International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 7, Issue 2, pp 36-41, February-2023
https://doi.org/10.47001/IRJIET/2023.702005

A Review: Scheduling Methods in Serverless Edge Computing

[1]*Aseel Thamer Ebrahem, [2]Basil Sh. Mahmmood

[1]Department of Computer Engineering, Northern Technical University, Mosul, Iraq
[2]Computer Engineering Department, College of Engineering, University of Mosul, Mosul 41002, Iraq
*Corresponding Author's E-mail: aseelthamer@ntu.edu.iq

***Abstract* - Recently, the emergence for serverless edge computing has been an attractive topic of research community. It is a new understood in which brings computational resources proximity to edge node network which permits the computation of tasks which triggers the program carry out as a reply for assigned events. The purpose of this research is to investigate diverse methods relevant to scheduling tasks in serverless edge computing. Consequently, these methods are classified depending on their objective functions, precisely, reducing execution cost, low latency, reducing of makespan, scalability, offloading, and intelligent scheduling. The trade-off and dependency among the goals play a critical role in make decision to determine the scheduling policy efficiently. These trade-off and dependency may be led to waste of resources. This issue can be resolved by integrating among the above objectives. Serverless edge computing is predicted to have a substantial influence on the process of the scheduling. Using serverless edge computing, we can run the fine-grained requests on serverless platform by scheduler within coordination item. This allows in expanding size of the resource's utilization due to the optimal use of the scheduling process. An extension of the constrained time limit of objectives by cloud providers will permit the execution of the entire tasks, and get ridding of the rescheduling issue using a serverless edge computing mechanism.**

Keywords: Edge Node Computing; Scheduling Methods; Serverless Edge Computing; Cloud Computing; dependency.

I. INTRODUCTION

Serverless edge computing is an advanced direction for the internet of things (IoT) and users, that aims to place serverless platform in the edge computing just like in the cloud computing. In short, from an edge perspective, the devices for internet of things are in connection with fog computing nodes, which can run their functions based on serverless platform distributed across the network from the cloud to the edge. From serverless perspective, it is providing a powerful distributed event-processing platform [1]. From the internet of things perspective, less adjustment is required and it merely has to apply microservices policies. From cloud perspective, the native Serverless within the cloud platform is aiming to combine itself with its distributed accomplice in fog computing. [1, 2, 3, 4, 5, 6, 7, 8, 9].

Clients can allocate their cloud utilization demands of the resources such as operating system, servers, storage, and IDE. Besides, the clients can utilize from resources of the cloud based on service level agreement that involves the required service quality restricts [10]. With diverse resources of cloud platforms, companies and individuals are capable to deploy code functions, process and store data, and mange code functions using virtual machines and containers etc. [11,12].

In cloud, serverless enables to merge between technological evolves on business logic like containerization, event-driven programming, microservices, containerization, Function as a Service [13], and the concept of pure pay-as-you-go paradigm along with effortless scalability [14]. Meanwhile, the significant developments for internet of things applications, clients were needed to dedicate extra effort to meet their demands such as real-time execution, low latency, event-driven programming, and powerful deployments [15]. Cloud platform with efficient processes could be good option to achieve IoT demands but suffering from issues like access speed owing to being far away from users [16]. In cloud struggle and Internet of Things demands, the edge computing proposed as a complementary choice [17]. In fog computing, the process is closer to edge nodes and real-time execution will be closer to complete. The available resources of the fog computing as known are limited, but edge nodes will leverage from the distributed style of tasks to offer a pool of available resources.

Serverless edge computing is the paradigm in which the user's functions are deployed on the edge-cloud computing. The edge-cloud manages everything including the functions and makes the resources available for the users. Serverless edge computing consists of three main tiers are computation, execution, and coordination as illustrated in figure 1[36]. The computation layer: includes three components are computing,

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 2, pp 36-41, February-2023

https://doi.org/10.47001/IRJIET/2023.702005

storage, monitor. The computing is responsible for creating a new object and requested resources. The storage component provides function's state, the needed dependencies for running a task in the applications.

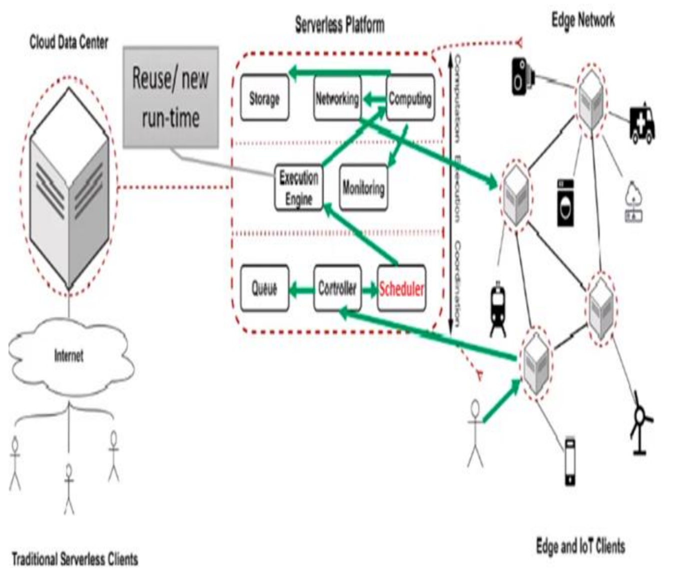

Figure 1: A simplified Serverless Edge Computing model [36]

The first layer is the coordination layer, the edge node will be dispatching the task and its related data to the controller component which works as a load balancer. In case the request of client is valid, and the received message is lightweight, the request of client is dispatched to the scheduler item, which based on the triggers and rules passes the request to matching worker relevant to a function [1]. Contrary, the request of client is kept in the queue item for certain causes like failure handling and in another repetition, it is dispatched to the scheduler. The execution engine item will call a function for having runtime environment of the client request if the function already runs it can reuse their function instance. Otherwise, it will ask the computing item to provision resources again then computing would look at the monitoring item. In case there is enough resources to provision for this function, asks the storage item to read the data of the function and provisioning the resource. Otherwise, it will ask the networking component to upload the function to another edge node, may be to the cloud depending on the implementation.

In this research, we investigate diverse methods relevant to scheduling tasks in serverless edge computing. Furthermore, we illustrate how the architecture of this scheduling issue can be offered by applying serverless platform. The reminder of the research is structured as follows. In section 2, we investigate diverse methods that studied the scheduling issues according to their goals. Section 3 gives a discussion of the issues and how serverless platform can contribute in minimizing the utilization of the resources. Finally, section 4 concludes the major outcomes and suggests future works.

II. SCHEDULING METHODS

Issues of scheduling tasks have been addressed intensely in the previous works (i.e. state-of-art). Typically, the shed light on the reducing execution time, the reducing execution cost, scalability, offloading, and intelligent scheduling. In this part, we address methods that studied these challenges while intended to emphasize the utilities of applying serverless edge computing in the issue of scheduling. This part classifies the scheduling methods based on their objective functions into: (1) execution cost, latency, and execution time (2) scalability, offloading, and revenue (3) Intelligent Scheduling.

2.1 Scheduling of Execution Time, Latency, and Execution Cost

Many methods have studied the issues of execution time, latency, and execution cost under the presence of budget restrictions [18, 19, 20, 21, 22, 23] Typically, the issues of execution time, latency, and execution cost are dominated by the vital role in tasks scheduling.

Rausch and his colleagues [18] proposed a container scheduling system (Skippy) that allows of serveless platforms to efficiently leverage of edge infrastructures. Their scheduler makes comparative between data and computation movement, and takes into account workload-specific compute demands like graphics processing unit acceleration. Moreover, they proposed a technique to automatically fine-tune the parameters of scheduling restrictions to improve high-level operational goals like latency, cloud execution cost, or mitigating task execution. Their findings appear enhance the quality of task placement and help significantly in meeting operational goals. Hassan et al. in [19] mentioned in their survey some of scheduling methods that categorized according to reduce the cost, makespan, and both in serverless computing.

Das et al. in [20] developed method to manipulate a queue of events, and for each event, it sends the job to a function in the suitable container, from among the edge container and a group of containers in the cloud with diverse resource customizations. Their method manipulates two improvement issues the minimize latency and minimize cost, where the access speed is computed in terms of the time from the ingestion of the event to the storage of the outcomes in the cloud. Andreades et al. in [21] proposed a high-speed control plane design adopt on a central switch scheduler to reduce the node-to-node latency in the fog computing.

ISSN (online): 2581-3048
Volume 7, Issue 2, pp 36-41, February-2023
https://doi.org/10.47001/IRJIET/2023.702005

Aslanpour et al. in [22] implemented paradigm an energy-aware resource scheduling issue in serverless platform, given a network of renewable-energy and battery-operated powered nodes. They invented priority-based and zone-oriented techniques to enhance the operational availability of low powered nodes. This issue running on a serverless platform is formally formulated and priority-based manner with characteristics like "warm scheduling" and "sticky offloading". Their suggested techniques seek to surge the operational availability of edge nodes while reducing the latency. Wang et al. in [23] suggested a job scheduling framework, which can handle various nodes based on priorities aiming to minimize the access speed.

The key problem in implementing time, cost, and latency minimization methods is the budget restrictions. Budget restrictions limit the size of resources that can be allocated, and this obstructs the process of improvement. The scheduling approach reflects the trade-off and dependency among the objectives which have a main impact on the efficiency.

2.2 Scheduling of Scalability, Offloading, and Revenue

The issues of scheduling of scalability, offloading, and revenue have been addressed in the literature [24, 25, 26, 27, 28, 29, 30, 31]. Bermbach et al. in [24] suggested an auction method in which developers bid on resources whereas edge nodes determine locally which functions to carry out and which to migrate in order to maximize income (AuctionWhisk). Their method can act in a decentralized manner and online. This manner, over-burden edge nodes can make local decisions about task accomplish. Therefore, also the task placement is distributed across nodes and no longer the scalability problemit is with centralized scheduling policies.

Ahvar et al. [25] achieved a generic energy paradigm to precisely assess and trade-off the power wasted by these new cloud-related architectures. They provided a scheduling method determining the set of virtual machines on each cloud network and the bandwidth with another virtual machines.

Aumala et al. [26] proposed a package-aware scheduling (PASch), a new scheduling method to allow nodes can reuse execution platforms with preloaded packages. Furthermore, they proposed a scheduling method that goals package affinity during scheduling, while actively avoiding execution engine overload.

Li et al. in [27] presented framework adopt on combined time scheduling, resource customization, and function executor determent for collaborative function offloading, and suggested a real-time auction method to surge the social welfare.

Alameddine et al. in [28] addressed the dynamic task offloading and scheduling issue (DTOS) in multi-access mist computing, where tasks are sharing in execution. Additionally, Josilo et al. in [29] suggested a processing offloading scheduling framework to decide whether to migrate the tasks of end-users to edge networks, in order to reduce the cost that is an integration of energy consumption and delay.

Luo et al. in [30] presented a collaborative task data scheduling framework in vehicle edge computing, where the events of nodes can be not merely executed locally (i.e., migrated to Roadside Units), but also can be offloaded to other nodes with inactive resources. Cao et al. In [31], the authors proposed a programming style to determine the best task placement computing from the costumers' view to prevent task offloaded, hence reducing the time cost. From the platform's perspective, the powerful guiding method is shown to schedule tasks to reduce the overall cost.

Typically, the scalability, offloading, and revenue include the service cost. The price that customers pay for the use of resources regardless of whether they are consumed or not is the aggregate rental time. It is predicted that the devises will be inactive for a substantial amount of time owing to the data dependency restrictions among the objectives. By applying serverless edge computing, we can substantially optimize the usage of resources.

2.3 Intelligent Scheduling Methods

Many methods have investigated the issue of optimizing the tasks scheduling using artificial intelligence [32, 33, 34, 35]. Ning and his colleagues in [32] adopted the Deep Q-network method for modeling an intelligent scheduling method for Vehicle Edge Computing. One of type machine learning that is called the federated learning model, works on training the resource scheduling method on servers or edge nodes that do not communicate local data samples [33]. Further, to participate customize storage, computing, and communication resources in fog computing. Wang and his colleagues [34] presented hybrid technique by combination between federated learning technique and deep reinforcement learning technique in fog computing and suggested a new scheme, in which the weights of the learning paradigm are exchanged between end-users and edge computing to improve the scheduling paradigm. In addition to, Qian and his colleagues [35] presented hybrid method from federate learning model and a centralized greedy technique for tackling the issue of functions placement with kept on privacy in the fog computing.

The complexities of these issues are due to its conflicted objectives. For example, incrementing the execution time normally results in minimizing the cost, whereas reducing the

ISSN (online): 2581-3048
Volume 7, Issue 2, pp 36-41, February-2023
https://doi.org/10.47001/IRJIET/2023.702005

execution time may result in incrementing the cost as trade-off between them. The flexible scalability, latency, offloading, pure pay-to-use model contribute to improve cloud services. The artificial intelligence can be developing scheduling methods in serverless edge computing. The table1 outlines the diverse scheduling methods in the previous works.

Table 1: outline of scheduling methods

Reference	Year	Time	Cost	Latency	Scalability	Offloading	Budget	AI
Rausch et al. [18]	2021	√	√	√		√	√	
Hassan et al. [19]	2021	√	√					
Das et al. [20]	2020		√	√				
Andreades et al. [21]	2019			√				
Aslanpour et al. [22]	---				√	√	√	
Wang et al. [23]	2017			√				
Bermbach et al. [24]	2022				√	√	√	
Ahvar et al. [25]	2019			√	√			
Aumala et al. [26]	2019		√			√		
Li et al. [27]	2019				√	√		
Alameddine et al. [28]	2019		√	√		√		
Josilo et al. [29]	2020		√			√		
Luo et al. [30]	2020		√			√		
Cao et al. [31]	2018		√			√		
Ning et al. [32]	2019							√
Konečný et al. [33]	2015							√
Wang et al. [34]	2019							√
Qian et al. [35]	2019			√				√

III. DISCUSSION

As mentioned in the above section, the trade-off and the dependency among the objectives play critical role in determining the powerful of the scheduling policy. These trade-off and dependency may be lead to reduce the utilization of resources. this challenge can be mitigated by depending on a hybrid strategy that integrates among the goals. Serverless edge computing has found position at the edge computing to leverage edge node network to deploy segments of tasks as real-world business rules, specifically events that can be triggered within millisecond.

Serverless edge computing is predicted to have extremely influence on the process of scheduling tasks. The task scheduling will include various VMs and owing the precedence restrictions, many of these VMs may stay in state of inactive while the scheduling. The serverless edge computing contributes to execute the fine-grained tasks from end-edge to cloud and highlights only on scheduling the coarse tasks, this allows in surging the advantage of the resources.

Integrating among the objectives lead to reduce the expected time, cost, and latency for scheduling. The flexible scalability, latency, offloading, pure pay-to-use model contribute to improve cloud services. The artificial intelligence can be developing scheduling methods in serverless edge computing.

IV. CONCLUSION AND FUTURE WORKS

In this research, we studied methods that addressed the issue of scheduling on serverless edge computing. we discussed some variations of this issue namely time executing, latency, executing cost, offloading, revenue, and intelligent scheduling. We studied these issues and showed that by using serverless edge computing we can simplify this issue, thus attain better schedules in terms time, cost, latency, scalability, budget, and AI. Because the constrains on the executing time cost latency in serverless edge computing, an extension for the executing time, cost, and latency limit by cloud providers could permit the option of totally executing the complete scheduling by applying serverless edge computing structure and therefore, avoid repeat process of the scheduling. In future works, for scheduling methods in serverless edge computing to be studied deeply, we highlight on the state-of-art research problems and look forward to more research investment in artificial intelligence specially using convolutional neural network or wavelet neural network.

REFERENCES

[1] Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, RodricRabbah, and Aleksander Slominski. 2017. Serverless computing: Current trends and open problems. In Research Advances in Cloud Computing. Springer, 1–20.

[2] Paarijaat Aditya, Istemi Ekin Akkus, Andre Beck, Ruichuan Chen, Volker Hilt, Ivica Rimac, Klaus Satzke, and Manuel Stein. 2019. Will Serverless Computing Revolutionize NFV? Proc. IEEE 107, 4 (2019), 667–678.

[3] Javadi Bahman, Sun Jingtao, and Ranjan Rajiv. 2020. Serverless architecture for edge computing. In Edge Computing: Models, technologies and applications. Institution of Engineering and Technology, 249–264. https://doi.org/10.1049/ pbpc033e_ch12.

[4] Claudio Cicconetti, Marco Conti, Andrea Passarella, and Dario Sabella. 2020. Toward Distributed

ISSN (online): 2581-3048
Volume 7, Issue 2, pp 36-41, February-2023
https://doi.org/10.47001/IRJIET/2023.702005

Computing Environments with Serverless Solutions in Edge Systems. IEEE Communications Magazine 58, 3 (2020), 40–46.

[5] Eric Jonas, Johann Schleier-Smith, VikramSreekanti, Chia-Che Tsai, Anurag Khandelwal, Qifan Pu, Vaishaal Shankar, Joao Carreira, Karl Krauth, and NeerajaYadwadkar. 2019. Cloud programming simplified: A berkeley view on serverless computing. arXiv preprint arXiv:1902.03383 (2019).

[6] Young Ki Kim, M Reza HoseinyFarahabady, Young Choon Lee, and Albert Y Zomaya. 2020. Automated Fine-Grained CPU Cap Control in Serverless Computing Platform. IEEE Transactions on Parallel and Distributed Systems 31, 10 (2020), 2289–2301.

[7] Dayeol Lee, David Kohlbrenner, Shweta Shinde, Krste Asanović, and Dawn Song. 2020. Keystone: An open framework for architecting trusted execution environments. In Proceedings of the Fifteenth European Conference on Computer Systems. 1–16.

[8] Jussi Nupponen and Davide Taibi. 2020. Serverless: What it is, what to do and what not to do. In 2020 IEEE International Conference on Software Architecture Companion (ICSA-C). IEEE, 49–50.

[9] Davide Taibi, J. Spillner, and K. Wawruch. 2021. Serverless Where are we now and where are we heading? IEEE Software 38, 1 (2021).

[10] Abrishami, S., Naghibzadeh, M. and Epema, D.H., 2013. Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. Future generation computer systems, 29(1), pp.158-169.

[11] Arya, L.K. and Verma, A., 2014. Workflow scheduling algorithms in cloud environment-A survey. 2014 Recent Advances in Engineering and Computational Sciences (RAECS), pp.1-4.

[12] Bardsiri, A.K. and Hashemi, S.M., 2012. A review of workflow scheduling in cloud computing environment. International Journal of Computer Science and Management Research, 1(3), pp.348-351.

[13] Erwin Van Eyk, Lucian Toader, Sacheendra Talluri, Laurens Versluis, Alexandru Ut,ă, and Alexandru Iosup. 2018. Serverless is more: From paas to present cloud computing. IEEE Internet Computing 22, 5 (2018), 8–17.

[14] JussiNupponen and DavideTaibi. 2020. Serverless: What it is, what to do and what not to do. In 2020 IEEE International Conference on Software Architecture Companion (ICSA-C). IEEE, 49–50.

[15] Mohammad S. Aslanpour, Sukhpal Singh Gill, and Adel N. Toosi. 2020. Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research. Internet of Things 12 (2020), 100273. https://doi.org/10.1016/j.iot.2020.100273.

[16] RajkumarBuyya, Satish Narayana Srirama, Giuliano Casale, Rodrigo Calheiros, Yogesh Simmhan, Blesson Varghese, ErolGelenbe, Bahman Javadi, Luis Miguel Vaquero, and Marco A S Netto. 2018. A manifesto for future generation cloud computing: Research directions for the next decade. ACM computing surveys (CSUR) 51, 5 (2018), 1–38.

[17] Mostafa Ghobaei-Arani, AlirezaSouri, and Ali A Rahmanian. 2020. Resource Management Approaches in Fog Computing: a Comprehensive Review. Journal of Grid Computing 18, 1 (2020), 1–42. https://doi.org/10.1007/s10723-019-09491-1

[18] Rausch, T., Rashed, A. and Dustdar, S., 2021. Optimized container scheduling for data-intensive serverless edge computing. Future Generation Computer Systems, 114, pp.259-271.

[19] Hassan, H.B., Barakat, S.A. and Sarhan, Q.I., 2021. Survey on serverless computing. Journal of Cloud Computing, 10(1), pp.1-29.

[20] Das, A., Imai, S., Patterson, S. and Wittie, M.P., 2020, May. Performance optimization for edge-cloud serverless platforms via dynamic task placement. In 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID) (pp. 41-50). IEEE.

[21] Andreades, P., Clark, K., Watts, P.M. and Zervas, G., 2019. Experimental demonstration of an ultra-low latency control plane for optical packet switching in data center networks. Optical Switching and Networking, 32, pp.51-60.

[22] Aslanpour, M.S., Toosi, A.N., Cheema, M.A. and Gaire, R., Energy-Aware Resource Scheduling for Serverless Edge Computing.

[23] Wang, H., Gong, J., Zhuang, Y., Shen, H. and Lach, J., 2017, December. Healthedge: Task scheduling for edge computing with health emergency and human behavior consideration in smart homes. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 1213-1222). IEEE.

[24] Bermbach, D., Bader, J., Hasenburg, J., Pfandzelter, T. and Thamsen, L., 2022. AuctionWhisk: Using an auction-inspired approach for function placement in serverless fog platforms. Software: Practice and Experience, 52(5), pp.1143-1169.

[25] Ahvar, E., Orgerie, A.C. and Lebre, A., 2019. Estimating energy consumption of cloud, fog and edge computing infrastructures. IEEE Transactions on Sustainable Computing.

[26] Aumala, G., Boza, E., Ortiz-Avilés, L., Totoy, G. and Abad, C., 2019, May. Beyond load balancing: Package-

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 2, pp 36-41, February-2023

https://doi.org/10.47001/IRJIET/2023.702005

aware scheduling for serverless platforms. In 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID) (pp. 282-291). IEEE.

[27] Li, G. and Cai, J., 2019. An online incentive mechanism for collaborative task offloading in mobile edge computing. IEEE Transactions on Wireless Communications, 19(1), pp.624-636.

[28] Alameddine, H.A., Sharafeddine, S., Sebbah, S., Ayoubi, S. and Assi, C., 2019. Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing. IEEE Journal on Selected Areas in Communications, 37(3), pp.668-682.

[29] Jošilo, S. and Dán, G., 2020. Computation offloading scheduling for periodic tasks in mobile edge computing. IEEE/ACM Transactions on Networking, 28(2), pp.667-680.

[30] Luo, Q., Li, C., Luan, T.H. and Shi, W., 2020. Collaborative data scheduling for vehicular edge computing via deep reinforcement learning. IEEE Internet of Things Journal, 7(10), pp.9637-9650.

[31] Cao, Z., Zhang, H. and Liu, B., 2018, November. Performance and stability of application placement in mobile edge computing system. In 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC) (pp. 1-8). IEEE.

[32] Ning, Z., Dong, P., Wang, X., Rodrigues, J.J. and Xia, F., 2019. Deep reinforcement learning for vehicular edge computing: An intelligent offloading system. ACM Transactions on Intelligent Systems and Technology (TIST), 10(6), pp.1-24.

[33] Konečný, J., McMahan, B. and Ramage, D., 2015. Federated optimization: Distributed optimization beyond the datacenter. arXiv preprint arXiv:1511.03575.

[34] Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X. and Chen, M., 2019. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. IEEE Network, 33(5), pp.156-165.

[35] Qian, Y., Hu, L., Chen, J., Guan, X., Hassan, M.M. and Alelaiwi, A., 2019. Privacy-aware service placement for mobile edge computing via federated learning. Information Sciences, 505, pp.562-570.

[36] Aslanpour, M.S., Toosi, A.N., Cicconetti, C., Javadi, B., Sbarski, P., Taibi, D., Assuncao, M., Gill, S.S., Gaire, R. and Dustdar, S., 2021, February. Serverless edge computing: vision and challenges. In 2021 Australasian Computer Science Week Multiconference (pp. 1-10).

[37] T.S. Chen, H.-W. Tsai, Y.-H. Chang, and T.-C. Chen, “Geographic converge cast using mobile sink in wireless sensor networks,” Comput. Commun., vol. 36, no. 4, pp. 445–458, Feb. 2013.

Citation of this Article:

Aseel Thamer Ebrahem, Basil Sh. Mahmmood, “A Review: Scheduling Methods in Serverless Edge Computing” Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 2, pp 36-41, February 2023. Article DOI https://doi.org/10.47001/IRJIET/2023.702005

****