.00277 bpi) task between one edge node and three equivalent cloud servers.

C. Edge-to-multiple clouds for task partitioning

When multiple cloud resource sites are available to the edge-cloud orchestrator, increasingly large reductions in total task completion times can be achieved. For example, with three equivalent cloud sites and WAN speeds $\geq$ 80 Mbps 50% reductions in total task completion times are approached (Figure 8).

In this scenario, makespan analysis is relevant [25]. In all cases where a task is partitioned between one edge node and three cloud servers, the makespan (i.e., the last task to be finished) is that of processing in the edge node. Increasing degrees of task partitioning to greater numbers of servers progressively reduces total task completion times if only edge-to-cloud routes are used. A more effective solution is, however, for the orchestrator to route data transfer and return directly to and from a cloud resource server, this bypassing the edge node; at a WAN speed of 100 Mbps, a 75% saving on total task completion time for the lowest-complexity tasks can be achieved but a minimum client-to-cloud WAN speed of 25

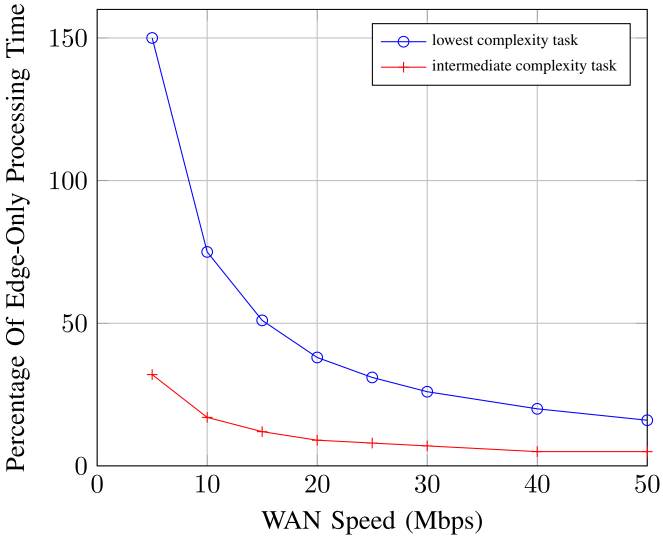

Figure 9: Effect of WAN speed on time savings by orchestrating data traffic direct to cloud resources

Mbps is required for any advantage over edge-only processing to be possible.

Even at low WAN speeds (10 Mbps), partitioning the lowest-complexity task into three equal portions and transmitting the data directly to three equivalent cloud resources under the direction of the edge-cloud orchestrator reduces the total task completion time by 25%, which increases to a 84% saving with a WAN speed of 50 Mbps, i.e., equal to the WLAN speed (Figure 9). Much greater reductions can be obtained with higher complexity tasks using the same range of WAN speeds (Figure 9).

IV. Conclusions and Future Work

Edge-cloud synergy is a powerful means of reducing the time required for complex manipulations in Big Data analytics. At low WAN speeds for edge-to-cloud data transfer, partitioning the high-complexity tasks between the two sets of resources can find optimum solutions; this scenario could, for example, occur if high latency or job queuing at cloud servers compromises otherwise acceptable WAN speeds. As the task complexity decreases, higher WAN speeds are required for optimum time reductions relative to edge-only processing to occur.

The challenge of achieving computational efficiency for low-complexity tasks is partly soluble by using multiple cloud sites or cloud resources and harnessing greater degrees of parallel processing or direct client-to-cloud data transfer at fast WAN speeds. Nevertheless, the time reductions possible with low-complexity tasks do not rival those achieved with high-complexity tasks but further improvements would be possible with higher cloud-to-edge processing ratios.

Future work in this area could address how edge nodes operating in Symmetric Multiprocessing [26], Massively Parallel Processing [27], Clustered Memory Scheduling [28] or Non-Uniform Memory Access [29], [30] systems could be synergised with cloud resources for increased efficiency in Big Data analytics.

V. Acknowledgements

This work was partially funded by European Commission's Horizon 2020 project titled "Digital twins bringing agility and innovation to manufacturing SMEs, by empowering a network of DIHs with an integrated digital platform that enables Manufacturing as a Service (MaaS)" (DIGITbrain) under grant agreement no. 952071.

References

[1] X. Hu, L. Wang, K.-K. Wong, M. Tao, Y. Zhang, and Z. Zheng, "The synergy of edge and central cloud computing with wireless mimo backhaul," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[2] L. Gao, "Research on intelligent logistics architecture based on edge cloud synergy," in *2021 2nd International Conference on Urban Engineering and Management Science (ICUEMS)*. IEEE, 2021, pp. 20–24.

[3] S. Taherizadeh, V. Stankovski, and M. Grobelnik, "A capillary computing architecture for dynamic internet of things: Orchestration of microservices from edge devices to fog and cloud providers," *Sensors*, vol. 18, no. 9, p. 2938, 2018.

[4] D. Pizzolli, G. Cossu, D. Santoro, L. Capra, C. Dupont, D. Charalampos, F. De Pellegrini, F. Antonelli, and S. Cretti, "Cloud4IoT: A heterogeneous, distributed and autonomic cloud platform for the IoT," in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2016, pp. 476–479.

[5] M. Villari, A. Celesti, G. Tricomi, A. Galletta, and M. Fazio, "Deployment orchestration of microservices with geographical constraints for edge computing," in *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2017, pp. 633–638.

[6] E. Yigitoglu, L. Liu, M. Looper, and C. Pu, "Distributed orchestration in large-scale IoT systems," in *2017 IEEE International Congress on Internet of Things (ICIOT)*. IEEE, 2017, pp. 58–65.

[7] A. Zanni, S. Forsstrom, U. Jennehag, and P. Bellavista, "Elastic provisioning of internet of things services using fog computing: An experience report," in *2018 6th IEEE international conference on mobile cloud computing, services, and engineering (MobileCloud)*. IEEE, 2018, pp. 17–22.

[8] N. Wang, B. Varghese, M. Matthaiou, and D. S. Nikolopoulos, "Enorm: A framework for edge node resource management," *IEEE transactions on services computing*, vol. 13, no. 6, pp. 1086–1099, 2017.

[9] A. Lertsinsrubtavee, A. Ali, C. Molina-Jimenez, A. Sathiaseelan, and J. Crowcroft, "Picasso: A lightweight edge computing platform," in *2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*, 2017, pp. 1–7.

[10] E. Yigitoglu, M. Mohamed, L. Liu, and H. Ludwig, "Foggy: a framework for continuous automated iot application deployment in fog computing," in *2017 IEEE international conference on AI & Mobile Services (AIMS)*. IEEE, 2017, pp. 38–45.

[11] G. Davoli, D. Borsatti, D. Tarchi, and W. Cerroni, "Forch: An orchestrator for fog computing service deployment," in *2020 IFIP Networking Conference (Networking)*. IEEE, 2020, pp. 677–678.

[12] M. Alam, J. Rufino, J. Ferreira, S. H. Ahmed, N. Shah, and Y. Chen, "Orchestration of microservices for iot using docker and edge computing," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 118–123, 2018.

[13] 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.

[14] R. Singh, J. Kovacs, and T. Kiss, "To offload or not? an analysis of big data offloading strategies from edge to cloud," *IEE AI IOT 2022*, 2022.

[15] M. Gao, R. Shen, L. Shi, W. Qi, J. Li, and Y. Li, "Task partitioning and offloading in dnn-task enabled mobile edge computing networks," *IEEE Transactions on Mobile Computing*, 2021.

[16] L. Ale, S. A. King, N. Zhang, A. R. Sattar, and J. Skandaraniyam, "D3pg: Dirichlet ddpg for task partitioning and offloading with constrained hybrid action space in mobile edge computing," *IEEE Internet of Things Journal*, 2022.

[17] J. Liu and Q. Zhang, "Adaptive task partitioning at local device or remote edge server for offloading in mec," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.

[18] Y. Ding, C. Liu, X. Zhou, Z. Liu, and Z. Tang, "A code-oriented partitioning computation offloading strategy for multiple users and multiple mobile edge computing servers," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4800–4810, 2019.

[19] H. Tianfield, "Towards edge-cloud computing," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 4883–4885.

[20] C. Esposito, A. Castiglione, F. Pop, and K.-K. R. Choo, "Challenges of connecting edge and cloud computing: A security and forensic perspective," *IEEE Cloud computing*, vol. 4, no. 2, pp. 13–17, 2017.

[21] D. Haja, B. Vass, and L. Toka, "Improving big data application performance in edge-cloud systems," in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. IEEE, 2019, pp. 187–189.

[22] C. Yang, S. Lan, L. Wang, W. Shen, and G. G. Huang, "Big data driven edge-cloud collaboration architecture for cloud manufacturing: a software defined perspective," *IEEE access*, vol. 8, pp. 45 938–45 950, 2020.

[23] M. Babar, M. A. Jan, X. He, M. U. Tariq, S. Mastorakis, and R. Alturki, "An optimized iot-enabled big data analytics architecture for edge-cloud computing," *IEEE Internet of Things Journal*, 2022.

[24] 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.

[25] V. Pandey and P. Saini, "Application layer scheduling in cloud: Fundamentals, review and research directions," *Comput. Syst. Sci. Eng*, vol. 34, no. 6, pp. 357–376, 2019.

[26] G. A. Malazgirt, B. Kiyan, D. Candas, K. Erdayandi, and A. Yurdakul, "Exploring embedded symmetric multiprocessing with various on-chip architectures," in *2015 IEEE 13th International Conference on Embedded and Ubiquitous Computing*. IEEE, 2015, pp. 1–8.

[27] K. Semba, H. Katagiri, T. Asanuma, M. Miwa, H. Sano, and T. Yamada, "Realistic and very fast simulation of electric machines and apparatus by using massively parallel processing," in *2017 20th International Conference on Electrical Machines and Systems (ICEMS)*. IEEE, 2017, pp. 1–5.

[28] Y. Kim, M. Papamichael, O. Mutlu, and M. Harchol-Balter, "Thread cluster memory scheduling: Exploiting differences in memory access behavior," in *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 2010, pp. 65–76.

[29] X. Guo and H. Han, "A good data allocation strategy on non-uniform memory access architecture," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. IEEE, 2017, pp. 527–530.

[30] W. Liu, H. Liu, X. Liao, H. Jin, and Y. Zhang, "Hngraph: Parallel graph processing in hybrid memory based numa systems," in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2021, pp. 388–397.