Scheduling job and Online Dispatching in Edge Cloud

Soniya R [1], Rashmi N Kuppast[2], Sinchana M N[3], Ila Aishwarya B P[4]

[1] *Assistant Professor, Artificial Intelligence and machine learning, RajaRajeswari college of Engineering, Bengaluru ,Karnataka, India*
[2] *Assistant Professor, Artificial Intelligence and machine learning, RajaRajeswari college of Engineering, Bengaluru, Karnataka, India*
[3] *Assistant Professor, Information science and engineering, RajaRajeswari college of Engineering, Bengaluru, Karnataka, India*
[4] *Assistant Professor, Department of Computer Application, Nagarjuna degree College,Bengaluru, Karnataka, India*

ABSTRACT

The development of 5G and the Internet of Things has made edge computing a more promising paradigm for low latency applications. Since the edge has limited resources, jobs requiring a lot of computing are typically offloaded to the cloud. The Edge cloud is a platform which promises to reduce the workloads that leads to too much of congestion and delay, so scheduling will play a significant role in offloading decisions in the Edge Cloud collaboration. As a result, a lot of work has already been done on scheduling, with the main goals being to minimise latency and improve performance and quality of experience. This study examines several scheduling algorithms in the context of edge cloud computing, including topics such as fault tolerance, QoS parameters, and benefits and drawbacks.I sincerely think that this survey would be beneficial in the creation of novel scheduling strategies since some of the problems, barriers and future way is been studied on the same. and hence forth sub dividing algorithms in every class based on the properties of algorithms.

Keyword : - *Internet of Things, Cloud Computing, Fog Computing, Edge Computing.*

1.INTRODUCTION

Smart devices are a hit because they are crucial to facial recognition, natural language processing, and augmented reality (AR) (NLP). A 2019 estimate from Cisco predicts that by the beginning of 2024, there will be about 30,000 million network devices worldwide. With recent developments in the telecommunications industry, such as 5G technology and AI (Artificial Intelligence (Deep Learning algorithms)), it is clear that there is also significant growth on the service provider's end.Numerous mobile devices are sufficiently powerful, boasting numerous cores, and some even feature excellent GPUs for computing workloads. Nevertheless, these mobile devices have restrictions because of their power, bandwidth, and space constraints. Because cloud computing has a lot of resources, it is therefore the most ideal to handle such obstacles and provide a smooth platform. However, this is ineffective for applications that require little latency, such as video analytics and real-time disaster preparedness. This issue can be addressed by implementing the edge computing paradigm, which involves bringing the resources closer to the mobiles However, because of its constrained resources, edge computing appears to be the problem. As such, it is unable to satisfy every request from the user or provide a high-quality user experience. Although it lacks the capacity to address every issue raised above, the edge cloud is a good way of addressing the advantages of cloud and overcoming the delays.computing to address some delay-sensitive, real-time applications. Since my tasks need to be scheduled at the edge or in the cloud, scheduling aids in improving the effectiveness of Edge Cloud collaboration. Therefore, we can conclude that the Edge Cloud needs a scheduling mechanism that

addresses difficult problems like inter-task dependency, heterogeneous resources, and variable user requirements and user mobility. The edge cloud architecture is shown in figure 1.

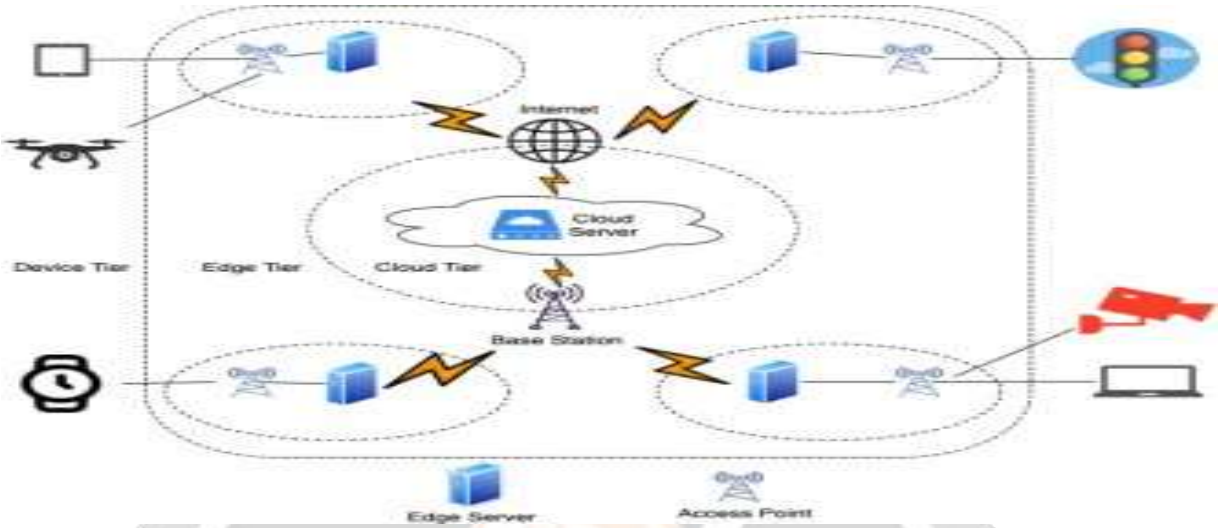

Figure. 1. The Edge Cloud Computing scenario.

The device tier, edge tier, and cloud tier are the three levels. Every connected device in the device layer completes its computation locally before being pushed to the cloud when the local resources are taking longer to complete the operation. We are aware that planning is the The difficult element of this process is determining when to schedule a specific job at the edge or cloud to maximise resource utilisation. This way the resources are allocated to every users. These days, a variety of scheduling algorithms are accessible, including meta-heuristic, machine learning-based, mobile-aware, dynamic, and static algorithms. The primary challenges are dependability, load balancing, availability, and dynamic resource allocation. Because of this, the Edge Cloud's technology and architecture are always in need of the best scheduling algorithm techniques.

2.LITERATURE SURVEY

As we move on to thee next generation of the technologies like IoT i.e., Internet of things are becoming more and more common. This leads to the need for the Edge Cloud to meet changing user needs, requirements, and mobility, particularly for applications that have delay-sensitive requirements. As a result, a variety of studies that look at various surveys on scheduling and resource allocation in the Edge Cloud have been provided. Since Edge Cloud scheduling is still in its early stages and requires a lot of work, new survey papers pertaining to Edge Cloud scheduling will be developed, which will be helpful for my survey as well.From the study on stochastic- based scheduling, the mechanism of scheduling was divided into three main types, Markov chain, process, and hidden Markov. From the survey this this could be the best approach where the algorithms are based on the features like cooperativeness, mobility of users, balancing load, time taken to respond and the consumption of energy. An additional piece concerning variable task scheduling approaches focused on two areas: heuristics and meta-heuristics, which examine the benefits and drawbacks of scheduling task algorithms, and edge computing architecture, which considers resource allocation, task offloading strategies, and user mobility.

3. MOTIVATION

In order to investigate a wide range of end devices, some previous research on Edge Cloud computing proposed several drone-based edge deployments. However, none of these studies concentrated on the ability to seamlessly integrate edge resources and service entities operating across multi-drone deployments in a single pool so that these resources can be managed and controlled holistically from a single plane, eliminating vendor lock-in situations. However, because of their limited resources, mobility, and availability issues, these are in fact difficult What is covered in the problem statement are the following points: • Since I only looked at survey studies from 2015 to

2021, they did not take into consideration the most recent articles published on Edge Cloud.• Few machine learning (ML)-based scheduling methods have been available in the Quality of Service (QoS), and Quality of Experience (QoE) of the Edge Cloud scenario. • The Edge Cloud's offloading mechanisms need broader classification and technological features. • To the best of my knowledge, no paper assesses so many criteria and includes a thorough part on the Edge Cloud's ML (machine learning) component.

4. PROPOSED WORK

Online dispatching and scheduling algorithm, which aids in the competitive analysis of the afore mentioned general model. In addition, I go over how to expand the technique to work in distributed systems and other more generic contexts. Survey has been divided into two border categories which are commonly known to be heuristic meta-heuristic algorithms so the taxonomy of 6 scheduling algorithms in the Edge Cloud is show in Figure This survey's ultimate objective is to offer a foundation for scheduling algorithms that aid in the development of unique scheduling algorithms.

Fig. 2. Scheduling algorithms in the Edge Cloud.

A. Online Algorithm

The policies, namely the policy of scheduling and policy of dispatching, drive the two components of this algorithm. It is also known that every mobile device uses the similar scheduling policy like dispatching to reduce jobs, and every server uses the same scheduling policy to decide which jobs to complete first and in what order.

• Scheduling Policy:

I have looked at many scheduling issues in Edge Cloud cooperation because scheduling strategies are becoming more and more popular with the development of EC (Edge Cloud) scenarios. Many distributed systems have adopted this popular technique as a guiding concept because most User Equipments (UEs) require communication with near-edge cloud servers. The goal of the edge cloud servers' latency is to reduce the overall Lighted Response Time in order to solve the latency problem. Computational resources are allocated fairly across all active activities according to their Lights on the edge cloud servers using a two-phase task scheduling approach for task allocation, which also improves the overall cost while taking deadlines into account. The job allocation model selects the least expensive cloud-lets One paper focused on solving the caching problem different from the solution based on neighbourhood search concept, which can shorten the execution time, delay workload allocation cost by dividing sub-problems like resource allocation phase, classification phase, and clustering. Few study indicates that the resources required for computing are more. The problem is addressed on the cost of switching which is based on the previous data on configuration in the edge cloud.In a different research, the cache placement issue was addressed by reducing task execution delays in an Edge Cloud scenario where the model was partially trained locally on raw data to minimise data transfer and maintain privacy. Another study from 2013 examined the issue of content caching in the technology of the Internet of Things by taking into account the devices' constrained computational capacity to minimise transmission latency. Additionally, two algorithms—one for data allocation to operate appropriate edge nodes and the other for data caching scheme to select the right cache function—are put forth. These algorithms can aid in resolving the

cellular network's resource allocation issue for content caching in the context of mobile edge computing.This research, however, approaches the caching problem in a different way by examining a different multipliers-based algorithm that uses partial knowledge to maximise system utility in response to cooperation joint service placement caching issue with user requests. It accomplishes this by putting forth the ICE iterative algorithm, which lowers response time in Mobile Edge Computing and is based on Gibbs sampling.

• Policy for Dispatching:

According to my model, a task must be sent to either the cloud or an edge server that has been equipped with an application. The cloud would then view this as a unique edge server with limitless processing capacity because there is no waiting time for tasks on the cloud hence this technology can be used hence forth.

5.RESULTS AND DISCUSSION

The time needed for processing and releasing of the work are stated by Google cluster data collection. The job weights were established based on the Google cluster trace's definition of job priority, which ranges from 0 to 11. We then adjusted the range to obtain positive WRTs. Across the course of 24 days, over 1,35,000 jobs from a heterogeneous cluster of 13,469 compute nodes were gathered and stored in the data-trace. Due to the fact that most of the jobs are small and involve brief data analytics, their features align with those of latency-sensitive activities in EC. I have selected roughly 50,000 jobs at random from all of the occupations and arranged them into ten non-overlapping groupings. Every experiment carried out on these ten work sets was used as a basis for reporting performance results, which are just averages with upload/download delays to the edge clouds and remote clouds set to be between 0.01 and 0.03 seconds and 0.2 and 0.4 seconds, respectively. For ease of calculation, the processing time of the edge-cloud servers was adjusted to roughly 0.75 times that of the remote cloud server.

6. CONCLUSION

Here an online method based on the speed augmentation model and demonstrate that it is (1+) speed and (1/) competitive for any small constant, showing that it is scalable, with the goal of minimizing the total WRT of all the jobs. Subsequently, I modify the suggested algorithm to include fairness by including a fairness knob, which makes the suggested online algorithm simple to apply in distributed systems. In this study, I address the online job dispatching and scheduling problem in edge-cloud systems, where jobs might be offloaded to unrelated servers with delays in both upload and download, and are released from mobile devices at random times and in arbitrary orders.

As a result, when compared to other heuristic algorithms, my simulations based on actual workload traces demonstrate that my algorithm may operate effectively and aid in lowering the total WRT in edge clouds. The suggested technique in this research is aware of the transmission delay and processing time of a job on various servers at the moment of release. This is because, in real-world scenarios, it might be challenging to precisely estimate such data in a dynamic network environment. In order to further investigate other fair job scheduling policies that are based on fair job dispatching policies that should also be meaningful, it would be interesting to look into the job scheduling and dispatching problem in the future while also taking network congestion and inaccurate job information estimation into consideration.

7. REFERENCES

[1] H. Tan, Z. Han, X.-Y. Li, and F. C. M. Lau, “Online job dispatching and scheduling in edge-clouds,” in Proc. IEEE INFOCOM, May 2017.

[2] M. Jia, J. Cao, and W. Liang, “Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks,” in Proc.INFOCOM, Jun. 2015, pp. 725–737.

[3] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, “Efficient algorithms for capacitated cloudlet placements,” IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 10, pp. 2866–2880, Oct. 2016.

[4] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing," IEEE J. Sel. Areas Commun., vol. 37, no. 3, pp. 668–682, Mar. 2019.

[5] H. Tan, S. H.-C. Jiang, Z. Han, L. Liu, K. Han, and Q. Zhao, "Camul:Online caching on multiple caches with relaying and bypassing," in Proc. IEEE INFOCOM, Apr./May 2019, pp. 244–252.

[6] T. He, H. Khamfroush, S. Wang, T. L. Porta, and S. Stein, "It's hard to share: Joint service placement and request scheduling in edge clouds with sharable and non-sharable resources," in Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jul. 2018, pp. 365–375.

[7] L. Wang, L. Jiao, T. He, J. Li, and M. Mühlhäuser, "Service entity placement for social virtual reality applications in edge computing," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2018.

[8] Q. Liu, S. Huang, J. Opadere, and T. Han, "An edge network orchestrator for mobile augmented reality," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2018.

[9] R. Yu, G. Xue, and X. Zhang, "Application provisioning in fog computing-enabled Internet-of-Things: A network perspective," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2018.

[10] S. Im and B. Moseley, "An online scalable algorithm for minimizing knorms of weighted flow time on unrelated machines," in Proc. SODA,2011.

[11] W. Wang, S. Ma, B. Li, and B. Li, "Coflex: Navigating the fairness-efficiency tradeoff for coflow scheduling," in Proc. INFOCOM, May 2017.

[12] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jul. 2016, pp. 1451–1455.

[13] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Mar. 2017, pp. 1–6.

[14] U. P. Moravapalle, S. Sanadhya, A. Parate, and K.-H. Kim, "Pulsar: Improving throughput estimation in enterprise LTE small cells," in Proc. 11th ACM Conf. Emerg. Netw. Exp. Technol., 2015, Art. no. 20.

[15] S. Venkataraman, Z. Yang, M. Franklin, B. Recht, and I. Stoica, "Ernest: Efficient performance prediction for large-scale advanced analytics," in Proc. USENIX NSDI, 2016.