Task scheduling algorithm in edge computing: Status quo and prospects

Wang Jieyong Computer Science and Technology Class 2019-3

ABSTRACT

With the advent of the Internet of Everything era, the amount of data generated by network edge devices has increased rapidly, which has brought higher data transmission bandwidth requirements. At the same time, new applications also put forward higher requirements for real-time data processing. Traditional cloud computing models It has been unable to deal with it effectively, therefore, edge computing came into being. The basic concept of edge computing is to run computing tasks on computing resources close to the data source, which can effectively reduce the latency of the computing system, reduce data transmission bandwidth, ease the pressure on the cloud computing center, improve availability, and protect data security and privacy. Thanks to these advantages, edge computing has developed rapidly since 2014. In cloud computing scenarios, the general strategy for task scheduling is to migrate computationally intensive tasks to computing nodes with sufficient resources for execution. However, in edge computing scenarios, the massive data generated by edge devices cannot be transmitted to the cloud through existing bandwidth resources The computing center performs centralized computing, and the computing and storage capabilities of different edge devices are different. Therefore, the edge computing system needs to be dynamically scheduled according to the task type and the computing capabilities of the edge device. This article first introduces the research of task scheduling in edge computing Significance and background, then a brief summary of task scheduling, and finally a summary of current research on task scheduling algorithms.

***Index Terms*—Ant Colony Algorithm, Delay perception, Edge computing, Task scheduling.**

1. RESEARCH BACKGROUND

With the popularization of the Internet of Things, the amount of data generated by the Internet of Things terminal equipment is increasing. In order to solve the computational load and insufficient data transmission bandwidth caused by these data in the process of transmission, calculation and storage, researchers began to add data processing functions to terminal devices, the representative of which is mobile edge computing, fog computing and cloud-sea computing.

The advent of the Internet of Everything era and the rapid development of intelligent applications have caused explosive growth in the amount of data in terminal devices. The limitations of terminal device hardware and the mobility of users may cause problems in terms of stability and continuity of services. Edge computing enriches the computing and storage resources of terminal equipment, and improves the processing efficiency of terminal equipment tasks. The resources of edge computing are more sufficient than those on terminal devices, but the resources for supporting task processing such as computing and storage are still limited. Reasonable resource allocation can reduce the processing time of terminal tasks and ensure the stability and continuity of services.

With the advent of the Internet of Everything era and the intelligent development of applications, the number of terminal devices is growing rapidly. Cisco predicts that the number of terminal devices in the world may far exceed the number of people by 2023 (1). Due to the rapid growth of tasks and data, as well as the mobility of terminal devices and limited resources, complex calculation and analysis tasks are difficult to perform quickly on terminal devices. The physical location of edge computing is close to the terminal device or the user, which can enrich the computing and storage resources of the terminal device, and improve the real-time performance and efficiency of the tasks to be processed. Users are users of terminal devices, but also customers and service subscribers of edge computing service providers. Edge computing servers are usually deployed and managed by service providers, and can be used as a small data processing center to process terminal equipment offloading tasks through network resources, computing resources, and storage resources, improving task processing efficiency and reducing delay-sensitive tasks The processing time and the energy consumption of the terminal equipment. Compared with traditional cloud computing, the computing and storage resources that support task transmission and processing in the edge computing server are limited, and the user's terminal equipment is in a mobile state. If the resources are not properly allocated and utilized, it may lead to terminal tasks. The problem of long processing time and low efficiency affects the user’s service experience, the balance of benefits between the user and the service provider, and the quality of service. In this "end-side"

With the rapid development of mobile Internet, Internet of Things and other technical fields, cloud computing technology has also been developed by leaps and bounds. More and more applications are deployed on cloud servers to provide services for low-cost terminal devices. In the intelligent era of "Internet of Everything", more and more terminal devices will be connected to the Internet, and cloud computing networks are under increasing pressure. According to Cisco’s forecast, more than 50 billion devices will be connected to the network in 2020, which means that massive amounts of data will be sent to the cloud computing center via the Internet. In the face of massive data processing, cloud computing is under tremendous pressure. The development of the Internet of Things technology is facing a bottleneck, and there is an urgent need for a computing service that can provide low latency. As a supplement to cloud

computing technology, edge computing technology provides computing services by deploying computing-capable devices at the edge of the network close to terminal devices, which makes up for the deficiencies of cloud computing. The concept of edge computing is to extend cloud computing to network edge processing, so that data obtained from edge terminals can be calculated close to user terminals. At the same time, data that does not need long-term storage does not have to be transmitted to cloud service centers for backup, which is relatively reduced This reduces the occupancy of network bandwidth and reduces the power consumption and load of the cloud.

Terminal equipment has relatively powerful data processing functions, and various new mobile applications continue to appear in front of people. For example, face recognition, road monitoring, human-computer interaction and other applications. Although these applications belong to different fields, they have the same characteristics, that is, they have relatively high requirements for real-time performance, which is the so-called delay-sensitive application. For example, in industrial production, the operating efficiency of the machine will directly affect the output of the factory's products, so under normal circumstances, the factory will monitor the machine in real time. Once a problem is found at the work site, it must respond in a timely manner, otherwise it may cause major problems. loss. Although mobile terminal devices, such as some smart sensors, drones, and vehicle-mounted devices, have been constantly updated to make the processor's data processing capabilities more and more powerful, they still can't fully satisfy users' demands for service quality in the face of massive amounts of data. For complex calculation tasks, it is likely that they cannot be completed in a short time, and long-term calculations will cause a large amount of power loss, resulting in insufficient battery life of the mobile terminal. In order to solve these problems, the terminal equipment can transmit the computing tasks to the central server for processing through the Internet.

Data in the central server calculation has several advantages:

(1) The central server has powerful computing capabilities, can handle complex calculations, and generates low calculation delays.

(2) It can run complex applications and provide reliable guarantee for data calculation

(3) It can provide huge storage space.

But handing over the computing task to the central server also has obvious drawbacks, that is, it has higher requirements for the network environment. When the network communication environment is poor or the terminal device is far away from the central server, there will be a high delay. Therefore, how to dynamically schedule tasks to reduce the time delay and energy consumption of data processing has become an urgent problem to be solved.

2. EDGE COMPUTING

The Internet of Things technology refers to the use of wireless communication technology, network technology, intelligent technology, etc. to jointly construct a global object information sharing Internet of all things network. With the development of Internet-related technologies, we are entering the era of Internet of Everything from the Internet of Things era. Its outstanding feature is that various devices in the network will have the function of sensing the environment, have stronger computing capabilities, and will exist in the network. The Internet of Things technology with billions or even tens of billions of connected nodes, while providing more intelligent functions, will also generate massive amounts of data in the network. How to deal with massive amounts of data is a prominent problem to be solved in the development of the Internet of Things technology. . Based on cloud computing solutions, the data that needs to be processed is sent to the cloud computing center, and the powerful computing power of the cloud computing center is used to centrally solve the data calculation and storage problems [the centralized processing method of the cloud computing solution, although it has solved Resource constraints, but due to the high delay between the terminal device and the cloud server, for tasks with timely requirements, it may bring huge delay time and affect the quality of service. In this case, based on cloud computing Can not efficiently support application services based on the Internet of Everything, edge computing technology as a supplement to cloud computing can better solve these problems.

In an edge computing network, data storage and processing operations will be distributed in edge servers located at the edge of the network, instead of centralized processing of massive amounts of data in cloud servers, which will be carried out in edge servers close to the data producer. The use of the main network is reduced, and the delay time of task processing is reduced. The distributed architecture adopted by the edge computing network, data storage and processing will be distributed in the server at the edge of the network. Compared with cloud computing, it will be extremely Dadi reduces network delays in these processes.

Nearly half of the data generated by the Internet of Things is generated on terminal equipment, and most of the data collected by terminal equipment is multimedia data. Image and video data account for a large proportion. To deal with this huge amount of data, there are currently two One method is to process the computing task on the terminal device, and the other method is to upload the computing task to an edge server or a central server for processing. However, using these two methods alone has certain problems. The first is that computing tasks are processed on terminal devices, which generally have small storage, which means that terminal devices will not have complex task processing models, that is, terminal devices have limited data processing capabilities and it is difficult to handle complex computing tasks. Secondly, the terminal equipment is largely limited by the power supply. The terminal equipment needs to work in a variety of environments, but not every place can provide power. Most of the terminal equipment is operated by internal power supply or solar panels. This means that the power supply capacity of the terminal equipment is insufficient, the endurance capacity is limited, and it is unable to work for a long time. Insufficient power will also cause the computing speed of the terminal device to be limited, and the processing delay of the computing task on the terminal device will be relatively high. Therefore, it is unreasonable to simply let the terminal device handle these data.

If you upload a computing task to the server for processing, you will also encounter many problems. The first is the problem of the network environment. Many of the terminal devices mentioned above use batteries or solar panels to supply power.

Due to insufficient power supply, 3G and 4G are rarely used as high-power transmission methods. They often choose low-power LoRa, ZigBee, etc. The central server conducts data exchange. Both LoRa and ZigBee are low-power LAN wireless standards. Low power consumption means that the transmission rate is slow and cannot cover far away. If the data to be processed is high-definition pictures or other data that takes up a lot of resources , It may take a long time to use the low-power transmission protocol for transmission, and it may even fail to upload the data to the central server. This is obviously unreasonable. Secondly, as mentioned above, terminal equipment may work in various environments. Of course, some areas with poor network signals cannot be avoided. For example, we need to detect forest fire control and early warning. Even if the terminal equipment in the forest uses 3G , 4G is a transmission method with a very high transmission rate, but the forest may not fully cover 3G and 4G signals, and the data may not be efficiently transmitted to the server. If the terminal device is a vehicle-mounted device, it may move anywhere, then the network environment in which the terminal device is located may change from time to time. The instability of the network causes the terminal device to be unable to upload data to the server at any time. It is even more unclear whether computing tasks should be processed locally or uploaded to the server.

At the same time, the server also has an upper load limit. If there are too many terminal devices requesting the central server, the server's resources may not be allocated enough, resulting in a high delay in data processing. Secondly, the locations of servers are often scattered. Some servers are likely to be far away from the terminal device. If data is uploaded to the server, it will have a higher transmission delay. If the application requires high real-time performance, it will definitely be Affect the user experience.

In summary, due to the various conditions of terminal equipment and servers, it is difficult to determine whether the computing task is processed on the terminal equipment or uploaded to the server for processing. Therefore, a dynamic processing method needs to be studied, based on the data processing speed of the terminal equipment. In consideration of factors such as storage capacity, power supply, network status, server load, etc., the data processing location is dynamically selected to minimize the time delay and energy consumption caused by data processing.

By deploying a group of servers close to user equipment, smart device applications can easily obtain back-end computing service support, which solves the current problem of insufficient computing storage capacity and power supply of smart devices, and enables allocation to the cloud for execution The task delay will not be too high. We call such a group of servers on the edge network edge computing nodes. Edge computing nodes are deployed near smart devices and end users, which can reduce data propagation delay and WAN bandwidth usage. The current edge computing nodes are generally managed by a unified computing platform or operator. For example, the edge computing services provided by Amazon, Alibaba Cloud, Huawei Cloud and Microsoft are all managed by The operator uniformly manages the resource allocation of the entire platform. These platforms usually separate the management of the platform from the operation of the service, which is convenient for both enterprises and customers. This is also a typical demand for many modern cloud computing services.

The combination of edge computing and cloud computing will be the development trend of the Internet of Things network architecture. With the advantage of edge servers being close to data producers and the powerful computing and storage capabilities of cloud servers, it will be effective for the massive data generated in the future Internet of Things environment. Storage and processing. In the edge computing architecture, the edge of the network is composed of edge servers to form an interconnected network. The adjacent edge servers are connected to each other as a network. They are connected to the cloud server through the core network. All terminal devices have adjacent edge servers for use. Accept its possible transfer tasks. The computing power, storage capacity, upload bandwidth, download bandwidth, and energy consumption cost of computing unit calculation of edge servers are different. Compared with edge servers, the computing power and storage capacity of cloud servers far surpass the latter.

The network architecture that combines edge computing and cloud computing provides a strong guarantee for the development of the Internet of Things technology, and it also brings new challenges. Unlike cloud computing, in the network architecture combined with edge and cloud, the location of task processing can be selected. For tasks with different requirements and different characteristics, the locations that are suitable for processing are also different. Therefore, the scheduling strategy used in cloud computing cannot be directly processed. Applicable to the current computing architecture combined with edge and cloud. Resource allocation and task scheduling algorithms will play a key role in arranging the location and sequence of task processing. In summary, finding suitable resource allocation and task scheduling algorithms for the edge-cloud combined architecture will make an important contribution to the development of edge computing and the Internet of Things technology.

3. TASK SCHEDULING

In the cloud computing scenario, the general strategy of task scheduling is to migrate computing-intensive tasks to computing nodes with sufficient resources for execution. However, in the edge computing scenario, the massive data generated by edge devices cannot be transmitted to the cloud computing center for centralized computing through existing bandwidth resources, and the computing and storage capabilities of different edge devices are different. Therefore, the edge computing system needs to be based on Task types and computing capabilities of edge devices are dynamically scheduled.

Scheduling includes 2 levels:

(1) Scheduling of cloud computing center and edge equipment;

(2) Scheduling between edge devices.

The scheduling between cloud computing center and edge equipment can be divided into two ways: bottom-up and top-down. Bottom-up is to preprocess some or all of the data collected or generated by edge devices at the edge of the network to filter useless data to reduce transmission bandwidth;

top-down refers to the complex calculations performed by the cloud computing center Tasks are divided and then assigned to edge devices for execution, so as to make full use of the computing resources of edge devices and reduce the delay and energy consumption of the entire computing system. In 2017, Kang et al. designed a lightweight scheduler Neurosurgen, which can automatically distribute the computing tasks of different layers of deep neural networks between mobile devices and data centers, reducing the power consumption of mobile devices by up to 94.7%. The system delay is accelerated by up to 40.7 times, and the throughput of the data center is increased by up to 6.7 times. Dynamic scheduling is also required between edge devices. The computing and storage capabilities of edge devices are different and will change over time, and the types of tasks they undertake are also different. Therefore, tasks on edge devices need to be dynamically scheduled to improve overall system performance and prevent occurrences. The computing task is scheduled to a device under the condition of system task overload. Zhang et al. [143] designed an edge task scheduling framework CoGTA for the delay-sensitive social perception task. The experiment proved that the framework can meet the needs of applications and edge devices.

4. EXISTING WORK

4.1. Edge computing resources and allocation algorithms based on usage forecast

1) Background

The processing of terminal tasks is usually accompanied by resource scheduling of edge computing servers. Therefore, many existing researches rely on resource allocation to complete the offloading and execution of terminal tasks. Different tasks require different amounts of resources and resource conditions. Some tasks do not have special requirements for server resources. The difference in processing on servers with different processing capabilities is only the difference in time; some tasks require deterministic resources, and processing this type of task requires a lot of resources. When the available resources are difficult to meet the minimum requirements for the required resources, this type of task will be difficult to handle. For the latter task type, it is necessary to reserve and pre-allocate resources in advance to ensure the timely processing of this type of task.

The pre-allocation of resources can reserve resources for large indivisible tasks and ensure the timely processing of these types of tasks. At present, the research on the pre-allocation of edge computing server resources is relatively limited. Therefore, the earlier research literature on resource prefetching for cloud computing is also worth discussing. These literatures generally focus on the prediction of task load. The literature combines the status quo and resource distribution of the system, uses virtualization technology to abstract the task scheduling problem as a virtual machine deployment problem, predicts the future workload of the system, and proposes a three-time exponential smoothing algorithm based on a conservative control strategy. On this basis, they proposed a multi-objective constrained power consumption based on probability matching. Optimization model and low energy resource allocation algorithm. Some documents have proposed a clustering and prefetching technology to effectively manage cloud storage resources, identify the trend of different request flows in each category through automatic classification, and complete the pre-allocation of resources in a predictive manner. The above documents have studied the prefetching technology of cloud computing resources, but due to the distributed nature of edge computing servers and the limited resources, the existing research on centralized cloud computing resource pre-allocation may not be suitable for edge computing scenarios.

There have been some literatures on the pre-allocation of cloud computing resources, but the research on the pre-allocation of edge computing server resources is still relatively scarce, and many issues remain to be studied: First, the processing of large computing tasks requires a certain amount of resources. , The available resource amount of the edge computing server covering the terminal device may not be enough to process the task. At this time, the task may be in a waiting state until the resource amount meets the demand, resulting in that this type of task cannot be processed for a long time. Second, the existing research focuses on the optimization of system performance, and rarely starts from the satisfaction of different types of users and service providers. Third, in actual scenarios, the user's task arrival time is not completely synchronized, so offloading tasks and resource allocation algorithms are difficult to complete through batch processing. In the process of resource pre-allocation, resource transactions may require the participation of a third party, leading to leakage of sensitive information and waste of resources.

2) Results

The proposed edge computing resource pre-allocation algorithm based on bilateral auctions explores the resource auction scheme with the highest satisfaction for both parties involved in resource transactions according to different personality characteristics. This algorithm avoids the waste of resources caused by the intervention of third parties in the existing resource transaction scenarios, and can take into account the role needs of users and service providers, while taking into account the processing delay and cost of terminal equipment tasks and the benefits of service providers. Improve the overall satisfaction of both parties by adjusting the parameters according to the special needs of the scene. Under the premise of protecting the privacy information of the terminal, it can accurately predict the destination and resource demand of the mobile terminal device, and reduce the processing of terminal tasks

Aiming at the problem of resource pre-allocation in edge computing, a DAER algorithm for edge computing resource pre-allocation based on usage prediction is proposed. First, considering the high-speed mobility of terminal equipment, a state prediction model is established to predict the staged destination of terminal equipment. Secondly, a resource demand model and an auction model are established to describe the relationship between users and service providers. In the modeling process, the individual consciousness and behavior preferences of users and service providers are considered, and users and service providers are divided into multiple types according to the satisfaction evaluation criteria. Finally, the

optimal resource pre-allocation scheme is obtained based on the genetic algorithm to maximize the overall satisfaction of users and service providers.

4.2. Edge computing task scheduling strategy based on latency awareness

1) Background

The advantage of the edge computing network is that it provides nearby data processing resources for data producers, which greatly reduces the amount of data transmitted to the upper server. The traditional static task scheduling strategy in the edge computing system lacks the connection to the server at the edge of the network. When the terminal device transmits the original data to the edge server that directly communicates with it, if the server cannot process all the modules of the application, The modules that exceed the capacity are placed upwards, ignoring the edge servers in the same layer that have lower communication delays. Aiming at the shortcomings of traditional static task scheduling, a dynamic task scheduling strategy based on delay awareness is proposed for edge computing network architecture. And compared with the traditional static task scheduling strategy.

The proposed latency awareness-based dynamic task scheduling strategy (Latency Aware Online task scheduling strategy, LAO) runs on edge servers deployed at the edge of the network, and no monitoring nodes are set up between the edge servers. The goal of this task scheduling strategy is to find a suitable server in the edge computing network to process tasks in order to achieve low latency in task processing. Due to the use of a dynamic task scheduling strategy, it is allowed to dynamically create virtual machines and dynamically allocate resources for virtual machines in the server.

2) Results

Aiming at the task scheduling problem in the edge computing system, a dynamic task scheduling strategy based on delay awareness is designed. The response delay of the application is kept at a low level. Finally, the effectiveness of the strategy is proved through simulation experiments in a simulated environment.

4.3. Task scheduling algorithm based on improved ant colony algorithm

1) Background

Due to the heterogeneity of data in edge computing and the limitations of computing and storage resources, the use of limited resources to achieve effective support for delay-sensitive applications is a key issue that needs to be resolved urgently. Then, whether to realize the reasonable scheduling of tasks is one of the ideas to solve the above-mentioned problems. At present, some researchers have proposed some solutions for task scheduling, specifically: some use dynamic resource scheduling algorithms to realize resource allocation at the edge; use resource matching algorithms to allocate cloud data center resources on the edge, through both Collaboration to achieve low latency and high quality of service for edge computing. Some achieve global optimal scheduling by changing the sorting and allocation method of Storm's Task instances and the mapping relationship between the two; at the same time, in view of the defects of topological task concurrency, a scheduling strategy based on bat algorithm is proposed to meet the needs of edge nodes. High real-time processing requirements of the time. Some are building a real-time data processing application on the edge network, hoping to reduce the amount of data transmission in this way to reduce energy consumption. Some combine the optimal placement of data blocks and optimal scheduling of tasks to reduce the calculation delay and response time of submitted tasks and improve the experience of edge computing users. The above algorithms have improved the efficiency of task completion from the perspective of different needs, but fewer people pay attention to the unbalanced load of computing resources due to unreasonable task scheduling, which makes the limited resources in the edge environment unable to be fully utilized, thereby improving User response time. In this article, based on the ant colony algorithm, the task resource demand is used as the constraint to realize the load balancing of the edge task distribution, so that the limited edge resources can exert the maximum effect.

The algorithm abstracts the problem to be solved into the following model:

The number of tasks received by an edge node at a certain moment is n, expressed as $T_N = (t_{n_1}, t_{n_2}, \dots, t_{n_n})$; task i's demand for resources is described as $(d_{cpui}, d_{gpui}, d_{memi})$, where d_{cpui} represents the task For CPU requirements, d_{gpui} represents the task's demand for GPU, and d_{memi} represents the task's demand for memory; and the node needs to allocate these tasks to the edge network where it is located. The number of computing nodes that can be allocated in the edge network is m, expressed as$P_M = \{p_{m1}, p_{m2}, \dots, p_{mm}\}$; The resources available in node j are described as $(r_{cpuj}, r_{gpuj}, r_{memj})$ where r_{cpuj} represents the available amount of CPU of the node, and r_{gpuj} represents the available amount of GPU of the node, r_{memj} represents the amount of memory available for the node.

The assignment relationship between task T_N and node P_M can be expressed by matrix A as:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \tag{4-1}$$

Here a_{ij} expressed as the corresponding relationship between task T_i and node P_j. $a_{ij} \in \{0,1\}$, $i \in \{0, 1, \dots, n\}$, if $a_{ij} = 1$ it means that task i is assigned to node j.

2) Results

In the edge computing environment, the resource gap between edge nodes is too large and the load of task allocation is unbalanced. A task scheduling method based on ant colony optimization algorithm is proposed. The method takes the difference in the requirements of different tasks for computing resources such as CPU, memory, and bandwidth as the constraint conditions for task selection of edge nodes, and aims to achieve overall load balance in the edge cloud, and improves the heuristic factors, pheromone update and other conditions.

Improve the overall calculation efficiency of the algorithm, reduce the calculation time, and finally obtain the optimal allocation method by using the ant colony algorithm to realize the reasonable allocation of tasks in the edge environment. The method can avoid the deployment of tasks of the same type in the same node to improve task execution efficiency and computing resource utilization. The simulation experiment results show that the algorithm can allocate more tasks under the same number of nodes, and the overall edge node has a lower load imbalance under the same number of tasks, which improves the utilization of resources and reduces the overall task calculating time.

Compared with polling algorithm (RR) and traditional ant colony algorithm (ACO), the proposed algorithm increases the maximum number of tasks that can be carried, and improves the efficiency of edge nodes, thereby reducing the total traffic in the network center and reducing the response requirements for users. Average response time; and the IAC algorithm will generate multiple approximate optimal solutions each time, which can be freely selected according to needs. Through the comparison of the three figures, it can be seen that when all the small tasks are small, the gap between the algorithms is very small. When the large tasks are added, it can be seen that the algorithm proposed in the article is better. At the same time, if the load demand is not high, the distribution efficiency can be improved by reducing the number of ants, and the running time of the algorithm can be reduced.

When the number of edge nodes is constant, the load balance of the proposed IAC algorithm is better than the remaining two algorithms as the number of tasks increases, and there is a clear gap in the degree of excellence. This is because as the number of tasks increases, there are more possibilities for allocation during the allocation process, and an excellent allocation method is more likely to appear. Therefore, the overall load balance of the algorithm shows a downward trend. In summary, it can be concluded that the IAC algorithm has a good performance compared to the remaining two algorithms in terms of load balance and task allocation.

The load balancing of edge nodes is the next very important issue in edge computing. The current method in terms of resource allocation mainly uses time delay as the criterion to ignore the limitations of the edge environment. Aiming at the limitations of resources in the edge environment, the article first describes the task allocation problem of edge nodes; then designs an improved ant colony Algorithms are used to solve this problem. The algorithm uses the resource requirement of the task as a heuristic factor, changes the pheromone update method, accelerates the convergence speed, and improves the quality of the final solution. The results of simulation experiments show that the algorithm can effectively reduce the load imbalance of nodes.

4.4. Edge computing resource scheduling algorithm for real-time streaming data processing

1) Background

With the rapid development of the Internet of Things technology, more and more terminal devices such as drones, implantable medical devices, vehicle radars, etc. are deployed on a large scale in various life scenarios, and continuously collect environmental information such as images and videos. To support real-time data processing tasks such as electric power inspection, intelligent medical treatment, and unmanned driving. However, due to the limited computing power of terminal equipment and the high cost of communication with the data center (core network), the traditional "terminal equipment-data center" architecture is difficult to meet the demand for low latency for real-time data computing tasks. . In recent years, researchers have begun to use the framework of intelligent edge computing to realize task offloading. The basic idea is to deploy computing equipment at the edge of the network close to the data source to form a "terminal equipment-edge node-data center" system Architecture, and dynamically deploy applications in edge nodes or data centers based on the monitored task load information, which can not only make full use of edge devices with stronger computing capabilities than terminal devices, but also effectively avoid data transmission in the core network. High latency, thereby effectively improving service quality. In intelligent edge computing, limited by the bandwidth of edge devices and other resources, a very critical issue is how to reasonably use the monitored task load for resource scheduling to reduce application latency.

In the process of task scheduling in edge computing, it is not only necessary to monitor the information of task running in real time, but also to consider the data transmission problem in the interaction process with other edge nodes and even other tasks in the data center. Multi-level, multi-data flow, task-dependent resource scheduling problem. Most of the current research work is mainly for the application of data flow single-task processing. When multiple data streams generated by terminal devices distributed under the coverage of multiple edge nodes need to be processed, traditional methods often ignore the communication between tasks. It is often difficult to apply due to the relationship, which brings additional communication overhead and increases the delay.

We abstract the system into the following model:

The system contains several computing nodes, the number of which is M, and the set is $D = \{d_1, d_2, \dots, d_M\}$. These computing nodes include 1 data center and $M-1$ edge computing equipment, let computing equipment d_1 denote data center. Among them, for any edge computing node d_i, it directly covers several terminal intelligent devices. Each device as a data producer will generate a data stream. The data stream will be processed at the edge computing node or forwarded to other computing devices via the edge computing node. Processing is performed, that is, the set of data streams generated by the terminal intelligent devices covered by the edge computing node $d_i$is Q_{d_i}.

In the system, the number of stream data processing applications that need to be deployed is N, that is, the set is $S = \{s_1, s_2, \dots, s_N\}$. For any stream data processing application s_i, there may be multiple data producers, each data producer generates a data stream, and its set isQ_{s_i}; and there is only one data consumer, this article uses the variable x_{ij} to represent the stream data processing application s_i Whether the data consumer of is $d_j \in D$, if yes, then $x_{ij} = 1$, otherwise $x_{ij} = 0$. At the same time, each task can only be deployed on a single node. Obviously, there are the following constraints:

$$x_{ij} \in \{0,1\}; \forall x \in \{1,2,\dots,N\}, j \in \{1,2,\dots,M\} \quad (4\text{-}2)$$

$$\sum_{j=1}^{M} x_{ij} = 1\,; \forall i \in \{1,2,\dots,N\} \quad (4\text{-}3)$$

Since different tasks will compete for bandwidth resources, bandwidth limitations between nodes need to be considered during task deployment. Let P_{ij}^{k} denote that when the application s_i is deployed on the computing node d_j, the computing node d_k is the bandwidth consumed to support the application s_i. Obviously, P_{ij}^{k} can be expressed as:

$$P_{ij}^{k} = \sum_{q \in \{Q_{si} \cap Q_{dk}\}} r_q\,; \quad \forall i \in \{1,2,\dots,N\}, \forall j,k \in \{1,2,\dots,M\}, j \neq k \quad (4\text{-}4)$$

$$P_{ij}^{k} = \sum_{q \in \{Q_{si} - Q_{si} \cap Q_{dk}\}} r_q\,; \quad \forall i \in \{1,2,\dots,N\}, \forall j,k \in \{1,2,\dots,M\}, j = k \quad (4\text{-}5)$$

where: r_q represents the bandwidth consumed by the data stream q, $Q_{si} \cap Q_{dk}$ represents the set of data streams forwarded by the computing device d_k to be processed by the application s_i, $Q_{si} - Q_{si} \cap Q_{dk}$ represents the data stream processed by the application s_i A collection of data streams that are not covered by the computing device d_j. This article assumes that the location of the terminal smart device and the bandwidth resources consumed by the data stream are fixed. Therefore, P_{ij}^{k} appears as a fixed value in the system.

For any computing node d_j, its bandwidth carrying capacity is fixed, denoted as U_j. For any application s_i, if it is deployed on the computing node d_j, the size of the bandwidth resource consumed by the computing node dj is P_{ij}^{j}; if it is deployed on the computing node $d_k (j \neq k)$, it consumes the bandwidth of the computing node d_j. The size of the resource is P_{ik}^{j}. It is required that the bandwidth consumed by each computing node does not exceed the bandwidth that the computing node can provide, so there are the following constraints:

$$\sum_{i=1}^{N} \sum_{k=1}^{M} \left(x_{ik} P_{ik}^{j}\right) \leq U_j; \forall j \in \{1,2,\dots,M\} \quad (4\text{-}6)$$

In order to achieve optimal task deployment, the goal of this paper is to minimize system delay. The system delay can be defined as the sum of the delays of all data flows from the data source node to the task target node. Since the computing node and data source node of different tasks are often different, it is necessary to reasonably plan the location of the task deployment to achieve the optimal System delay.

For any stream data processing application s_i, there are several data streams processed, and each data stream is sent on the terminal smart device, forwarded by the computing node d_j that directly covers the smart device, and transmitted to the computing node d_k where s_i is deployed for processing . Since the transmission delay from the smart device to the edge node is much smaller than the transmission delay between the edge node and the edge node, the delay cost in this paper only considers the transmission delay between the computing nodes. Let l_{ij} denote the transmission delay of the data stream between the computing nodes d_i and d_j. When $i = j$, $l_{ij} = 0$, otherwise, $l_{ij} > 0$ .Since the position between edge nodes is relatively fixed, suppose l_{ij} is a constant. Let l_{ij} denote the sum of the delays of the data streams processed by the application s_i when the data stream processing application s_i is deployed on the computing device d_j, which can be expressed as:

$$L_{ij} = \sum_{k-1}^{M} \sum_{q \in \{Q_{si} \cap Q_{dk}\}} l_{kj}\,; \quad \forall i \in \{1,2,\dots,N\}, \forall j \in \{1,2,\dots,M\} \quad (4\text{-}7)$$

Under any task deployment plan x, record the total delay cost of the system as V, which can be expressed as:

$$V = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij}\, L_{ij} \quad (4\text{-}8)$$

Based on the above definitions, this article considers the following issues: In a computing network with multiple edge computing nodes - single data center, for data stream processing applications, seek a task placement and bandwidth resource allocation plan, so that the conditions of bandwidth resource constraints can be met. The total delay of the system is the smallest. This problem can be formally defined as problem P_1:

$$\begin{aligned} &\min_{X} V = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} L_{ij} \\ &s.t. \\ &C1: x_{ij} \in \{0,1\}; \forall x \in \{1,2,\dots,N\}, j \in \{1,2,\dots,M\} \\ &C2: \sum_{j=1}^{M} x_{ij} = 1; \forall i \in \{1,2,\dots,N\} \\ &C3: \sum_{i=1}^{N} \sum_{k=1}^{M} (x_{ik} P_{ik}^{j}) \leq U_j; \forall j \in \{1,2,\dots,M\} \end{aligned} \quad (4\text{-}9)$$

Among them, the goal of optimization is the total delay cost under the deployment plan x .Constraint $C1$ means that the decision variable $X = \{x_{ij} | i,j \in \{1,2,\dots,N\}$has a value range of binary discrete values $\{0,1\}$, $x_{ij} = 1$ means that the applicationss_i is deployed on the computing node d_j;constraint $C2$ means that any application s_i must and can only be deployed on a certain computing node. The computing node includes 1 data center and $M-1$ edge computing points; constraints $C3$ represents any computing device d_j, and the bandwidth resource used does not exceed its available bandwidth resource.

2) Results

It mainly studies the application deployment of multi-data streams of the Internet of Things in the "terminal device-edge node-data center" network architecture. The goal is to use reasonable application deployment to make relevant stream data processing applications under the condition of satisfying

network resource constraints. The total service delay is the smallest. This article first proves that the problem is non-deterministic polynomial (NP) difficult. In order to achieve efficient task allocation, this paper designs a dynamic resource scheduling scheme FFS+IPFS. The main idea is to first select the applications and edge nodes that occupy the least bandwidth resources for deployment, and get the approximate best Excellent feasible solution, and then, on this basis, iteratively optimize the existing plan through the local search algorithm, so that the optimized deployment plan can be as homogeneous as possible, that is, the tasks with interdependence are deployed in close edge nodes , Thereby reducing the overhead caused by task communication. This paper has conducted a lot of simulation experiments to compare the algorithm with mainstream algorithms such as ODS (Only Data Center), Hash and Closest. The experimental results show that the FFS+IPFS algorithm proposed in this paper can be effectively implemented under different load conditions. Task scheduling, and the delay is more than 23% lower than other algorithms, which verifies the effectiveness of this algorithm.

The main contributions of this paragraph can be summarized as:

(1) The modeling of multi-data flow task scheduling for edge computing scenarios proves that this problem is NP-hard under the condition of network resource constraints.

(2) Taking advantage of the large number of tasks with low resource requirements in edge computing scenarios, a dynamic resource scheduling scheme FFS+IPFS is proposed, which effectively improves the performance of the algorithm through iterative optimization methods.

(3) A simulation experiment was conducted on a real edge computing task data set. The experimental results show that the FFS+IPFS proposed in this paper can effectively achieve task scheduling, and the average delay is reduced by more than 23% compared with other mainstream algorithms.

5. CONCLUSION

In summary, edge computing at this stage has the following advantages:

(1) Low latency: computing power is deployed near the device side, and the device requests real-time response;

(2) Low bandwidth operation: The ability to move work closer to the user or data collection terminal can reduce the impact of site bandwidth limitations. Especially when the edge node service reduces the request to send a large amount of data processing to the hub.

(3) Privacy protection: Local data collection, local analysis, and local processing effectively reduce the chance of data being exposed to public networks and protect data privacy.

The goal of dynamic scheduling is to schedule computing resources on edge devices for application programs to minimize data transmission overhead and maximize application program execution performance. When designing a scheduler, one should consider: whether tasks can be split and schedulable, what strategies should be adopted for scheduling, and which tasks need to be scheduled. Dynamic scheduling needs to find the optimal balance among edge device energy consumption, calculation delay, transmission data volume, bandwidth and other indicators. According to the current work, how to design and implement a dynamic scheduling strategy that effectively reduces the task execution delay of edge devices is a problem that needs to be solved urgently.

6. REFERENCES

[1] 郑纬民,潘毅,施巍松.专题:边缘计算技术及其应用[J].中兴通讯技术,2019,25(03):1.

[2] 肖楷乐. 边缘计算环境中资源分配和优化关键技术研究[D].北京邮电大学,2021.

[3] 李凯. 面向多目标优化的边缘计算资源分配与任务调度策略研究[D].北京工业大学,2020.

[4] 张云飞,高岭,丁彩玲,赵辉,金帅,高全力.边缘计算环境下改进蚁群算法的任务调度算法[J].计算机技术与发展,2021,31(09):86-91.

[5] 查满霞,祝永晋,朱霖,张伯雷,钱柱中.面向实时流数据处理的边缘计算资源调度算法[J].计算机应用,2021,41(S1):142-148.

[6] 施巍松,张星洲,王一帆,张庆阳.边缘计算:现状与展望[J].计算机研究与发展,2019,56(01):69-89.

[7] 吴鸿飞. 基于边缘计算的微服务调度算法研究[D].电子科技大学,2020.

[8] 杨煜坤. 移动边缘计算任务卸载调度优化问题研究[D].电子科技大学,2020.

[9] 施巍松,孙辉,曹杰,张权,刘伟.边缘计算:万物互联时代新型计算模型[J].计算机研究与发展,2017,54(05):907-924.

[10]张强,张宏莉.移动云计算中任务卸载技术的研究进展[J].智能计算机与应用,2016,6(06):1-4+8.

[11]崔勇,宋健,缪葱葱,唐俊.移动云计算研究进展与趋势[J].计算机学报,2017,40(02):273-295.

[12]邓晓衡,关培源,万志文,刘恩陆,罗杰,赵智慧,刘亚军,张洪刚.基于综合信任的边缘计算资源协同研究[J].计算机研究与发展,2018,55(03):449-477.

[13]简琎峰,平靖,张美玉.面向边缘计算的 Storm 边缘节点调度优化方法[J].计算机科学,2020,47(05):277-283.

[14]黄冬艳,付中卫,王波.计算资源受限的移动边缘计算服务器收益优化策略[J].计算机应用,2020,40(03):765-769.

[15]伏舒存,付章杰,邢国稳,刘庆祥,许小龙.移动边缘环境下面向工作流管理的计算迁移方法[J].计算机应用,2019,39(05):1523-1527.

Wang Jieyong, born in 2000. Undergraduate of China University of Mining and Technology. His main interests include edge computing, computer vision, and data analysis.