Resource Allocation and Task Scheduling Scheme in Priority-Based Hierarchical Edge Computing System

1st LIAO Jing Xian , 2nd WU Xue Wen
College of Computer and Information, HoHai University, Nanjing 211100, China
liaojingxian@hhu.edu.cn

***Abstract*—By offloading intensive computing tasks to the edge cloud, mobile edge computing can meet the end-to-end delay requirements of milliseconds. A hierarchical edge computing offloading framework based on emergency priority is proposed in this paper. A resource allocation and task scheduling optimization scheme based on service emergency priority is proposed to minimize the total delay of the system and ensure the minimum delay experienced by high priority services. Then, in order to ensure that the delay of high priority tasks is still minimum under very dense conditions, a dynamic priority task scheduling algorithm (DPTSA) is designed on the fog server. Simulation results show that the proposed system framework and algorithm can reduce the average delay of system tasks, and significantly reduce the delay of high priority tasks.**

Keywords*—*Mobile edge computing, Priority, Delay, Resource allocation, Task scheduling, Optimization

I. INTRODUCTION

With the development of 5G communication and IoT, a large number of mobile devices and their services have led to the explosive growth of mobile data traffic, more and more mobile applications put forward strict requirements for real-time communication and intensive computing [1]. In the traditional IoT network, the data of the terminal devices is usually transmitted to the cloud server for processing. However, the massive amount of IoT device data brings a heavy burden to the cloud server and wireless link, and the system performance will be affected and degraded sharply. [2]. By implementing an MEC server on a cellular base station, it pushes the processing of computing tasks to the edge of the network close to local users, which can provide higher quality of service (QoS) for IoT applications and meet the key end-to-end delay requirements of 5G networks.

The applications with the highest latency requirements in IoT are certain scenarios that are closely related to human safety and health. These scenarios not only need to transmit warning information timely, but also need to return detailed and accurate reports to help users analyze specific situations and make accurate judgments. For the two results returned by the same batch of application data processing, users have two different requirements. The former requires extremely low latency but small computations called a simple task, while the latter requires large computations but delays, it can be tolerated and is called a complex task.

We propose a hierarchical edge computing offloading framework. The first layer are edge nodes that are closer to the user equipment with fewer computing resources, and the second layer is a fog server which is farther from the user and has more computing resources. The edge nodes handle simple tasks with a small amount of computation in order to quickly return warning messages. After processing by the edge node, the tasks are divided into three priority levels according to the degree of urgency, and the executing time of the tasks generated by the user equipment with the highest priority will be significantly reduced.

In recent years, research on edge computing has mainly focused on offloading decision-making and resource management [3-4]. Mobile users can achieve different goals by choosing their own computing offloading strategy [5-6]. In the MEC system with a large number of offloading users, effective resource allocation becomes the key to efficient computing offloading [7]. Tran [8] et al. decomposes the original MINLP problem into a resource allocation problem with fixed task offloading decisions and a task offload problem. Ren et al. [9] studied three models of local compression, edge cloud compression and partial compression offloading. Due to limited and shared resources leading to non-cooperative competition among users [10], many works have adopted game-theoretic solutions [11-12]. Gu et al. [13] used the matching student project allocation game method to provide a distributed solution to the joint resource allocation problem formulated. Most of the above related work studied resource allocation or task scheduling separately, and does not consider the priority of users.

In section II, we propose a priority-based hierarchical edge computing offloading framework. In section III, we study the allocation of wireless resources and computing resources. In Section IV, we design the dynamic priority task scheduling DPTSA algorithm. In section V, the simulation results are shown. Section VI concludes it all.

II. SYSTEM MODEL

A. Hierarchical Edge Computing System

Fig.1 shows the hierarchical edge computing system, which contains a fog server, a set of edge nodes $\mathcal{N} = \{1,2,...,N\}$ .Each edge node is responsible for a group of local equipment denoted by k .The edge node i can serve a group of terminal devices $\mathcal{M}_i$,The task genated from the device k of the edge node i is denoted as $\tau_{i,k}$.

The initial requirements of the task $\tau_{i,k}$ are expressed as tuple $\{C_{i,k}, D_{i,k}, DL_{i,k}\}$,where $C_{i,k}$ represents the number of CPU cycles required for simple computation of the task,

2473-3636/20/$31.00 ©2020 IEEE
DOI 10.1109/DCABES50732.2020.00021

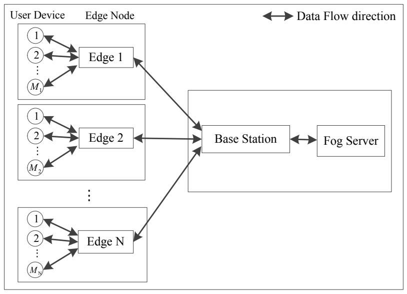

Fig.1 Hierarchical Edge Computing System Architecture

$D_{i,k}$ represents the data size of the task, and $TD_{i,k}$ represents the deadline for completion of the task. After the task is simply processed on the edge node, the requirements tuple becomes $\{C'_{i,k}, D'_{i,k}, DL_{i,k}, p, \alpha_{i,k}\}$. $C'_{i,k}$ is the number of CPU cycles required for the task to perform accurate computations on the fog server, $D'_{i,k}$ is the amount of data remaining after simply processing, p represents three priority levels distinguished according to the degree of urgency, namely $p=1$, $p=2$ and $p=3$. $\alpha_{i,k}$ is the proportion of radio resources allocated to the task for the base station.

B. Communication Model

The base station orthogonally allocates spectrum resources to the tasks of all user equipment in the system, and the total spectrum bandwidth is B .The uplink transmission rate $R_{i,k}$ of the task $\tau_{i,k}$ is denoted as $R_{i,k} = [\alpha_{i,k}(1-p_{i,k})]B\log_2(1+\frac{|h_{i,B}|^2 P_i d_{i,B}^{-r}}{\sigma^2})$,where P_i is the transmit power of the edge node, $h_{i,B}$ represents the uplink channel fading coefficient, $d_{i,B}$ is the distance between the base station and the edge node i , r is the path loss index, and σ^2 is the noise power. $\alpha_{i,k}$ indicates the proportion of uplink spectrum resources allocated to edge nodes, $\alpha_{i,k} \in [0,1]$ and $\sum_{i=1}^{N}\sum_{k=1}^{M_i}\alpha_{i,k} \le 1$, $p_{i,k}$ is the priority weight. The remaining amount of data of task $\tau_{i,k}$ processed by the edge node is denoted as $D'_{i,k}$, $D'_{i,k} = x_{i,k} \cdot D_{i,k}$, $x_{i,k}$ is the proportion of data after processing. The communication delay of the task is $t_{i,k}^{tran} = \frac{D'_{i,k}}{R_{i,k}}$.

C. Computation Model

Computing resources of all edge nodes connected to each fog server are equal, denoted as f_n .The computing capacity allocated by the edge node to the task on the device k is recorded as f_k^{edge} .The sum of computing capacity allocated to all devices by each edge node should not exceed the maximum computing resources of the edge node, namely, $\sum_{k=1}^{M_i} f_k^{edge} \le f_n$ .Denote f_s as the maximum computing capacity of the fog server, $f_{i,k}^{fog}$ indicates the computing capacity assigned by the fog server to the task from the device k on the edge node i .The sum of computing capacity allocated to all tasks should be less than the computing power of the fog server, namely, $\sum_{i=1}^{N}\sum_{k=1}^{M_i} f_{i,k}^{fog} \le f_s$.

The task performs more complex computations on the fog server, so the amount of computation is larger, and the number of CPU cycles required is more, denoted as $C'_{i,k} = y_{i,k} \cdot C_{i,k}$, fog computation delay is $t_{i,k}^{fc} = \frac{C'_{i,k}}{f_{i,k}^{fog}}$ .Then the computation delay is expressed as the sum of edge computation and fog computation delay, namely, $t_{i,k}^{comp} = \frac{C_{i,k}}{f_k^{edge}} + \frac{C'_{i,k}}{f_{i,k}^{fog}}$.

III. Resource Allocation Optimization

In the section, we will formulate the delay minimization resource allocation optimization problem, which is formulated as follows

$$\min_{\{\alpha_{i,k}, f_k^{edge}, f_{i,k}^{fog}\}} \sum_{i=1}^{N}\sum_{k=1}^{M_i}\left(\frac{D'_{i,k}}{R_{i,k}} + \frac{C_{i,k}}{f_k^{edge}} + \frac{C'_{i,k}}{f_i^{fog}}\right) \tag{1}$$

$$\begin{aligned} s.t.\quad & C1: \sum_{i=1}^{N}\sum_{k=1}^{M_i}\alpha_{i,k} \le 1, \alpha_{i,k} \ge 0, k \in \mathcal{M}_i, i \in \mathcal{N} \\ & C2: \sum_{k=1}^{M_i} f_k^{edge} \le f_n, f_k^{edge} \ge 0, k \in \mathcal{M}_i \\ & C3: \sum_{i=1}^{N}\sum_{k=1}^{M_i} f_{i,k}^{fog} \le f_s, f_{i,k}^{fog} \ge 0, k \in \mathcal{M}_i, i \in \mathcal{N} \end{aligned}$$

where C1 is the optimization constraint of the wireless resource allocation ratio, which guarantees that the sum of the wireless resources allocated to all devices will not exceed the maximum bandwidth of the system, and ensures that the wireless resource allocation ratio is non-negative.C2 and C3 are the optimization constraints for edge computing and fog computing, respectively. They ensure that the sum of computing resources allocated to all devices will not exceed the maximum computing capacity.

In this paper, there is no direct connection between wireless resource and computing resource allocation, so the joint resource allocation optimization problem is divided into two sub-problems, namely, radio resource allocation (RRA) and computing resource allocation (CRA).

The RRA problem is a convex optimization problem. In order to solve this convex problem, we let $r_{i,k} = B \cdot \log_2(1+\frac{|h_{i,B}|^2 P_i d_{i,B}^{-r}}{\sigma^2})$, a Lagrange function is defined as,

$$L(\alpha_{i,k}, \lambda) = \sum_{i=1}^{N}\sum_{k=1}^{M_i}\frac{D'_{i,k}}{\alpha_{i,k}(1-p_{i,k})r_{i,k}} + \lambda\left(\sum_{i=1}^{N}\sum_{k=1}^{M_i}\alpha_{i,k} - 1\right) \tag{2}$$

Using the KKT condition, the optimal solution for the wireless resource allocation ratio is

$$\alpha_{i,k}^* = \frac{\sqrt{\frac{D'_{i,k}}{(1-p_{i,k})r_{i,k}}}}{\sum_{i=1}^{N}\sum_{k=1}^{M_i}\sqrt{\frac{D'_{i,k}}{(1-p_{i,k})r_{i,k}}}}, k \in \mathcal{M}_i, i \in \mathcal{N} \tag{3}$$

Similarly, we define a Lagrange function for CRA problem, which is expressed as

$$L(f_k, f_{i,k}, \lambda, \mu) = \sum_{k=1}^{M_i} \left(\frac{C_{i,k}}{f_k} + \sum_{i=1}^{N} \frac{C'_{i,k}}{f_{i,k}}\right) + \lambda\left(\sum_{k=1}^{M_i} f_k - f_n\right) + \mu\left(\sum_{i=1}^{N}\sum_{k=1}^{M_i} f_{i,k} - f_s\right) \tag{4}$$

Using the KKT condition, the optimal solution for the computation resource allocation on edge nodes is

$$f_k^* = \frac{f_n \sqrt{C_{i,k}}}{\sum_{k=1}^{M_i} \sqrt{C_{i,k}}}, k \in \mathcal{M}_i \tag{5}$$

and the optimal solution for the computation resource allocation on fog server is

$$f_{i,k}^* = \frac{f_s \sqrt{C'_{i,k}}}{\sum_{i=1}^{N}\sum_{k=1}^{M_i} \sqrt{C'_{i,k}}}, k \in \mathcal{M}_i, i \in \mathcal{N} \tag{6}$$

IV. Dynamic Priority Task Scheduling Algorithm

We propose a task scheduling algorithm based on dynamic priority (DPTSA). The algorithm considers both the urgency of the task and the deadline of the task to ensure that the tasks with higher priority run first, which improves the scheduling performance.

The DPTSA algorithm is shown in Algorithm 1.Firstly, we set up three buffers, namely high priority class1, medium priority class2 and low priority class3, and the task enters the buffer of the corresponding priority; then execute tasks in class1, class2 and class3 sequentially, and each buffer is scheduled separately; The emergency degree indicator is set for the algorithm, which combines the waiting time of the task and the task execution time. As the waiting time increases, the urgency of all tasks changes dynamically. Denote the urgency as e , the waiting time of the task as x , and the remaining execution time of the task as, and the urgency of the task is expressed as:

$$e = \begin{cases} \dfrac{w_1 \cdot |x| + 1}{y}, x < 0 \\ \dfrac{1}{w_2 x + y}, x \geq 0 \end{cases} \tag{7}$$

Algorithm 1: DPTSA

```
Initialize: create 3 buffers called class1, class2 and class3 respectively
if a new task arrives:
    if p == 1:
        place task in class1;
    if p == 2:
        place task in class2;
    else:
        place task in class3;
if class1 is not empty:
    while True:
        update x, y of every task;
        calculating e of all the tasks in the buffer;
        pick the task with highest e in the buffer;
        calculate time piece with age;
        execute the task in one time piece;
        if task is not completed:
            update y, age;
            place the task back in the buffer;
        if a task is executed completely:
            remove the task from the buffer
        if the buffer is empty:
            break;
if class2 is not empty:
    executing the same scheduling process as class1;
if class3 is not empty:
    executing the same scheduling process as class1;
```

V. Simulation Results

The default settings of some simulation parameters are as follows: The system is configured with 1 fog server and 10 edge nodes. The number of user equipment managed by each edge node is 10~20. For wireless access, the spectrum bandwidth of uplink and downlink are equal, we set the bandwidth B = 200 MHz. The Edge node transmission power P_i = 27 dBm, The channel fading coefficient of MDs follow the exponential distribution with mean 1. The path loss factor r = 4. The computing capacity of edge nodes is 60000 Mega cycles and the computing capacity of fog server is 120000 Mega cycles. The computational load of the task follows a Gaussian distribution $CN(\mu, \sigma_1^2)$, $\mu = 500$, Mega cycles， $\sigma_1^2 = 100$.

A. Comparison of different resource allocation strategies

We compared the average delay of system tasks under different resource allocation strategies, which are the following four strategies: (1)Joint wireless and computing resource allocation optimization strategy; (2) Only computing resource allocation is optimized, and wireless resources are evenly allocated; (3) Only wireless resource allocation is optimized, and computing resources are evenly allocated; (4) No resource allocation optimization, Evenly allocate wireless and computing resources.

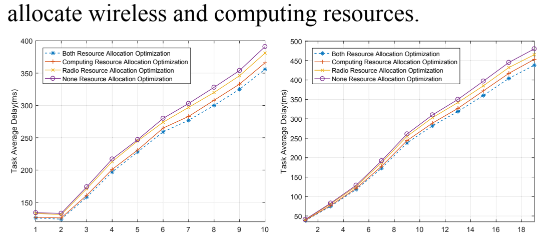

Fig.2 Task average delay of Four Resource Allocation Strategy

Fig.2 reflects the relationship between system delay and the total number of tasks. As the number of user equipment increases, the delays of all strategies are increasing due to limited resources. The joint wireless and computing resource allocation optimization strategy proposed has the smallest delay, and the non-resource allocation optimization strategy has the largest delay. The delay of only the computation delay allocation optimization strategy is smaller than that of only the wireless resource allocation optimization strategy. This is because in this system, the transmission delay caused by wireless communication is inherently smaller than the computation delay, so the optimization effect on the allocation of computing resources is more obvious.

B. Performance of DPTSA

In order to verify the performance of the DPTSA, we compare it with the first-in first-out (FIFO) algorithm and the pure priority scheduling algorithm (Priority). In order to evaluate the performance of the algorithm fairly, a Delay Degradation indicator is set, which is composed of two parts: the overall task timeout index and the average delay. The sum of the overtime indices of all tasks in the system plus the average task delay is the delay deterioration index.

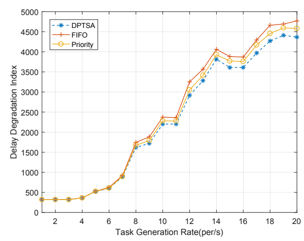

Fig.3 Delay Degradation with task generation rate

As the task generation rate gradually increases, the task timeout index and average task delay of the DPTSA algorithm are the lowest. The delay deterioration index of the Priority algorithm is lower than that of the FIFO algorithm because although the average delay of the two tasks is similar, the high priority task timeout index of the Priority algorithm is lower than that of the FIFO algorithm. The delay deterioration index of the DPTSA algorithm is lower than that of the Priority algorithm because although both have different priorities, in their respective priority buffers, the Priority algorithm is still first in, first out, and the DPTSA algorithm is based on the urgency of the task. To execute, the average task delay is lower than Priority algorithm. In the case of high task density, the performance of the DPTSA algorithm is better than other algorithms.

C. Performance comparison of three priority tasks

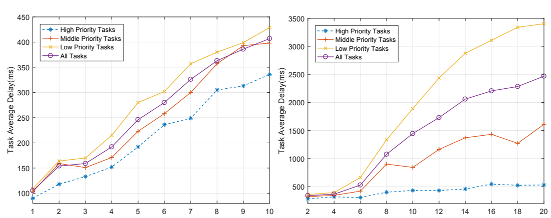

Fig.4 Task average delay of different priorities

Since we set up priority-based wireless resource allocation, the average delay of high-priority tasks is the lowest. The low latency of high-priority tasks is at the expense of the performance of low-priority tasks, because low-priority tasks are always greater than the overall task average delay. With the increasement of task generation rate, the number of queues in the system buffer increases, and the delay gap between the three priority tasks is getting bigger and bigger. This is because the DPTSA algorithm proposed in this paper plays a vital role.

VI. Conclusion

In this paper, a priority-based hierarchical edge computing offloading system framework is proposed for user scenarios where the situation is more urgent, and the resource allocation and task scheduling schemes based on the system are studied. In order to minimize the system delay, firstly, the allocation of wireless resources and computing resources is constructed and then their optimal solutions are obtained by using KKT conditions. We design the DPTSA algorithm, and gives the task scheduling algorithm and execution process of the entire system. Finally, the simulation results show that the resource allocation scheme in this paper can effectively reduce the task delay. The delay of high-priority tasks is lower than that of other priority tasks, and the DPTSA algorithm also has better performance than other scheduling algorithms.

REFERENCES

[1] Tao, Xiaoyi, et al. "Performance guaranteed computation offloading for mobile-edge cloud computing." IEEE Wireless Communications Letters 6.6 (2017): 774-777.

[2] Zhang, Guowei, et al. "Delay minimized task scheduling in fog-enabled IoT networks." 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2018.

[3] Mao, Yuyi, et al. "A survey on mobile edge computing: The communication perspective." IEEE Communications Surveys & Tutorials 19.4 (2017): 2322-2358.

[4] Mach, Pavel, and Zdenek Becvar. "Mobile edge computing: A survey on architecture and computation offloading." IEEE Communications Surveys & Tutorials 19.3 (2017): 1628-1656.

[5] Zhang, Huaqing, et al. "A hierarchical game framework for resource management in fog computing." IEEE Communications Magazine 55.8 (2017): 52-57.

[6] Tang, Ling, and Shibo He. "Multi-user computation offloading in mobile edge computing: A behavioral perspective." IEEE Network 32.1 (2018): 48-53.

[7] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," IEEE Trans. Comput., vol. 64, no. 8, pp. 2253–2266, Aug. 2015.

[8] Tran, Tuyen X., and Dario Pompili. "Joint task offloading and resource allocation for multi-server mobile-edge computing networks." IEEE Transactions on Vehicular Technology 68.1 (2018): 856-868.

[9] Ren, Jinke, et al. "Latency optimization for resource allocation in mobile-edge computation offloading." IEEE Transactions on Wireless Communications 17.8 (2018): 5506-5519.

[10] Li, Keqin. "A game theoretic approach to computation offloading strategy optimization for non-cooperative users in mobile edge computing." IEEE Transactions on Sustainable Computing (2018).

[11] Zheng, Jianchao, et al. "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach."

[12] Chen, Xu. "Decentralized computation offloading game for mobile cloud computing." IEEE Transactions on Parallel and Distributed Systems 26.4 (2014): 974-983.

[13] Gu, Yunan, et al. "Joint radio and computational resource allocation in IoT fog computing." IEEE Transactions on Vehicular Technology 67.8 (2018): 7475-7484