Task Scheduling in Edge Computing

By

KREETHI KUMARI MISHRA - AP19110010424

VAITLA VENKATA SAI DURGA HANISH - AP19110010426

HARSHINI CHINTALACHERVU - AP19110010442

LEHAR SANCHETI - AP19110010448

Mentored by

Dr. Sambit Kumar Mishra, Asst. Prof, SRM University AP

Introduction

●In a cloud computing model, users can rely on a cloud computing center's extremely rich storage and computing resources to expand the computing and storage power of devices and achieve the rapid processing of computing intensive tasks.However, there are some drawbacks to the device. cloud collaboration model such as high transmission delay and exceeding network bandwidth requirements.Hence traditional cloud computing is no longer sufficient to meet the adverse data processing needs of today's intelligent society, so edge computing technologies have emerged.

Edge Computing

●Edge computing is a new computing paradigm that performs computation at the network's edge. Its central concept is to bring computing closer to the source of data. In other words, Edge computing is the migration of the cloud's network, computing, storage capabilities, and resources to the network's edge, and the provision of intelligent services at the network's edge to meet the critical needs of the IT industry in agile linking, real-time business, data optimization, application intelligence, security, and privacy, as well as meeting the network's low latency and high bandwidth requirements.

●Edge computing resources in IoT are primarily made up of edge devices and edge servers. To reap the benefits of cloud centers, we must consider them as part of the overall system when task scheduling.

ISSUES AND CHALLENGES

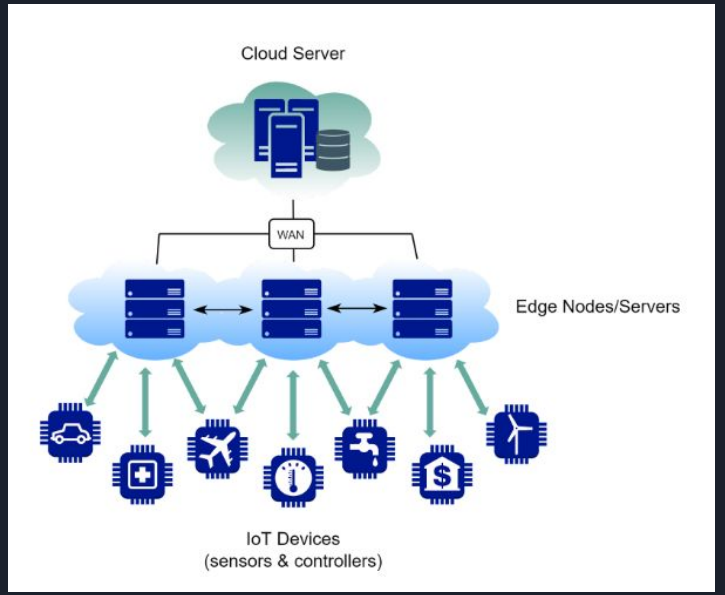
●Traditional edge computing task scheduling strategies involve offloading all computing-intensive tasks from edge devices to an edge server for processing. It may, however, result in a waste of computing and storage resources in edge devices and cloud computing centers. Furthermore multiple devices may connect to an edge server at the same time. As a result, the server may be overburdened with computing tasks, resulting in a long queue of tasks.Many edge devices, on the other hand, may be idle, wasting computing resources, and resource-rich cloud centers may be underutilized. To address the aforementioned issues, we can combine a cloud center, edge servers, and edge devices to efficiently hand' the computing tasks of edge devices through task offloading.

Motivation

- Traditional Cloud computing has more computational complexity and delay , so edge computing is used to decrease the computational complexity and delay and also in terms of backup & restore data and reliability in internet of things (IoT) based industrial applications.Edge computing is a new computing for performing computations at the network's edge.In contrast emphasizes proximity to the user and proximity to the source of the data.
- Edge computing allows edge devices to offload overburdened computing tasks to an edge server.
- Edge computing gains another advantage by eliminating long-distance data transmissions encountered in devices cloud computing, namely,the latter can more effectively guarantee user data security. As a result, using edge computing to complete various computing tasks for intelligent devices has become a significant development trend.

Edge Architecture

- Edge devices: Equipments with limited computing capabilities.
- Edge node: Gateways which can perform computation
- Edge Server: Computational Machines facilitated close to the edge device
- Cloud Server: Hosts, runs and manages edge nodes, maybe public or private



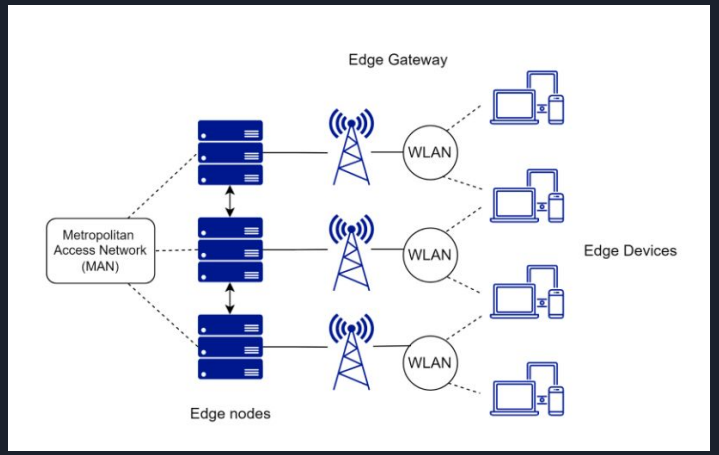
Edge Scenarios

Based on the resources used in the offloading & scheduling of tasks, four types of computing scenarios can be classified

1. Edge-only scenario
2. Edge with Scheduler
3. Edge-cloud scenario
4. Edge-cloud with scheduler

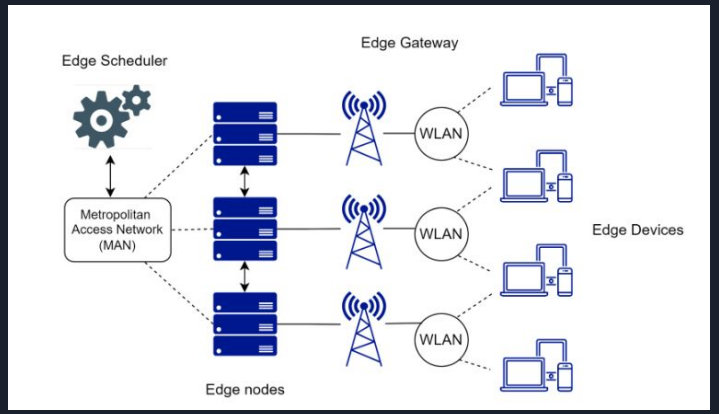
1. Edge-only scenario

- Edge can locally compute the task or offload it to the edge server.
- Suited for tasks that require less computation or strict delay requirements.



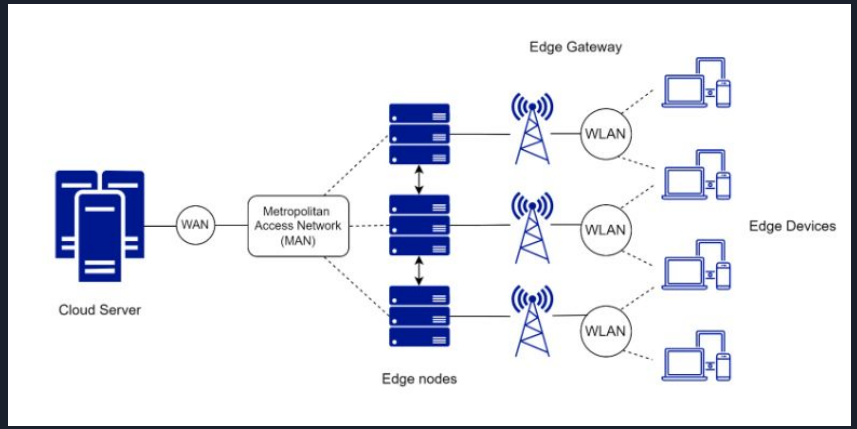
2. Edge with Scheduler

- We use scheduler to schedule the tasks, when the single node does not have enough power.
- Based on the resources available, and the requirements of the task, the scheduler schedules the task.



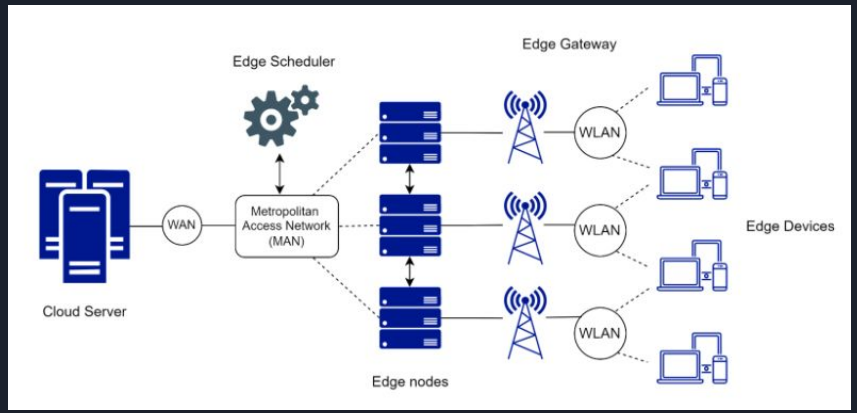
3. Edge cloud scenario

- Based on the computation power the task needs, it maybe offloaded to a cloud server.
- Tasks without any delay restrictions can be sent to the cloud.



4. Edge cloud with Scheduler

- With both scheduler and the cloud server, this architecture can reap the most benefits
- Tasks can be computed locally, or offloaded to edge server or cloud server, or can be sent to the task scheduler.



Task Scheduling

Tasks scheduling algorithms are defined as a set of rules and policies used to assign tasks to the suitable resources (CPU, memory, and bandwidth) to get the highest level possible of performance and resources utilization.

Task scheduling ensures low completion time,reduces energy/power consumption,and improves audience experience.

Limitations:

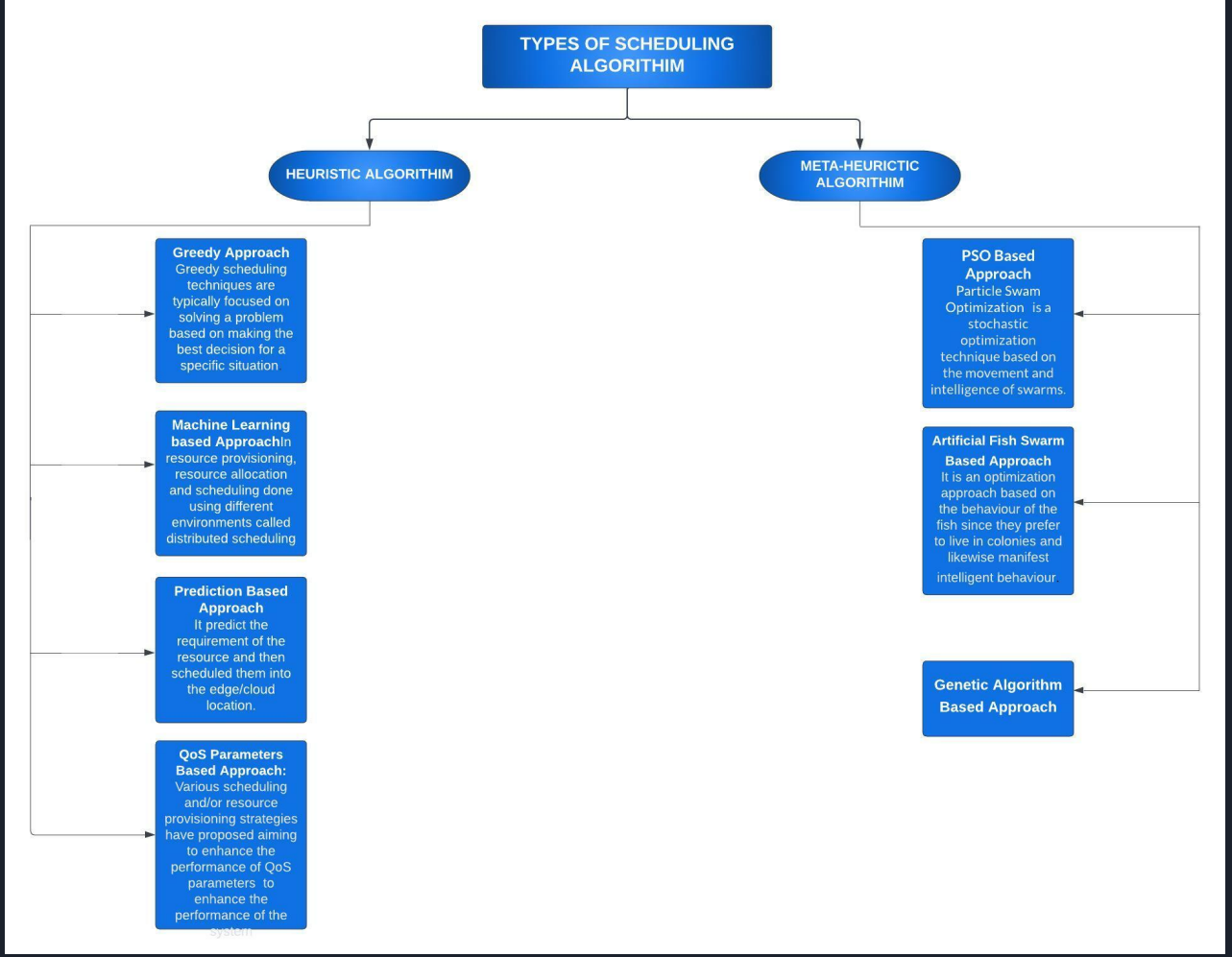
1. Wireless connection between edge server and device
2. Availability of resources varies
3. Factors such as task arrival rate, delay vary in different Iot devices

Task Scheduling Algorithm

The algorithms are broadly classified into two types namely Heuristic and MetaHeuristic Algorithms

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 etc

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.



Types of Scheduling Algorithm

Division of Different Algorithm

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

a. Minimum Response time

- 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.
- Many Algorithm are proposed to minimize the response time some of them are as follows

1. 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 . Then, in order to minimize request response delays, they combine Joint request offloading and resource scheduling and created an algorithm.

   Drawback-: The Algorithm provides shorter response time but when bs get full it reduces the response time

2. 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.

3. 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.

B. 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

C. 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

D. 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.

- Li et al. [2] 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.

- Chen et al [3] worked on combined VM resources to maximize the longevity of QoE parameters .He proposed an intelligent scheduling framework which considered expected delay requirements.

CONCLUSION

This paper discussed different methods of computing scenarios 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.In future we would like to create our own algorithm that helps to reduce any of the above mentioned factors