TASK SCHEDULING IN EDGE COMPUTING

BY

KREETHI KUMARI MISHRA - AP19110010424
VAITLA VENKATA SAI DURGA HANISH - AP19110010426
HARSHINI CHINTALACHERVU - AP19110010442
LEHAR SANCHETI - AP19110010448

UROP REPORT

A report submitted in partial fulfillment of the degree of Btech in Computer Science

Supervised by: **DR. Sambit Kumar Mishra**

Asst. Professor

Department of CSE

SRM University AP

Amaravathi

ABSTRACT

Normal cloud computing is not enough to meet the various data processing needs of today's intelligent society, so edge computing has emerged. It is a new computer paradigm for calculating the edge of the network. In contrast to computation in cloud, it elaborates user proximity and proximity to the data source. It is easy on the edges of the network to store local, small and processed data.The edge paradigm, which is intended to be a leading computation due to its low latency, also faces many problems and challenges due to computational capabilities and resource availability, especially in many human situations of population .Edge computing allows edge devices to discharge heavy loads,computational operations on the remote server. This allows you to take full advantage of server-side of edge in computing and storage, as well as to successfully perform computing tasks. However, if they uninstall/offload all highly compressed computing operations on a remote server at the same time, it may become overcrowded, resulting in high processing delays for many computing operations and unexpectedly high power consumption. On the other hand, device resources on the idle edge may be wasted, and rich cloud facilities may be used less. As a result, it is important to investigate the collaborative planning process(scheduling) for a desktop/edge server with a cloud server, cloud computing, and edge devices based on task features, development objectives, and system status. It can assist in performing all the computing functions efficiently and effectively. This paper analyzes and summarizes computing conditions at the end of the computing edge paradigm context. It then classifies computing functions as edge computing situations/scenarios. It then develops the problem of uninstalling computational loading from the computational system of edge. Collaborative planning methods for computational functions are then updated according to the problem structure.

1. Introduction

With the speedy development of the internet of the whole thing (IoE), the quantity of clever devices related to the internet is growing, resulting in large-scale records, which has brought on problems in traditional cloud computing models which include bandwidth load, sluggish reaction velocity, bad security, and bad privateness.despite swiftly increasing computing electricity, they may be not able to attain real-time and green execution due to confined computing assets and ever demanding applications. Cloud computing can process enormously complex computing duties and offerings in order to obtain device cloud collaboration.

In a cloud computing version, customers can depend on a cloud computing center's extremely rich storage and computing sources to amplify the computing and storage strength of devices and achieve the rapid processing of computationally intensive tasks However, there are some disadvantages to the device cloud collaboration mode, such as excessive transmission delay and exceeding community bandwidth requirements.consequently conventional cloud computing is now not enough to satisfy the various records processing wishes of latest clever society, so edge computing technology have emerged.

Edge computing isn't always the same as conventional cloud computing. it is a new computational model that performs computation at the network edge. Its crucial idea is to bring computing in the way of the information's source. edge computing is defined in another way by means of researchers. Shi et al. introduced the concept of side computing, stating, "Edge computing is a new computing mode of execution at the community's edge." The Edge computing downlink records represents cloud service, the Edge computing uplink information represents the internet of the entirety, and the edge computing part refers to the arbitrary computing and community resources among the information source and the path of cloud computing center."In other phrases, edge computing is the migration of the cloud's network, computing, storage competencies, and resources to the network's edge, and the availability of wise services on the network's edge to satisfy the important wishes of the IT industry in agile linking, real-time commercial enterprise, statistics optimization, application intelligence, safety, and privacy, in addition to assembly the community's low latency and excessive bandwidth necessities.

Edge computing is proposed to remedy the problems of cloud computing for data processing by using preferred computing offerings for users by utilizing computing, community, garage, and other resources on edge that's close to a physical entity or data source. while in comparison to cloud computing, a few edge computing user applications may be processed on an edge server near smart devices, substantially reducing data transmission delays and network bandwidth load required in edge-cloud collaboration. Edge computing gains another advantage by means of disposing of long-distance data transmissions encountered in devices–cloud computing, namely, the latter can more effectively promises consumer data protection. As a end result, using edge computing to finish numerous computing tasks for smart devices has come to be a significant improvement trend.In this report, those devices are called edge devices.

Typical edge computing task scheduling strategies involve releasing all complex computer functions from edge devices to edge servers for processing. However, it may result in the loss of computer and storage resources on edge devices and cloud computing facilities. Additionally, multiple devices may be connected to a edge server at the same time. As a result, the server may be overburdened by computer activities, resulting in a long queue of tasks. This increases the duration of all tasks and queries ordered accordingly in a queue, resulting in delays in processing tasks on the edge server and on edge devices. -With rich cloud facilities may be used sparingly. To address the above-mentioned problems, we can integrate cloud computing, edge servers, and edge devices to effectively manage the computing tasks of edge devices through offloading. We must use cloud computing and storage resources, edge servers, and edge devices, and schedule computational operations on them for processing where necessary, based on computer functionality, development objectives, and system status. It can effectively reduce the load on edge servers, improve service utilization, and reduce the timeframe of computational operations in the system.

This report analyzes the critical problem of organizing computing task scheduling on the edge computing paradigm under IoT. Normal distributed computing task scheduling focuses on distributing and organizing large tasks across multiple similar computing nodes that are dynamic and does not solve download loading problems on edge computing. In Section 2, we take a brief look at past work done on the topic and gather information from it. Section 3 examines computing conditions on the edge scenarios, specifying their design, features, and fields of the application. In Section 4, we examine and summarize the collaborative planning methods for computer tasks based on computing scenarios, computational tasks, and integrated optimization model. Section 5 concludes this by mentioning section reviews and the main goal also what researchers prefer and we have also added conclusion by mentioning tasks by identifying the remaining issues that need to be addressed before we develop the required computational planning program for the edge computing.

2. Literature Review

As technology is widely growing nowadays we are entering into the era of 5G communication, the Internet of Things(IoT). Smartphones are becoming popular, and the Edge Cloud is meeting all the demands of users, especially for delay-sensitive application requirements. Many studies presented various surveys on resource allocation and scheduling in the Edge Cloud. It is important to remember that the Edge Cloud scheduling is in its early stage and there is a lot more to work on. We will see various survey papers that are related to Edge Work which are helpful for our survey. In Edge Computing, a variety of task scheduling has been analyzed. Nevertheless, the majority of work focuses on resource allocation, and less focus on time scheduling.Presently, researchers have done a significant amount of research on the offloading decision problem, such as choosing the best answer after formulating the problem with classical algorithms and increasing the reward to perform appropriate decision actions ith reinforcement learning algorithms. Task offloading and resource allocation can only be improved statically with traditional methods.Although deep reinforcement learning techniques can meet the requirements of today's IoT dynamic task computing, most algorithms enhance the model's convergent time and achieve better results by improving the method during model training, ignoring the optimization of the task interface testing process, resulting in relatively high response delays in real scenarios.

In [14], the shortest-job-first scheduling as the name suggests the scheduling is done on the basis of minimum delay. The Markov decision process is a potent approach for long term objectives when dealing with sequential problems.DRL deals with time and resource allocation in recent years . DQN Algorithm dealt with minimizing resource utilization and delay by using resource allocating strategies using MDP and multiple replay memories.

To minimize the total delay and resource utilization, the computation resource allocation problem in edge computing is formulated as an MDP, and multiple replay memories are used for the deep Q-network (DQN) algorithm.
Jung.at.el[15] evaluated scheduling concepts based on their benefits and drawbacks, energy consumption, SLA violations, resource utilization execution time ,scalability,availability, throughput, makespan time, profit, and various constraints such as fault-tolerance,deadline. Furthermore, various scheduling approaches have been studied to determine which are feasible under certain conditions and which should be avoided.

Paper(year)	Execution Time	Makespan Time	Execution cost	Response Time	Resource Utilization	Energy	Throughput	SLA	Availability	Scalability	Profit
Meng et al (2019)	x	x	✓	x	x	x	x	x	x	✓	x
Huawei et al(2017)	x	x	x	x	x	x	x	x	x	x	✓
Hou et al(2016)	x	x	✓	x	x	x	x	x	x	x	x
Wang et al (2018)	x	x	x	✓	x	✓	x	x	x	x	x
Tuli et al(2020)	x	x	✓	✓	x	✓	x	✓	x	x	✓
Urgaonkar et al(2015)	x	x	✓	x	x	x	x	x	x	x	x
Han et al(2019)	x	x	x	✓	x	x	x	x	x	x	x
Wang et al(2017)	x	x	✓	x	x	x	x	x	x	x	x
Zhong et al(2019)	x	x	x	x	✓	x	x	x	x	x	x
Farhadi et al(2019)	x	x	x	x	x	✓	x	x	x	x	x
Wang et al (2019)	x	x	✓	x	x	✓	✓	x	x	x	x
Liu et al(2021)	✓	x	x	x	✓	x	x	x	x	x	x
Li et al(2021)	x	x	✓	x	x	x	x	x	x	x	x
Yin et al(2017)	x	x	x	x	x	x	x	✓	x	x	x
Zhang et al(2018)	x	x	x	x	✓	x	x	x	x	x	✓
Lee et al(2020)	✓	x	x	x	x	x	x	x	x	x	x
Yu et al(2018)	✓	x	✓	x	x	✓	x	x	x	x	x
Kaur et	x	x	x	x	x	x	x	x	✓	✓	x

Table I : Summary of Scheduling Algorithms with QoS Parameters[15].

3. Computing Scenarios

Edge computing resources in IoT mainly consist of edge servers and services. During task scheduling, the cloud centers are seen as a part of the whole system, so that we can extract most benefits. A cloud centre has numerous high performance computing servers. We must use storage, bandwidth and other resources logically, in order to become efficient. This section

analyzes and summarizes various computing scenarios based on the composition of computing resources.

An edge is present in the architecture, on which the edge scheduler, server and edge devices reside. The server's job is to provide various services, like networking, computing, storage, etc. Resources like bandwidth are required as a support for the tasks in edge computing. Edge devices carry out different computing tasks, or they can offload them to devices which are idle, or directly to the server. They have limited resources that are enough to carry out small computational tasks. They have fewer resources compared to edge servers. The resources and status of all edge servers and edge devices under its supervision help to deliver the task scheduling services. It's a scheduling controller that allows edge servers and edge devices to communicate. However, the edge-computing system does not require it.

Based on the computational resources used in the offloading and scheduling of computational tasks in edge computing, four types of computing scenarios can be classified: edge only, edge with scheduler, edge cloud and finally edge cloud with scheduler. Figure I depicts an edge computing architecture.

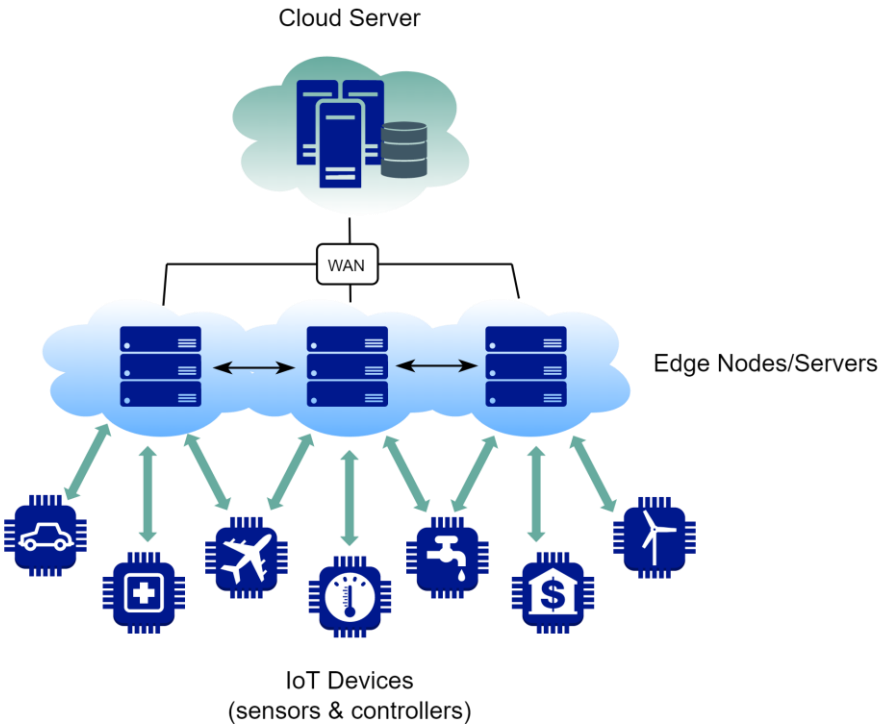

Figure I : Edge computing architecture

3.1 Edge only scenario

The infrastructure that is closely located to the data source is called “Edge”. It's the framework for processing data as near as possible to the source. This architecture needs the effective use of non-networked resources such as smartphones, computers, tablets, sensors. Edge computing, also known as local cloud/fog computing, has a wide range of technologies like wireless sensor networks, peer-to-peer ad-hoc networking, etc.

Similar to smart terminals, edges are also close to the data source. The data is stored and processed at the edge of network. It is aware of its surroundings and provides near-end services to its consumers. The information is real-time secure and also efficient. It can also solve the problem of high cloud energy usage computing, lower expenses, and relieve network bandwidth pressure. Manufacturing, energy, home automation, and transportation are also all examples about where edge computing is used. Edge computing is now more popular because the Internet of Things has risen in popularity. Computing models are in desperate need and are becoming increasingly prominent as research topics.

In this scenario, only edge servers and devices are present but there isn't any scheduler on the edge. So the edge device can choose to compute the task locally or it can offload the task and send it to the edge server. The server computes the task and communicates the results with the corresponding device. It is carried out in a similar way to the offloading in cloud centers. It is the simplest scenario present in edge computing. The edge scheduler is absent here. The locations for offloading tasks are fixed. There are various types of computing tasks predetermined by the resources present in the edge server. It also lacks a cloud computing center. So, it is better suited for processing tasks requiring less computation, tasks sensitive to delay, and process them in a closed environment.

According to the above analysis, the proposed scenario needs to meet the quality of service expectations. The quality of service is assessed by considering the time taken to complete a task. It is assumed that all edge devices and servers are the same. The majority of the content offered can be simply extended to heterogeneous servers & devices, it should be highlighted.

By multiplying the number of packets transmitted by the number of packets transmitted, the completion time of a task that has been offloaded to an edge server can be calculated. Processing time, waiting time and latency are also taken into account.

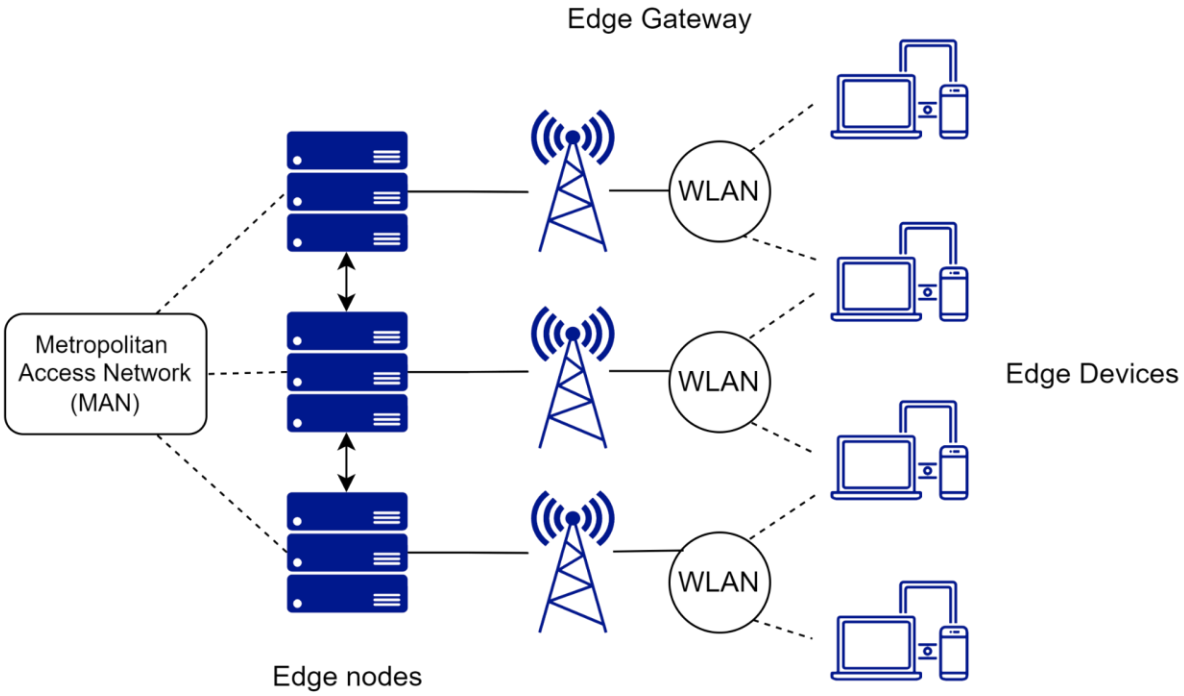

Figure II : Fundamental Edge Scenario

3.2 Edge computing with scheduler

If a single node does not have enough power to compute tasks, we use scheduling for the tasks. This helps to save energy, be efficient, Reduce the completion time for the tasks, and in turn improves the experience of the user .For computing systems, various scheduling strategies are considered. In all cases, ensuring fairness is a critical prerequisite, as many undesirable situations can arise if this is not done.

The architecture in this scenario has an edge scheduler along with the devices and servers. It is different from the first scenario, as it contains a scheduler. The role of this scheduler is to schedule tasks tactically. In this scenario, an edge device can either locally process a computational task or it may perform offloading and assign it to the scheduler. Based on the scheduling policies, the scheduler may assign the task to the devices or other servers. The factors like current computing, task execution, network etc. influenced a lot while creating these policies. These factors are observed in all the edge devices and servers. The scheduler finally sends the computed results back to the source devices. In this scenario, an edge scheduler can schedule the computing tasks reasonably to various different devices and servers.

Various types of resources are available on the edge. The computing tasks are also of various types just like the resources. In the cloud centers, there are vast resources that are available, whereas on the edge, the resources available are very limited. As shown in Figure 3, we can clearly see that this scenario is pretty useful for tasks that need less resources, very little computational time, and strict latency requirements.

In this architecture, the edge scheduler will assign the computing tasks to various servers for them to process. The significant difference between the first and second scenarios is that in this scenario, the scheduler can send tasks to servers on the edge which are idle, by considering their real time status. If the processing time of a task makes up a major part in its total computation time, then this scenario can perform better.

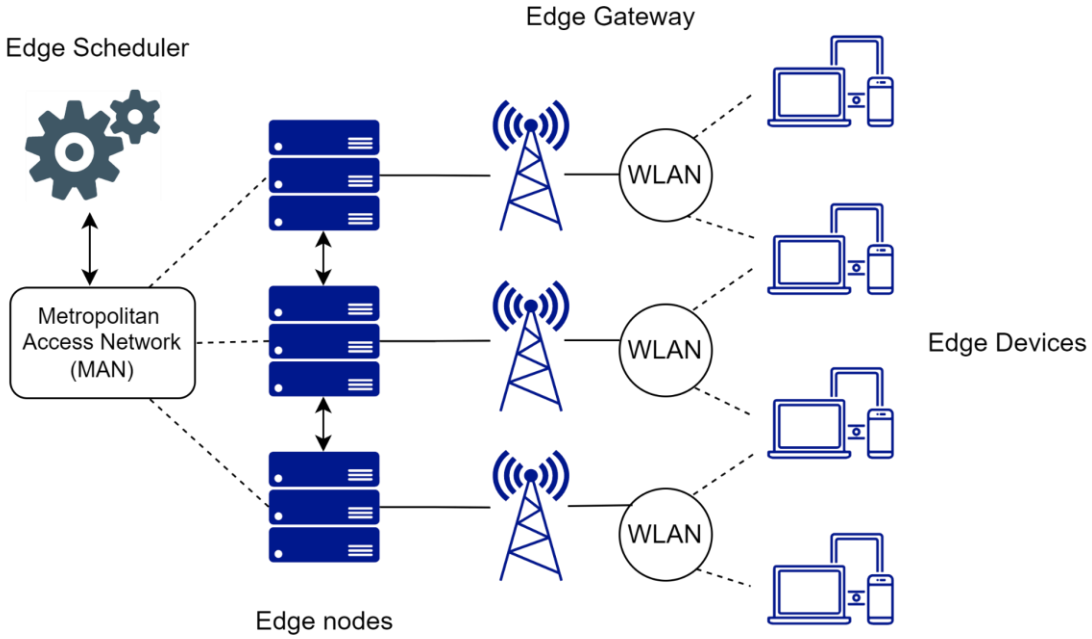

Figure III : Scheduler based edge scenario

3.3 Edge cloud scenario

Traditional centralized cloud architectures will fail to match these services' Quality of Experience (QoE) demands, necessitating a more dynamic and distributed cloud architecture. To satisfy the desired QoE, compute and storage cloud resources must shift closer to the network's edge, where information is both created and consumed.
This scenario contains edge servers, devices, and a cloud hub. But, in this architecture, the edge scheduler is not present. Edge devices are network nodes with sufficient memory, processing power, and computing resources to gather, process, and execute the data almost in real time. On

the basis of computing task requirements, the edge device determines if the task should be locally executed or offload it to a cloud server. It may be partial offloading or full offloading.

The basic edge computing scenario is quite similar to the edge cloud computing scenario but the difference is that in the first scenario, there is no cloud computing center. Here, in the edge cloud, we can offload the tasks to the cloud center by taking the latency and computing requirements into account. This architecture gives us the choice of handling tasks by any of the edge and cloud servers.
Based on the features of their computing processes and user QoS needs, edge devices determine where and how the offloading must take place. It tells if the task must be offloaded to the edge or to the cloud The main feature of both the edge only scenario and edge cloud scenario are the same.

Since this scenario has a cloud center component, tasks that require larger computation power and are insensitive to latency can be offloaded. Thus, the computation power does not have any influence in this architecture. Instead, the environmental resources available on the edge affect the tasks.

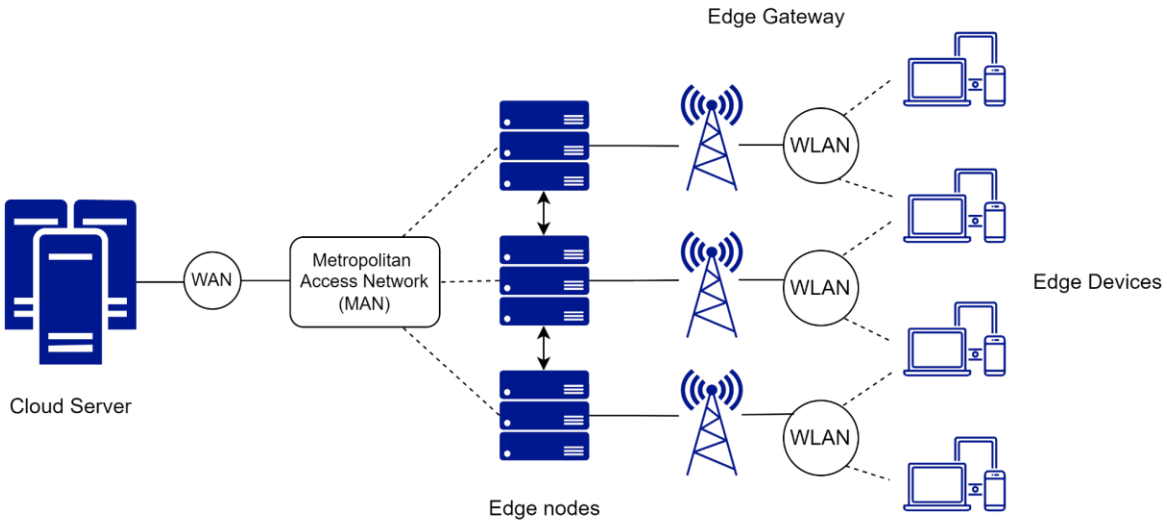

Figure IV : Edge cloud scenario

3.4 Edge cloud computing with scheduler

A scheduler's task is to identify the tasks along with their requirements and schedule them in a strategic manner to make proper use of the resources. The edge scheduler decides whether the tasks must be offloaded or computed locally. It also decides how to divide the computational power among the tasks.

This scenario consists of things that are present in edge cloud computing which are edge devices, edge server, cloud centers along with edge scheduler which was not present in the third scenario. In contrast to the third scenario, the edge scheduler offloads all computing tasks through the devices and servers on the edge. The edge devices can locally compute activities or they can offload these tasks to the server on the edge. The only difference between the previous and present scenarios is that, in this scenario, after the edge devices offload their tasks, they are scheduled and assigned with proper resources and computational power. It can reap the benefits of the interactions among the edge servers, devices and cloud center to its fullest. Types of edge environment resources affect the processing of computing activities, not the amount of computing.

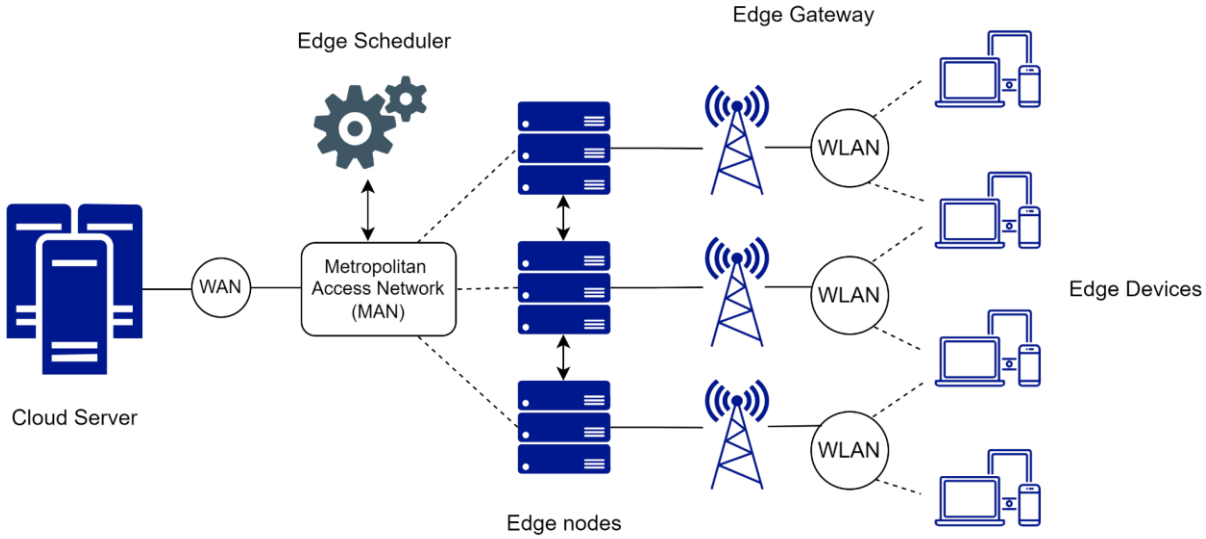

Figure V : Scheduler based edge cloud scenario

4. Different Task Scheduling Analysis

As cloud computing advances, scheduling techniques have gained a lot of attention recently. In this section, we looked at various scheduling issues in Edge Cloud collaboration.

In edge computing nodes are dynamic, and resources are not fully available at a given point of time.During task scheduling nodes should be evaluated in order to find the availability of the resources so that it can prevent unnecessary exploitation of resources.
Task Scheduling can be difficult due to many reasons, some of them are as follows.First As the connection between the edge node and device is wireless which is very unpredictable. Second, There is a huge difference in terms of resource availability ,response time,speed etc for different

VM's. Finally, Factors such as task arrival rate,delay vary in different Iot devices,All these factors make task scheduling a tad more challenging.

Task scheduling ensures low completion time,reduces energy/power consumption,and improves audience experience.There have been numerous proposed algorithm Algorithms such as Max-Mix, Suffrage Algorithm, First come First Serve, Bat Algorithm, Particle Swarm Optimization etc. Many of Algorithms provide optimization,but create load imbalance problems while some algorithms cannot guranetee optimized solutions to the problem.The algorithms are broadly classified into three types namely Heuristic and MetaHeuristic and Traditional Algorithms

Traditional Algorithms mostly use delay as an optimization goal, but this can easily lead to a load imbalance problem among computing nodes.Example Min-Min,Sufferage algorithm etc

Heuristic algorithms usually concentrate on one problem and differ for different problem, and produce reliable result in a specific field of problems.Because heuristic algorithms works with specific domain,These algorithm are good for time constraints as they provide solution in a fixed time frame but they cannot guarantee optimality of the solution .Example of Heuristic Algorithm are Genetic Algorithm,Simulated Annealing,Tabu Search

MetaHeuristic algorithms in contrast cover a broader range of problems. They do not target any particular problem and seek near-optimal solutions. Example of MetaHeuristic algorithm Novel Directional,Catastrophic Genetic Algorithm,SOP etc.

We tried to summarize and compare various algorithms on the basis of minimum response time,power consumption, minimum completion time,cost effectiveness

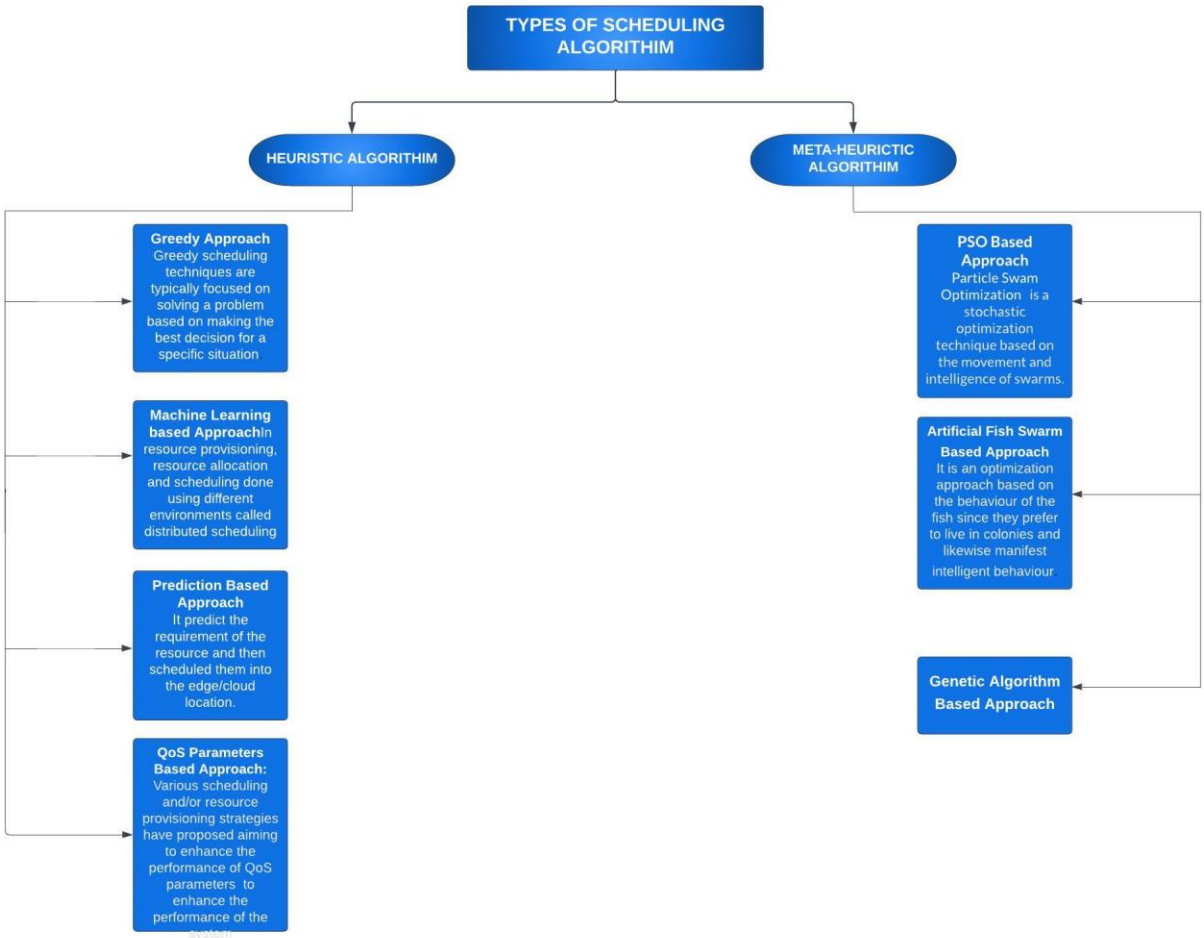

Diagram : Types of Scheduling Algorithm

4.1 Minimum Response time

Edge servers in edge computing networks can provide computing or storage resources, but these are very limited. Algorithms which can fulfill these parameters and can provide optimal solutions and allocate resources are needed.Response time can be defined as the time required to process a request to the server followed by setting VM and response from server for offloading. Fast response times are important as it improves user Quality of Experience.

Response time refers to time required to respond to a request for service. Hu et al [1] proposed an algorithm using NOMA based technique. In the experiment, they formulated the transmitting

power allocation problem and addressed the PA problem with the quasiconvex technique and presented a model (NCGG). Then, in order to minimize request response delays, they combine Joint request offloading and resource scheduling and created an algorithm .The Algorithm provides shorter response time but when bs get full it reduces the response time . Zhang et al. [2] used a greedy approach where they send tasks which have the shortest response time .The result showed that it has a shorter response time than random task allocation. Tuli et al[3] proposed an algorithm using Real time scheduler ,its experiment results show shortened energy consumption,and response time. Quin et al [3] approached to solve real world problems using workload allocation algorithms. His main goal was to reduce response delay and user resources on the network wisely. Tan et al. [4] added weight to response which indicate delay sensitivity I.e task with more sensitivity will have more priority .Wei at el [5] proposed a back-propagation (BP) neural network-based service cache selection algorithm. The result showed significant decrease about 13.2 percent decrease in delay time.Ma et al [6] created a Iterative Caching update algorithm which helped in reducing the response time.Sajnani et al. [7] proposed a novel Multi-Layer Latency Aware Workload Assignment Strategy (MLAWAS) algorithm which finds the best cloudlet to unload User Equipment .Furthermore, in order to achieve a faster response time, this algorithm has sacrificed some latency. Han et al.[8]used Speed augmentation model ,experiment shows it reduces the total weighted response time.

4.2 Cost Effectiveness

One of the main issues in edge computing is how to reduce cost of edge computing while offloading tasks. We develop a task scheduling model for the delay-sensitive input tasks.In edge computing offloading a task can be troublesome and costly,It is very necessary to maintain reduced time but considering the cost simultaneously . Y.et.al[9] proposed Two-stage Task Scheduling Cost Optimization where they used a NP-hard cost optimization problem and the result showed reduced cost considering the needed delay requirement .A real time scheduler using A3C was used in multiple agents.The result showed 4.67% decrease in running cost. Li et al used Lyapunov based strategy where he exploit spatial temporal diversities to reduce cost and load balancing but the prior knowledge of all the resources are needed.Chen et al proposed a Framework named DCC which consist of a device layer,cloud layer and cloudlet layer.In the framework the task which required high computation is offloaded to either of the cloud layer or cloudlet layer whereas task with low computational cost is offloaded to device layer, distribution of task in this way reduce the delays in processing. Another greedy algorithm was proposed which used a graph partition approach i.e combining greedy optimization approach and computational capabilities of devices which can minimize communication cost.Online Predictive Algorithm predict the dynamic information of task on the basis of edge network and server load. It uses Long Short-Term Memory and Deep Reinforcement Algorithms to make offloading decision on factors such as energy consumption,processing and tolerance delays,to prevent

server overloading and network error which help in reducing task dropping rate resulting in minimizing cost.

The major cost involved in task scheduling is the cost of transmission and cost of computation and cost at the edge server. Majority of the work done in reducing cost took first two into consideration but cost at the edge server is rarely consider in these works

4.3 Power Consumption

Power Consumption can be due to many factors but resources are one the most important , Resources can be local,remote,tasks and network.The resources determines energy consumption based on location remote clouds,bandwidth, size of data, it can also very from path taking to transfer data from local nodes to remote sources

Markov chain theory discusses the average delay and power consumption under task scheduling constraint with the help to this power constrained delay minimization problem is formulated .One of the major power enhancing factor in edge computing is transferring of data ,S. et al. [10] Showed assignment of tasks on cloudlets and on public clouds in such a way that delays are maintained with minimum energy consumption is minimum . Xu et al. [11] propose a PSO for offloading tasks to edge servers. It takes into account various types of computing resources and aims to reduce mobile device energy consumption while meeting response time constraints. The results show convergence and adaptability resulting in optimization
Sun et al. use SCP and Independent Lyapunov techniques which improve the power consumption task dispatching .

Computational demand is increasing, as is power consumption in the Data Center (DC). This massive consumption generates carbon footprints that are unfriendly to the environment. When there is a multiple resource in the system, the energy consumption problem becomes more difficult.

4.4 Minimum Completion Time

Completion time is defined as the total time required to transmit data to server followed by processing task and execution of the task and finally waiting for the response from server.So In order to minimize the completion time we can reduce any of the above mentioned components .

Li et al. [12] proposed an approach to solve the caching problem that differs from the previous solution and is based on the neighborhood search concept, which reduces time, delay, and workload cost. A Greedy Approach with threshold strategy was proposed and the result showed

good dynamic performance,acceptable energy consumption. The Markov Decision aimed at reducing power consumption uses queuing state of task buffers to schedule tasks.The result showed shorter execution delay in comparison to the existing algorithm .Chen et al [13] worked on combined VM resources to maximize the longevity of QoE parameters .He proposed an intelligent scheduling framework which considered expected delay requirements.
The main objective of the algorithm is to determine the task execution order, and the other is assigning the task to the suitable VM.

5. Conclusion

This paper discussed different methods of computing scenarios which are mainly divided into four types: basic, scheduler-based, edge-cloud computing, and scheduler-based edge-cloud computing.,We further discussed about different offloading method which are divided into two types partial offloading and full offloading.Then we discussed about different type of algorithm for task scheduling which we divided into three type Traditional,Heuristic or MetaHeuristic,later we discussed about different algorithm and previous studies on the basis of Response Time,Execution time,power consumption and cost effectiveness.Due to the computational complexity of the reviewed problems, researchers prefer to use improved heuristic algorithms over mathematical programming algorithms.The main goal of this paper is to provide a deeper understanding of task scheduling algorithm in the Edge computing , paving the way for the development of more optimized and cutting-edge scheduling approaches.

References

[1] S. Hu and G. Li, “Dynamic Request Scheduling Optimization in Mobile Edge Computing for IoT Applications,” IEEE Internet of Things Journal, vol. 7, no. 2, pp. 1426–1437, Feb. 2020, conference Name: IEEE Internet of Things Journal [CrossRef]

[2] G. Zhang; W. Zhang; Y. Cao; D. Li; L. Wang Energy-Delay Tradeoff for Dynamic Offloading in Mobile-Edge Com-puting System with Energy Harvesting Devices. IEEE Trans. Industr. Inform. 2018, 14, 4642–4655. [CrossRef]

[3] Z. Qin, Z. Cheng, C. Lin, Z. Lu, & L. Wang (2021). Optimal Workload Allocation for Edge Computing Network Using Application Prediction. *Wireless Communications and Mobile Computing*, *2021*. [CrossRef]

[4] H. Tan, Z. Han, X. Y. Li, & F. C. Lau (2017, May). Online job dispatching and scheduling in edge-clouds. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications* (pp. 1-9). IEEE. [CrossRef]

[5] H. Wei, H. Luo, & Y. Sun (2020). Mobility-aware service caching in mobile edge computing for internet of things. *Sensors*, *20*(3), 610. [CrossRef]

[6] X. Ma, A. Zhou, S. Zhang, & S. Wang (2020, July). Cooperative service caching and workload scheduling in mobile edge computing. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications* (pp. 2076-2085). IEEE. [CrossRef]

[7] D. K. Sajnani, A. R. Mahesar, A. Lakhan, & I. A. Jamali (2018). Latency aware and service delay with task scheduling in mobile edge computing. *Communications and Network*, *10*(04), 127. [CrossRef]

[8] Z. Han, Tan, H., Li, X. Y., Jiang, S. H. C., Li, Y., & Lau, F. C. (2019). Ondisc: Online latency-sensitive job dispatching and scheduling in heterogeneous edge-clouds. *IEEE/ACM Transactions on Networking*, *27*(6), 2472-2485. [CrossRef]

[9] Y. Zhang, X. Chen, Y. Chen, Z. Li, & J. Huang (2018, July). Cost efficient scheduling for delay-sensitive tasks in edge computing system. In *2018 IEEE International Conference on Services Computing (SCC)* (pp. 73-80). IEEE. [CrossRef]

[10] S. Yang, F. Li, M. Shen, X. Chen and X. Fu, "Cloudlet placement and task allocation in mobile edge computing", *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5853-5863, Jun. 2019. [CrossRef]

[11] Z. Sheng, S. Pfersich, A. Eldridge, J. Zhou, D. Tian and V. C. M. Leung, "Wireless acoustic sensor networks and edge computing for rapid acoustic monitoring," in IEEE/CAA Journal of

Automatica Sinica, vol. 6, no. 1, pp. 64-74, January 2019, doi: 10.1109/JAS.2019.1911324.[CrossRef]

[12] C. Li, Y. Zhang, & Y. Luo (2021). Neighborhood search-based job scheduling for IoT big data real-time processing in distributed edge-cloud computing environment. *The Journal of Supercomputing*, *77*(2), 1853-1878. [CrossRef]

[13] S. Sheng, P. Chen, Z. Chen, L. Wu, & Y. Yao (2021). Deep reinforcement learning-based task scheduling in IoT edge computing. *Sensors*, *21*(5), 1666. [CrossRef]

[14] C. Li, J. Tang, H. Tang, & Y. Luo (2019). Collaborative cache allocation and task scheduling for data-intensive applications in edge computing environment. *Future Generation Computer Systems*, *95*, 249-264. [CrossRef]

[15] Asghar, Hassan & Jung, Eun-Sung. (2022). A Survey on Scheduling Techniques in the Edge Cloud: Issues, Challenges and Future Directions. [CrossRef]