Computer Communications 193 (2022) 355–364

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

EC-MASS: Towards an efficient edge computing-based multi-video scheduling system☆

Shu Yang [a], Qingzhen Dong [a], Laizhong Cui [a,e,*], Xun Chen [b], Siyu Lei [c], Yulei Wu [d], Chengwen Luo [a]

[a] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[b] School of Communication, Shenzhen Polytechnic, Shenzhen 518055, China
[c] College of Computer Science, Sichuan University, Chengdu 610207, China
[d] College of Engineering, Mathematics and Physical Science, University of Exeter, United Kingdom
[e] Peng Cheng Laboratory, Shenzhen 518060, China

ARTICLE INFO

Keywords:
Edge computing
Video analytics
Task scheduling

ABSTRACT

Video cameras have been deployed widely today. Although existing systems aim to optimize live video analytics from a variety of perspectives, they are agnostic to the workload dynamics in real-world. We propose EC-MASS, an edge computing-based video scheduling system achieving both cost and performance optimization with multiple cameras and edge data centers. The intuition behind EC-MASS is to adaptively map cameras to different edge data centers according to dynamically updated configurations of cameras. We prove that generating the optimal mapping scheduling scheme is NP-Complete, and develop the scheduling algorithm by leveraging the insights of the economy consideration of camera allocation. Using the algorithm, EC-MASS is able to balance the workload among edge data centers while reducing the cost of video analytics system. We evaluate EC-MASS with datasets of video configurations from real-world cameras which randomly generate configurations for cameras, with a testbed that consists of 60 cameras and 4 edge data centers. Our results show that EC-MASS consistently outperforms the status quo in terms of cost and performance stability.

1. Introduction

Recent years have seen a rapid increase in the deployment of video cameras, which has been at places such as traffic intersections, factories, and supermarkets [1–3]. Analyzing live videos streamed from these distributed cameras is the backbone of a wide range of applications such as traffic control and security surveillance. As many of these applications require producing analytics results in real-time, achieving low-latency, high throughput, and scalable video stream processing is crucial [4].

A typical video analytics application consists of a pipeline of video processing modules. For example, the pipeline of a traffic application that counts vehicles consists of a decoder, followed by a component to re-size and sample frames, and an object detector. The pipeline has several "knobs" such as frame resolution, frame sampling rate, and detector model (e.g., Yolo [5], VGG [6] or AlexNet [7]). We refer to a particular combination of knob values as a configuration. Since the values of configurations are critical to the performance (accuracy and resource consumption) of object detection, we should optimize configuration in different scenarios. There are many video analytics technologies that focus on the configuration optimization.

- **One-time update:**VideoStorm [8], NoScope [9], MCDNN [10] and Focus [11] optimize video processing pipes by adjusting video analysis configuration knobs or training specialized neural network models. However, the video query is analyzed and optimized only once at the beginning of the video stream.
- **Periodic update:** Chameleon [2] proves that the best configuration changes over time, taking advantage of the time persistence of the configuration, the spatial similarity, and the independence of the configuration knob to update the optimal configuration periodically.

☆ This work was supported in part by the National Key Research and Development Plan of China under Grants No. 2018YFB1800805, National Natural Science Foundation of China under Grants 61772345, 61902258, Major Fundamental Research Project in the Science and Technology Plan of Shenzhen, China under Grants JCYJ20190808142207420, GJHZ20190822095416463, RCYX20200714114645048, Natural Science Foundation of Guangdong Basic and Applied Basic Research, China under Grants 2021A1515011857, and the Pearl River Young Scholars funding of Shenzhen University, China.

* Corresponding author at: College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China.
E-mail addresses: yang.shu@szu.edu.cn (S. Yang), 13476009347@163.com (Q. Dong), cuilz@szu.edu.cn (L. Cui), chenxun@szpt.edu.cn (X. Chen), 3232297508@qq.com (S. Lei), Y.L.Wu@exeter.ac.uk (Y. Wu), chengwen@szu.edu.cn (C. Luo).

https://doi.org/10.1016/j.comcom.2022.07.002
Received 14 February 2022; Received in revised form 19 May 2022; Accepted 4 July 2022
Available online 14 July 2022

Real-time video analytics systems need high-resolution cameras to capture high-quality visual data for analysis. Due to the lack of network bandwidth and the long transmission delay between the camera and the cloud, massive data cannot be transmitted to the cloud data center for real-time processing. The key to overcoming this bottleneck is to move computing resources near where the data resides. The current live video analytics system transmits the camera to the local edge cluster for data analysis, which is called edge computing-based video analytics.

Motivation for Mapping Scheduling Adaptation: To achieve the "best" resource–accuracy tradeoff, we adopt configuration adaptation in video analytics system, which will result dynamics of workload in edge data centers. At present, the mapping connections between the cameras and the edge data centers in the real-world video analytics system are usually static or fixed. While video configurations of cameras vary over time, the workload of the required bandwidth resources and computing resources consumption may exceed the resource limits of the edge data centers, and the processing delay of video analytics tasks will increase and the system performance will decline, as shown in Fig. 1.

However, blindly upgrading edge data centers to increase their processing capability will lead to the increase of economic cost and the low utilization rate of resources sometimes. So we need to adopt mapping scheduling adaptation between cameras and edge data centers while the configurations of cameras vary over time.

The design of mapping scheduling involves two considerations.

- **Consideration #1: Bandwidth and Computing Price.** Because different edge data centers are located in different geographical locations, different network operators who provide network services for them have different charging standards. Therefore, the reasonable mapping scheduling between multi-cameras and multi-edge servers can reduce the overall bandwidth cost. Besides, the price of computing resource is different among edge data centers. So, we should take both bandwidth and computing price as our economic consideration.
- **Consideration #2: Performance Stability of Edge Server.** If only the greedy algorithm is used to map the cameras to the edge servers with the lowest network bandwidth charge, the computing resource load will reach a congested state. At this time, because the video analytics system periodically updates and adjusts the configurations of the camera, it will increase the demand for computing resources of the camera, which exceeds the constraint of the computing resources of the edge server. As a result, the edge server cannot achieve normal video analytics tasks, resulting in high delay, large error and other unwanted drawbacks.

To address the above two considerations, we design EC-MASS, a video analytics system, which can not only reduce the system cost, but also ensure the stability of the system performance. We define the objective equation which takes both performance and economy as metrics based on the mathematical model of EC-MASS. Unfortunately, the objective equation determines that the problem of mapping scheduling between the overall cameras and the edge data centers is NP-Complete. That is, we cannot retrieve the optimal mapping scheduling scheme in linear time.

To address the NP-Completeness of generating the optimal mapping scheduling scheme, we apply the augmented Lagrangian to obtain the ideal values of workload assigned to each edge data center, as a guiding scheme for us to design mapping scheduling algorithm. However, there may be multiple mapping scheduling schemes that satisfy the guiding scheme. To ensure the uniqueness and rationality of the mapping scheduling scheme, we design the mapping scheduling equations that leverage insights on the economy consideration of camera allocation.

Economy consideration: Although the configuration of cameras varies over time, it cannot exceed the golden configuration, which is the most expensive configuration. Therefore, there is an upper limit of the resource consumption of cameras. During the interval between two mapping scheduling processes, we can find that the increase of resource consumption may be higher for the cameras with lower resource consumption. It can result in higher operation cost if we allocate these cameras to the edge data centers with higher resource price. Thus we can relatively reduce the operation cost of system if cameras with lower resource consumption are allocated to edge data centers with lower resource price.

In this paper, we leverage the mapping scheduling equations to design a mapping scheduling algorithm adaptively optimizing both economy and performance of the video analytics system when the configurations of cameras are dynamically updated. Then based on the algorithm, we develop EC-MASS, an edge computing-base video analytics system achieving above optimization goal in a scenario with multiple cameras and edge data centers. Using dynamically updated configuration set, we show that compared to a baseline system mapping connection to the nearest between cameras and edge data centers once upfront, EC-MASS can achieve better performance (theoretically does not exceed the system resource limit, while the former occurs three times in 1.5 h), and lower cost (about 10.9%).

Our key contributions are as follows:

- We analyze the impacts of configuration dynamics for video analytics system based on the multi-cameras and multi-edge edge data centers architecture, and show that it can improve both economy and performance of the system if we adapt mapping scheduling according to the updated configurations. 2.2
- We design a mapping scheduling algorithm to adaptively generate the mapping scheme that satisfies the objective equation. 4
- We present a video analytics system, EC-MASS, with reduced operation cost and improved performance stability by leveraging the mapping scheduling algorithm. 3

2. Background and motivation

We begin with some background on live video analytics pipeline, and show the need for mapping scheduling adaptation when the video configurations of cameras change dynamically.

2.1. Live video analytics pipelines

The typical live video analytics pipelines consist of a front-end object detector and a back-end task-specific module. Object detection is a core vision task performed by live video analytics pipelines to identify objects of interests, their classes, speed, and locations within each video frame.

The workflow of object detection pipelines is that they first preprocess raw video frames by sampling and resizing frames or detecting regions with objects of interest by a light-weight background subtractor, then send the frames into pre-trained object-detection models or NN-based classifiers.

These object detection pipelines have their own sets of configuration knobs. For some pipelines, their configuration knobs consist of frame rate, image size, and object detection model. Configuration knobs of other pipelines include minimum size of the extraction region (with detected objects of interest as a fraction of the whole frame), and the classifier model. For simplicity, we call the combination of configuration knobs as configuration.

Since the values of configurations are critical to the performance (accuracy and resource consumption) of object detection, many video analytics systems, such as VideoStorm [8], NoScope [9], MCDNN [10] and Focus [11] optimize video configurations at the beginning of the video stream. As videos and the characteristics of video analytics pipelines exhibit substantial dynamics over time, we cannot achieve the "best" resource–accuracy tradeoff, if the configurations of the video pipelines are preset and fixed. Therefore, we need to continuously adapt the configurations. Chameleon [2] leverages insights on the temporal

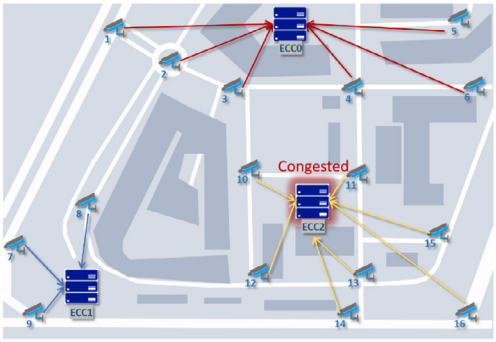


Fig. 1. System will be congested if mapping connections are static.

correlation, cross-camera correlation, and independence of configurations to optimize resource consumption and inference accuracy of video analytics pipelines by adapting their configurations in real-time.

For cameras deployed in large areas, such as campuses and communities, they are connected to the adjacent edge data centers to perform a variety of analytics tasks. When we adapt the video configuration in the video analytics system, the amount of the video stream uploaded by the cameras to the edge data centers is also changing over time. Therefore, we need to consider the problem of mapping scheduling between the cameras and the edge data centers when the video configurations change dynamically.

2.2. Need for mapping scheduling adaptation

To illustrate the impacts of dynamic changes in the video configurations of the cameras in the video analytics system, we show a campus map with 16 cameras and 3 edge data centers. At the end of each video configuration profiling phase, the $top-k$ configurations shared within each camera cluster are updated, resulting in variations in the amount of video stream data uploaded by cameras to the connected edge data centers. Then, the bandwidth resources and computing resources required by the video analytics tasks of cameras also vary over time. So we can observe the impacts as follows:

- There are upper limits of bandwidth resources and computing resources in real-world edge data centers. If the mapping connection between the cameras and the edge data centers remains static, congestion may occur in the edge data center (such as the edge data center ECC2 in Fig. 1) when the resource consumption of the camera connected to the edge data center exceeds its resource limit. Consequently, when the workload of edge data centers exceeds their processing capability, the video analytics tasks of cameras fail to be processed in time (i.e., the decrease in performance)
- There are differences in the equipment of each edge data centers and the charging standards of the network services providers, leading to the difference in their unit bandwidth resources and computing resources price. If the edge data centers with higher price have more workload than that of the edge data centers with lower price, the overall operating cost of the video analytics system will also be higher (i.e., the drawback of economy).

Existing live video analytics systems based on the cameras-edge data centers architecture are agnostic to the impacts of dynamics of the video configurations illustrated above. However, the mapping relationship between the cameras and the edge data centers remaining static during the video configurations variation will reduce the performance and economy of the system. These impacts altogether highlight the need for mapping scheduling between cameras and edge data centers to avoid the occurrence of either the decrease in performance or the drawback of economy. And this is the motivation of the design of EC-MASS.

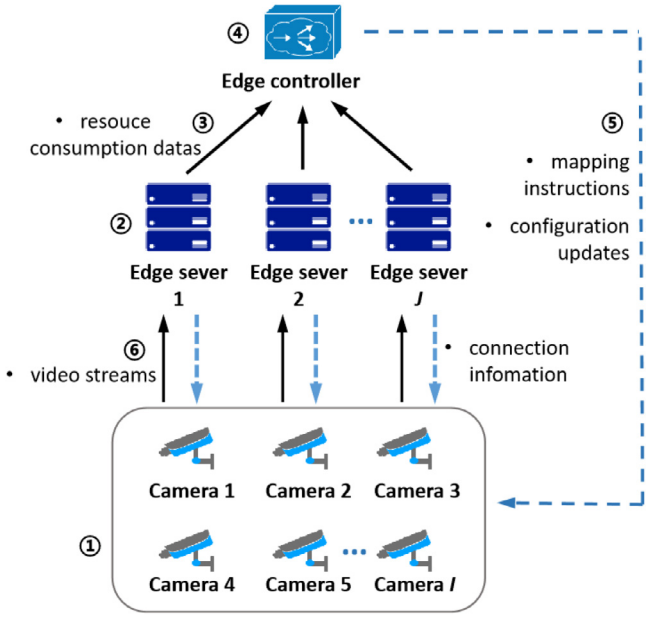


Fig. 2. System architecture of EC-MASS.

3. EC-MASS design

3.1. Overall architecture

Fig. 2 illustrates the overall architecture of EC-MASS. The system consists of EC-Controller, Edge Data Centers and Camera Clusters. The functions of the components of the system are as follows:

Camera clusters:

- The cameras in the system are divided into clusters according to the spatial similarity of the video configuration.
- Upload the video stream data from the main camera in each camera cluster to the edge data center to obtain the best video configuration and share it among the clusters.
- Upload the video stream data to the edge data center of the mapped connection according to the obtained video configuration.

Edge data centers:

- Periodically profile the optimal video configuration of the camera clusters, and send the configuration information to the EC-Controller.
- Receive the video stream data from the cameras and perform the corresponding video processing functions, such as target recognition, retrieval, tracking.

EC-Controller:

- Collects and updates the global camera clusters configuration information dynamically, and executes the scheduling algorithm accordingly, then generates the mapping scheduling scheme between multi-edge data centers and multi-cameras in real time.
- Selects the appropriate scheduling time, according to the economic and performance requirements of the system.
- Sends the configuration information and mapping scheduling scheme to the corresponding camera clusters to achieve real-time mapping scheduling.

As shown in Fig. 2, the running process of the system is mainly divided into the following steps: First, according to the spatial similarity of video configuration, We divide I cameras into Q camera clusters

Table 1
Variable list.

Variables	Meanings
CAM	The collection of cameras
I	The number of cameras
CCL	The collection of camera clusters
Q	The number of camera clusters
ECC	The collection of edge data centers
J	The number of edge data centers
C	The video configurations of cameras
s_i	Computing resource consumption of the camera i
b_i	Bandwidth resource consumption of the camera i
μ	The correlation conversion coefficient between s_i and b_i
A	The total amount of computing resources consumed
B_j	Bandwidth resource load of the edge data center j
SL_j	Computing resource limits of the edge data center j
BL_j	Bandwidth resource limits of the edge data center j
mc_j	Unit computing resource price of the edge data center j
mb_j	Unit bandwidth resource price of the edge data center j
m_j	Comprehensive resource price of the edge data center j
γ	The weight coefficient between the system cost and performance
M	The total resource cost of all edge data centers
$\sigma(S)$	The indicator of system performance
λ	The Lagrange multiplier

(①). According to the Principle of Nearest First (PNF), each camera is mapped and connected to the corresponding edge data center. In each camera cluster, we select a camera as the Main Camera (with better viewing angle and illumination conditions). Then, the edge data center periodically profiles the video configuration (②) of the connected main camera to obtain the optimal configuration, and uploads the updated configuration information to EC-Controller (③). By collecting the global camera configuration along with the preset system resource information, EC-Controller executes the mapping scheduling algorithm to generate a real-time mapping scheduling scheme (④) between multi-edge data centers and multi-cameras. Then, EC-Controller sends the optimal configuration and mapping information to the corresponding camera cluster (⑤) to realize real-time mapping scheduling. Finally, according to the dynamically updated video configuration and mapping information, the camera sends the video stream data to the mapped edge data center (⑥) to perform the video processing functions in different scenarios. The system periodically implements the above processes in order to achieve the improvement of the economy and performance of the system.

3.2. Mathematical model of mapping scheduling

In our system: EC-MASS, we schedule the real-time mapping between the cameras and the edge data centers according to the dynamic video configuration of each camera, so as to reduce the operating cost and improve the performance of the video analytics system. To achieve this goal, we present the mathematical model of system for mapping scheduling. In the following, we describe the mathematical conditions of EC-MASS, mapping status function, system metrics and objective equation.

In EC-MASS, the number of cameras is I, and we divide them into Q camera clusters, and the number of edge data centers is J. Before the operation of the system, a controller (EC-Controller) records the upper limit and unit cost of the bandwidth and computing resources of each edge data center in the system. In addition, EC-Controller also records the consumption of the bandwidth and computing resources of cameras in the system, which vary in real time but can be calculated from the video configuration of the camera.

3.2.1. Mathematical conditions

To model EC-MASS and formulate the objective equation, we list the initial mathematical variables involved and their corresponding meanings in Table 1.

3.2.2. Mapping status function and system metrics

To generate the mapping scheduling scheme of the video analytics system, we leverage the following mapping status equation to describe the mapping relationship between the cameras and the edge data centers. Then, utilizing this equation, we get the metrics to measure the cost and performance of the video analytics system as follows:

- To describe the overall connection status between cameras and edge data centers, we set mapping scheduling function as

$$F: f(i,j) = \begin{cases} 1 & \text{camera } i \text{ mapped to edge server } j \\ 0 & else. \end{cases}$$

- Since computing resource consumption s_i of the camera i depends on the amount of video stream data sent to the edge data center, which is reflected by its bandwidth resource consumption b_i, we can define the relationship between computing resource consumption and bandwidth resource consumption of the camera i as $s_i = \mu b_i$, where μ is the correlation conversion coefficient.
- We define the total amount of computing resources consumed by all cameras as $A = \sum_i s_i = \mu \sum_i b_i$. It is calculated by the EC-Controller at the beginning of mapping scheduling.
- For the edge sever j with cameras mapped to it, its bandwidth resource load is $B_j = \sum_i b_i f(i,j)$, computing resource load is $S_j = \sum_i s_i f(i,j) = \mu \sum_i b_i f(i,j) = \mu B_j$
- Define the total bandwidth and computing resource cost of all edge data centers as $M = \sum_j B_j m_j$, where $m_j = mb_j + \mu \cdot mc_j$, which is called the unit bandwidth converted price.
- We use the variance of the computing resource workload to represent the difference of the load distribution, as an indicator of system performance: $\sigma(S) = \frac{1}{J}\sum_j (S_j - \frac{1}{J}A)^2$

3.2.3. Objective equation

The goal of our designed video analytics system is to reduce the operating cost and improve the performance of system by adaptively scheduling the mapping relationship between the cameras and the edge data centers. So we define the objective equation of EC-MASS as the sum of M and $\sigma(S)$. To minimize the result of the following objective equation, we should find the optimal mapping scheduling function $F: f(i,j)$.

$$\min_{f(i,j)} [M + \gamma\sigma(S)]$$
$$s.t. \quad S_j \leqslant SL_j$$
$$B_j \leqslant BL_j, \quad j = 1, 2, 3 \ldots, J$$

where γ is the weight coefficient between the system cost and performance, called as Cost to Performance Balancer (CPB). CPB γ contributes to the balance between the system cost and performance according to the actual needs of users in different scenarios. We will discuss the effects of CPB in 5.5.

4. Details of mapping scheduling algorithm design

At the beginning of each mapping scheduling cycle, EC-Controller collects the global resource allocation information of the system, and periodically calculates the mapping scheduling scheme between the cameras and the edge data centers according to the objective equation. The reasonable mapping scheduling algorithm is the core of EC-MASS to generate mapping scheduling scheme adaptively.

According to the mathematical modeling of EC-MASS 3.2, we design a mapping scheduling algorithm that satisfies the objective equation. First, according to the objective equation, the mapping scheduling between the overall cameras and the edge data centers is NP-Complete. Therefore, the algorithm to retrieve the optimal mapping scheduling scheme cannot be completed in linear time 4.1. Second, for minimizing the objective equation, we introduce a method to solve the problem and the conditions to be satisfied by the mapping scheduling scheme

between the cameras and the edge data centers 4.2. Third, according to the mapping scheduling scheme, we introduce the key insight of designing the mapping scheduling algorithm 4.3. Finally, we describe the pseudo code of the designed algorithm and the corresponding analysis 4.4.

4.1. Proof of NP-completeness of overall mapping scheduling

For multi-cameras and multi-edge servers mapping scheduling, there is an optimal mapping scheduling scheme theoretically: each camera can be scheduled to the most appropriate edge server, so that the video analytic task has the lowest execution cost and meets the performance requirements. The optimal mapping scheduling algorithm needs to consider all possible mapping scheduling schemes and choose the one with the lowest computing delay. As long as the computing time is long enough, the optimal mapping scheduling strategy can be found. However, Theorem 4.1 will prove that the optimal mapping scheduling problem is a NPC problem, so it cannot be solved in polynomial time.

Theorem 4.1. *The optimal mapping scheduling problem is a NP complete problem.*

It is easy to verify a given mapping scheduling scheme in polynomial time, so the optimal mapping scheduling problem belongs to the NP class problem. Theorem 4.1 can be proved by reduction from the cabinet-packing problem which has been proved to be a NP complete problem [12]. The cabinet-packing problem is that the given input is the numerical sequence $a_1, a_2, \dots, a_n$ and the values b and k, determine whether the set can be divided into k subsets and the sum of values of each subset is less than or equal to b.

In order to prove that the optimal mapping scheduling(OMS) problem is NP difficult, we first prove that $Cabinet-packing \leqslant_p OMS$. Let the numerical sequence $a_1, a_2, \dots, a_n$ and the values b and k are an instance of Cabinet-packing. An instance of OMS can be constructed as follows: $w_i = 1$, $J = k$, $\gamma = 0$, numerical sequence $a_1, a_2, \dots, a_n$ and the values $b = \sum a_i$ and $k = J$. This is easy to do in polynomial time. Therefore, the optimal mapping scheduling(OMS) problem is a NP complete problem.

Therefore, the optimal mapping scheduling scheme can only be completed in non-polynomial time, and cannot be applied to the actual video analysis task scheduling system. Especially when the number of cameras and edge severs increases to a certain extent, the execution time of the optimal mapping scheduling algorithm will also increase exponentially. Therefore, in order to realize the mapping scheduling between camera and edge server in polynomial time, reduce the time complexity of mapping scheduling and maintain good performance, this paper develops the following heuristic mapping scheduling algorithm between camera and edge server.

4.2. Methodology of mapping scheduling scheme design

From the discussion in 4.1, we cannot obtain the optimal mapping scheduling scheme ($F : f(i, j)$) in linear time to satisfy the minimization of the objective equation. However, at the beginning of mapping scheduling, when the bandwidth resource consumption and computing resource consumption of the overall cameras are determined, we can seek the minimum value of the objective equation under this condition. So that we can obtain the ideal resource consumption load that should be allocated to each edge data center, and then determine which cameras should be mapped to the edge data center.

According to the operation process of EC-MASS, the unit bandwidth converted price m_j of each edge data center j is recorded in EC-Controller as a preset condition. Moreover, at the beginning of each mapping scheduling period, each edge data center profiles the video configuration of the main camera in the camera clusters and sends the optimal configuration to EC-Controller. Utilizing these video configurations, EC-Controller calculates the bandwidth resource consumption b_i of each camera i in the system in advance and records it as a condition for generating the mapping scheduling scheme.

First, we sort each edge data center j according to its unit bandwidth converted price m_j incrementally, that is, $m_1 \leqslant m_2 \leqslant m_3 \leqslant \cdots \leqslant m_J$. Then, we sort each camera i according to its bandwidth resource consumption b_i incrementally, that is, $b_1 \leqslant b_2 \leqslant b_3 \leqslant \cdots \leqslant b_I$.

Note that the objective equation presented in 3.2.3 is one form of convex optimization problems [13], we can solve the objective equation by applying the method of Lagrange multipliers [14], which is widely used to address this type of problem [15–17].

First, we rewrite the objective equation into the following form using the variable relationships introduced in 3.2.2.

$$\begin{aligned} M + \gamma\sigma(S) &= \sum_j B_j m_j + \gamma \frac{1}{J} \sum_j (S_j - \frac{1}{J}A)^2 \\ &= \sum_j B_j m_j + \gamma \frac{1}{J} \sum_j (\mu B_j - \frac{1}{J}A)^2 \end{aligned}$$

where $B_j = \sum_{i \in I} b_i$, which stands for the bandwidth resource load of the edge data center j. Besides, B_j is a discrete variable, since it is the sum of several b_i.

To minimize the above objective equation, we need to solve for specific values of $B_j (j = 1, 2, 3 \dots, J)$ satisfying the constraints. For simplicity, we replace the discrete variable B_j in the objective equation with the continuous variable x_j, resulting in the following equation.

$$g(x_1, x_2, \dots, x_J) = \sum_j x_j m_j + \gamma \frac{1}{J} \sum_j (\mu x_j - \frac{1}{J}A)^2$$

where $\sum_j x_j = \frac{A}{\mu}$ and $x_j \geqslant 0$. From this condition, we can construct the following equality constraint

$$\phi(x_1, x_2, \dots, x_J) = x_1 + x_2 + \cdots + x_J - \frac{A}{\mu} = 0$$

To incorporate these conditions into one equation, we introduce an auxiliary function

$$G(x_1, x_2, \dots, x_J, \lambda) = g(x_1, x_2, \dots, x_J) + \lambda\phi(x_1, x_2, \dots, x_J)$$

and solve

$$\nabla_{x_1, x_2, \dots, x_J, \lambda} G(x_1, x_2, \dots, x_J, \lambda) = 0$$

Note that this amounts to solving the following $J + 1$ equations in $J + 1$ unknowns.

$$\begin{cases} F'_{x_1} = m_1 + \frac{2\mu\gamma}{J}(\mu x_1 - \frac{A}{J}) + \lambda = 0 \\ F'_{x_2} = m_2 + \frac{2\mu\gamma}{J}(\mu x_2 - \frac{A}{J}) + \lambda = 0 \\ \dots \\ F'_{x_J} = m_J + \frac{2\mu\gamma}{J}(\mu x_J - \frac{A}{J}) + \lambda = 0 \\ x_1 + x_2 + \cdots + x_J - \frac{A}{\mu} = 0 \end{cases}$$

The solution is as follows:

$$\begin{cases} \lambda = -\frac{1}{J}\sum_j m_j \\ x_j = \frac{1}{J\mu}A - \frac{J}{2\mu^2\gamma}(m_j - \frac{1}{J}\sum_j m_j) \end{cases}$$

With this solution, we can get the minimum value of $g(x_1, x_2, \dots, x_J)$. In other words, we can minimize the objective equation when the bandwidth resource load B_j allocated to each edge data center j is equal to x_j. However, x_j is the ideal value obtained on the assumption that the variable is continuous, which does not accord with the actual situation that the camera bandwidth resource consumption is discrete. We will discuss the insight on the utilization of $(x_1, x_2, \dots, x_J)$ in the mapping scheduling algorithm in the next section.

4.3. Insight of mapping scheduling equations design

Although the ideal $(x_1, x_2, \dots, x_J)$ is not in line with the actual situation of camera bandwidth resource consumption in our system, we

can still use it as a guiding scheme for us to design mapping scheduling algorithm. For example, we can allocate cameras with bandwidth resource consumption b_i to the corresponding edge data center j while the condition that $B_j = \Sigma_{i \in I} b_i$ is close to x_j can be satisfied.

However, the above allocation method will lead to the multi-solution of the mapping scheduling scheme, that is, there will be multiple mapping scheduling schemes that satisfy the above condition. Therefore, it is necessary to further determine the appropriate allocation method to ensure the uniqueness and rationality of the mapping scheduling scheme during the operation of the system. In the following, we describe the insights on economy consideration of camera allocation in the mapping scheduling algorithm.

Since the configuration of the camera i changes over time, the bandwidth resource consumption of b_i also changes dynamically. At the same time, there is an upper limit of the camera bandwidth resource consumption of b_i because the camera configuration has a golden configuration, which is the most expensive configuration. So for the camera with lower bandwidth resource consumption b_i, its increase of bandwidth Δb_i is larger over time. Therefore, if the camera i with lower bandwidth resource consumption b_i is mapped at the t_1 moment to the edge data center j with higher unit bandwidth converted price m_j, the increase of the total bandwidth and computing resource cost ΔM of the system may also be higher during the interval $t_2 - t_1$, where t_2 is the moment when the next mapping scheduling begins.

In order to optimize the economy of the system during the mapping scheduling interval $t_2 - t_1$, cameras that consume less bandwidth resources should be mapped to edge data centers with lower unit bandwidth converted price. According to our insight, the camera allocation in the mapping scheduling algorithm is designed in the following manner:

$$B_1 = b_1 + \cdots + b_{i1}$$
$$B_2 = b_{i1+1} + \cdots + b_{i2}$$
$$\cdots$$
$$B_j = b_{i(j-1)+1} + \cdots + b_{ij}$$
$$\cdots$$
$$B_J = b_{i(J-1)+1} + \cdots + b_I$$

While B_j meets the condition that

$$B_j - b_{ij} \leqslant x_j \leqslant B_j$$

The above equations are called the mapping scheduling equations. Among them, B_j with the smaller sequence number corresponds to the bandwidth resource load of the edge data center j with a lower unit bandwidth converted price m_j; at the same time, the bandwidth resource consumption b_i of cameras are also sorted incrementally.

By serial number i and j, we can determine the corresponding cameras and edge data centers, and generate the mapping scheduling scheme $F : f(i, j)$.

4.4. Implementation and analysis of algorithm

Based on the scheduling equations of the algorithm discussed above, we provide the pseudo code of the mapping scheduling algorithm, and describe the analysis of the code.

Algorithm 1 lists the steps taken in each mapping scheduling period. First, the algorithm takes the resource conditions E of edge data centers, such as bandwidth resource limits BL_j, computing resource limits SL_j, and unit bandwidth converted price m_j as input. Besides, the dynamic video configurations C of cameras profiled by video analytics system are also taken as input. Based on C, algorithm calculates the resource consumption b_i (line 2). After the cameras and the edge data centers are sorted incrementally (line 4 and 5), camera tasks are assigned to each edge data center according to the mapping scheduling equations (line 6–10), and finally the mapping result is output (line 11).

The code of Algorithm 1 is mainly composed of two parts:

- Sorts the n cameras and m edge data centers;
- Assigns cameras to each edge data center following the mapping function.

The time complexity of part 1 consists of sorting n cameras and m edge data centers. Here, the sorting method we utilize is Quicksort. Quicksort is often the best practical choice for sorting because it is remarkably efficient on the average. The expected running time for part 1 is $O(n \log n + m \log m)$ [18].

Besides, the time complexity of part 2 is $O(n + m)$ which consists of the assignment between cameras and edge data centers. Consider the following allocation process: According to the mapping function, select a certain number of cameras in order from the sorted camera set CAM of size n, and allocate these cameras to the first edge data center ecc_1. Then continue to select cameras from the remaining camera set. and allocate to the ecc_2, keep going until the ecc_m. Intuitively, the running time of this allocation process is $O(n+m)$. So the total time complexity of algorithm 1 is $O(n \log n + m \log m)$.

Although the application of exhaustive algorithm (i.e. trying all choices of mapping between cameras and edge data centers) can provide the optimal mapping scheme, its time complexity is $O(m^n)$, which is exponential. Because for each camera, we have to consider its allocation to m edge data centers. For n cameras, the running time of this procedure is $O(m^n)$. Obviously, the algorithm we designed is better in terms of time complexity. The comparison of the mapping results between the two algorithms will be given in 5.3.

Algorithm 1 Mapping Scheduling.

Input: C: set of all configs of cameras
E: set of all resource conditions of edge data centers.
Output: The mapping scheme ($F : f(i, j)$): the map from each camera cam_i to each edge data center ecc_j.

```
1: function MappingScheduling(C, E)
2:     Calculate cameras resource consumption b_i
3:     Result ← ∅
4:     Sort cam_i by their resource consumption b_i
5:     Sort ecc_j by their comprehensive resource price m_j
6:     for each ecc_j do
7:         By scheduling function
8:         ecc_i(cam) ← (cam_i1, ··· , cam_in)
9:         Result.add(ecc_i(cam))
10:    end for
11:    return Result
12: end function
```

5. Evaluation

We evaluate EC-MASS on a dataset of video configurations of real-world cameras which randomly generates configurations for cameras according to uniform distribution. Our key findings are listed in the following.

- EC-MASS can ensure that the cameras' task load is relatively evenly distributed in the edge data centers in real time 5.2.
- Strategy utilized in EC-MASS is reliable and has lower control overheads compared with the strategy that exhaustively searches the optimal scheduling scheme 5.3.
- EC-MASS achieves lower operating cost and better system performance than a baseline of the traditional system mapping connection to the nearest once upfront 5.4.
- By adjusting the CPB, EC-MASS shows the ability of accommodating the actual needs of different users for the cost and performance of the video analytics system in different scenarios 5.5.
- For the scalable video analytics system, EC-MASS can achieve promising performance advantages as well 5.6.

Table 2
System parameters.

Edge data centers	Ecc_0	Ecc_1	Ecc_2	Ecc_3
Bandwidth resource limits	1200	1200	1200	1200
Computing resource limits	2000	2000	2000	2000
Unit bandwidth resource price	0.2	0.4	0.4	0.5
Unit computing resource price	0.11	0.11	0.12	0.12

5.1. Dataset and setup

We used a dataset of video configurations from the real-world camera. In Chameleon, the video analytics system periodically profiles the video configuration for each pre-divided camera cluster in order to provide the optimal configuration. Because the conditions of the real environment change randomly over time, the optimal video configuration of the camera can be regarded as changing randomly over a period of time (24 h). With the above insight, in the simulation experiment, for simplicity, we generate real-time configuration periodically and randomly for each camera cluster according to the probability model of uniform distribution from the configuration dataset. Because the experiment is to verify the running status of the EC-MASS system when the video configuration is updated in real time, therefore, the above simplified measures are reasonable and can meet the needs of the experiment.

Our testbed consists of 60 cameras and 4 edge data centers. In the simulation experiment, we need to balance the load of different edge data centers to achieve stable performance while minimizing the overall computing and bandwidth cost of the system. For simplicity, cameras are of the same model, while the edge data centers have the same upper limits of bandwidth and computing resources and different bandwidth and computing prices. Referring to the configuration and charging standard of GPU CVM provided by Aliyun website, we set experimental parameters such as resource limits and price of edge data centers, which are listed in Table 2.

5.2. The running state of EC-MASS

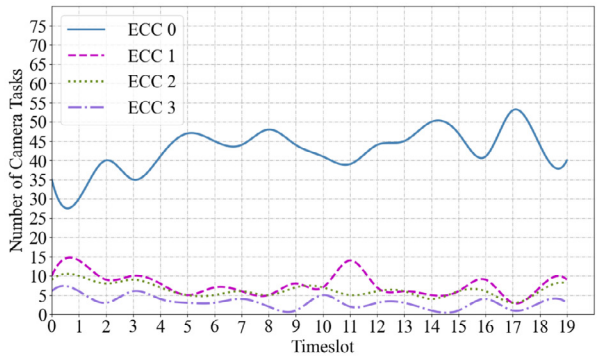
We run EC-MASS for 24 h to observe the running status of the system. In the experiment, we divide the camera clusters in advance. We update the best configuration for different camera clusters every 16 s, and schedule the mapping between the cameras and the edge data centers in the meanwhile. We set CPB γ to 0.0009, and we will discuss CPB in 5.5.

Fig. 3(a) shows the variation of the number of camera tasks in each edge data center during EC-MASS operation. Although the amount of tasks in the edge data center ECC_0 is much higher than other tasks, it can be seen from Figs. 3(b) and 3(c) that the consumption of the bandwidth and computing resources on ECC_0 is still relatively reasonable. The underlying reason is that the resource requirements of camera tasks mapped to the ECC_0 are relatively low.

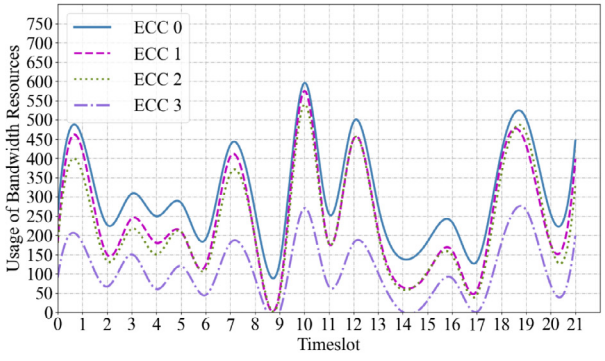
Experiments show that during the running period, EC-MASS, according to the real-time video configuration of cameras, can realize the dynamic mapping schedule between cameras and edge data centers, so that the workload of the camera tasks on each edge data center can be relatively balanced.

5.3. Strategy reliability and control overheads

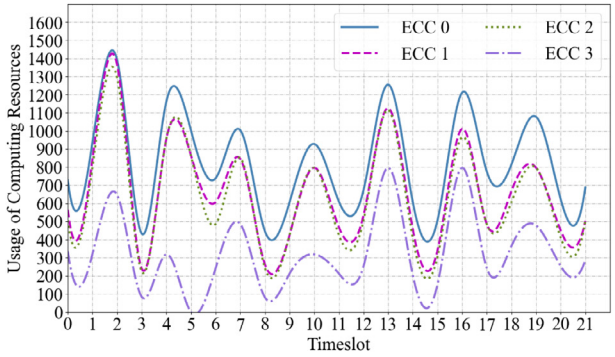
To reduce the time complexity of the running time, the scheduling strategy applied in EC-MASS does not search the entire scheduling scheme space. Therefore, the heuristic algorithm we designed may ignore the optimal scheme that minimizes the objective equation, and can only achieve suboptimal performance. To verify the reliability of the designed strategy, we compared with the optimal scheme obtained by the exhaustive search algorithm based on the depth-first strategy.



(a) Variation of the number of camera tasks



(b) Variation of bandwith resource consumption



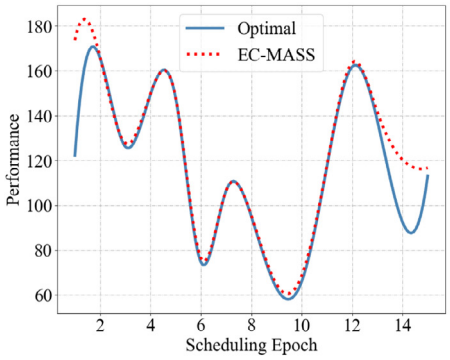
(c) Variation of computing resource consumption

Fig. 3. The running state of EC-MASS.

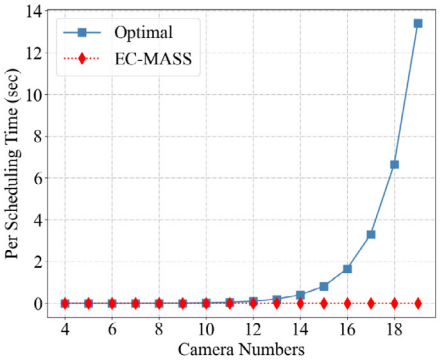
And we set the minimum value of the objective equation during the scheduling process as the performance metric.

Fig. 4(a) shows that the map-scheduling algorithm adopted in EC-MASS produces the same optimal performance in the majority the time (85%) compared to the exhaustive search algorithm (Optimal). Furthermore, the time complexity of the algorithm to obtain the optimal scheme is exponential. As demonstrated in Fig. 4(b), when the scale of cameras is increased from 4 to 19, the time per scheduling of Optimal algorithm rises to 13 s, while the running time of EC-MASS algorithm is still under millisecond level.

Then, we compared the differences between the two algorithms in the aspect of control overheads. Considering that Optimal algorithm is limited by the scale of devices, in the experiment, we deployed 2 edge data centers and 10 cameras. We used a desktop computer with Inter(R) Core(TM) i7-10700 CPU as EC-Controller.



(a) Performance analysis of EC-MASS and Optimal



(b) Running time of EC-MASS and Optimal in the scalable case

Fig. 4. Strategy reliability of EC-MASS compared with the optimal algorithm.

Table 3
Control overheads of EC-MASS and Optimal.

	EC-MASS	Optimal	Improv.
CPU utilization	8%	11%	27.2%
CPU occupancy time	41.8 msec	12.3 sec	99.6%

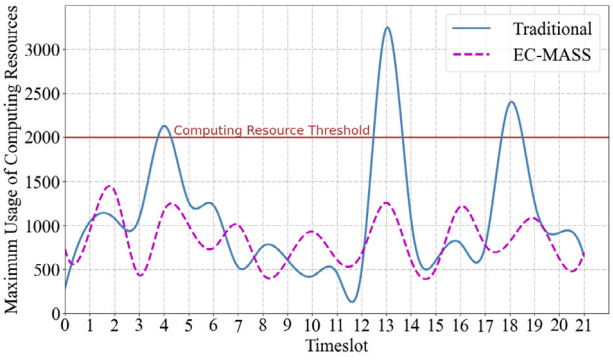
Table 3 shows the experimental results after 500 rounds of mapping scheduling. We can observe that the EC-MASS algorithm achieves 27.2% reduction in CPU utilization and 99.6% reduction in CPU occupancy time compared with Optimal algorithm.

Therefore, the scheduling strategy we designed can guarantee reliable optimization performance and has significant advantages in terms of time complexity and control overheads.

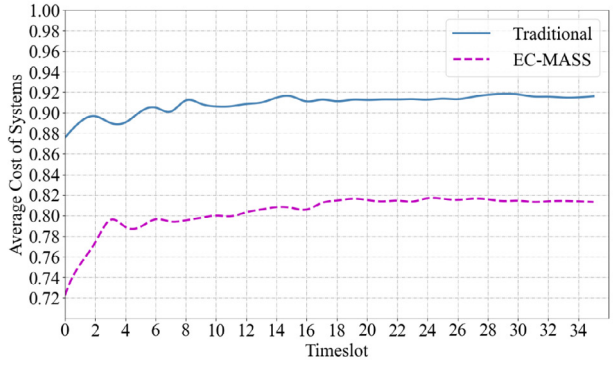
5.4. Performance improvement

We use the traditional system mapping connection to the nearest once upfront as our baseline. In the experiment, we analyze the performance and cost of EC-MASS. We use the number of times the computing load of each edge data center exceeds its upper limit within 1.5 h of the experiment as an indicator of performance comparison. Fig. 5(a) shows that the performance of EC-MASS is better, and there is no load exceeding the upper limit, compared with three times in traditional system experiments. Fig. 5(b) shows that EC-MASS has also improved significantly in terms of economy. Compared with traditional systems, the average operating cost of EC-MASS is reduced by about 10.9%.

The experimental results show that EC-MASS has obvious advantages over the existing traditional video analytics systems in terms of performance and cost.



(a) Performance analysis of EC-MASS and Traditional



(b) The average cost of EC-MASS and Traditional variation

Fig. 5. Performance and cost improvement of EC-MASS.

5.5. Impact of CPB γ

In the mapping scheduling algorithm we designed, the function of CBP γ is to balance the economy and performance of EC-MASS. Fig. 6(a) shows that as γ increases, the average cost of the system increases accordingly. But that does not mean we should choose the smallest γ.

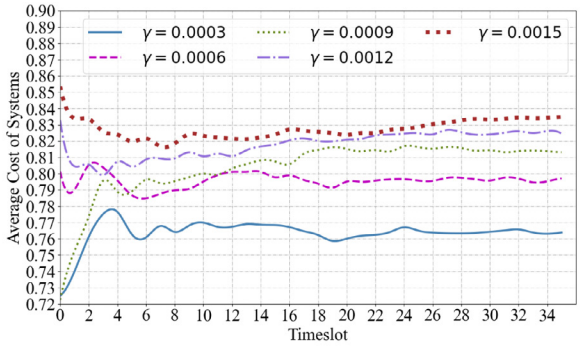
Fig. 6(b) shows the change in the total cost and average performance (expressed by the mean of the variance of the computing load) of the system running for 24 h as γ increases. We find that when γ decreases, the variance of the computing load of the system increases greatly. This will lead to the existence of idle status in some edge data centers, resulting in a reduction in system resource utilization. Therefore, in different scenarios, users should choose their own γ value according to their actual requirements for the economy and performance of the video analysis system. In our experiment, we chose the intermediate value $\gamma = 0.0009$.

5.6. Scaling performance

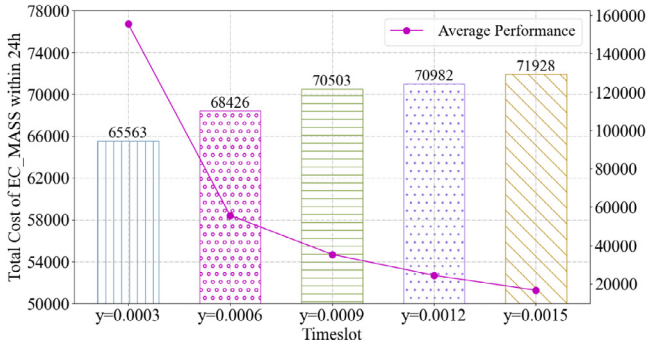
On the basis of our experiments, we increase the number of cameras in the system to observe the scalability of EC-MASS. Fig. 7 shows that when the number of cameras increases from 60 to 80, then to 100, the system can still meet the performance requirements (e.g. not exceed the bandwidth and computing resource constraints of the edge data centers). Therefore, EC-MASS is applicable for the scalable video analytics system.

6. Related work

Video analytics system: Since 2016, a sizable body of work on video analysis has emerged in the systems and data management



(a) Impact of CPB γ on the cost



(b) Impact of CPB γ on the performance

Fig. 6. Impact of CPB γ of EC-MASS.

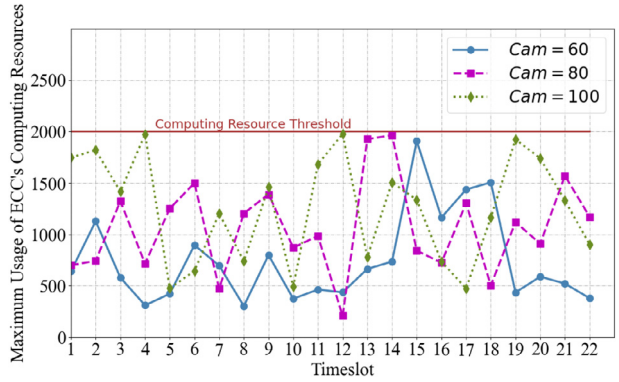


Fig. 7. Scaling performance of EC-MASS.

community. Chameleon exploits correlations in camera content (e.g. velocity and sizes of detected objects) to amortize profiling costs across cameras over time [2]. Optasia parallelizes the video query plans and reduces the repetitive work of commonly used modules in the data flow framework, such as background subtraction, to shorten query completion time and reduce resource usage [19]. VideoStorm investigates differences in quality lag requirements between common video analytics queries (e.g. scanning license plates billing on toll routes vs. issuing AMBER Alerts) and proposes offline profilers and online schedulers for best performance [8]. NoScope accepts specialized queries (e.g. "find all frames with buses in Taipei feeds") and uses difference detectors and specialized models to build a model cascade to accelerate most inputs [9]. Focus invokes object clustering and low-cost models to index videos cheaply during ingestion, thus supporting low-latency, post-event queries for historical videos [11]. CONVINCE proposes to push video analysis to the edge of the network, and designs a centralized collaborative cross-camera video analysis system at the edge, which uses the spatio-temporal correlation and knowledge sharing across cameras to reduce computing and bandwidth requirements while protecting privacy [20].

Video processing optimization: Most papers consider to optimize the video processing pipelines by adjusting the configuration knobs or training specialized NN models. Chameleon demonstrates that the optimal configurations do change over time, so it adjusts the configuration of the video analytics pipeline in real time to optimize resource consumption and inference accuracy by using the temporal and spatial correlation of the configuration [2]. Similarly, [21] retrains the NN model to detect a set of popular objects that change over time. However, other papers such as VideoStorm [8], NoScope [9], MCDNN [10] and Focus [11], all report significant improvements in accuracy and/or resource consumption, but they analyze and optimize video queries only once at the beginning of the video stream. They do not report how the best profile changes over time, nor do they handle changes in the content of the video stream. Among them, VideoStorm first profiles each video query running in the cluster, and then adjusts its configuration to achieve the right balance between accuracy, processing latency, and resource demand [8–11] specialize neural networks trained based on objects that usually appear in a particular video stream to detect objects and identify characters and texts.

Edge computing-based task scheduling: Edge computing, also known as mobile edge computing [22], fog computing [23], cloudlet [24] and so on, is an emerging technology that brings cloud computing capabilities closer to user equipment to perform computationally-intensive tasks and store a massive amount of data [25–27]. In edge computing, task scheduling and computation offloading are the key solutions to enhance the overall performance and capacity of this computational paradigm [28]. A majority of these techniques are designed to apply to different application scenarios. For example, Ahmed et al. [29] leveraged genetic algorithm and conflict graph models to achieve parallel and sequential task offloading to multiple mobile edge computing (MEC) servers and minimize both offloading latency and failure probability. Wang et al. [30] presented a two-layer optimization method, ToDeTaS, to establish a new multiunmanned aerial vehicle enabled MEC system for joint deployment and task scheduling optimization. Ning et al. [31] designed a scheme called MEES for constructing an energy-efficient scheduling framework and minimized the energy consumption of MEC-enabled road side units (RSUs) under task latency constraints. Wang et al. [32] proposed a task scheduling approach for health IoT systems at smart homes called HealthEdge, which can assign tasks between the edge workstation and the private cloud data center with the shortest estimated processing time.

For the scenario of video analytics in edge computing, task scheduling also helps to improve resource utilization, response time, and energy consumption. LAVEA [33] provides low-latency video analytics at places closer to the users by offloading computation between clients and edge nodes while collaborating nearby edge nodes. JCAB [34], jointly optimizing configuration adaption and bandwidth allocation for multiple video streams, can effectively balance the analytics accuracy and energy consumption while keeping low system latency. Different from these works, instead of computation offloading or bandwidth allocation for reducing system latency, EC-MASS dynamically schedules video analytics tasks between multiple edge data centers to guarantee system performance stability while reducing system costs.

7. Conclusion and future work

In this paper, we argue that the mapping scheduling of video analytics system should be adapted according to the updated configurations of cameras, otherwise it may result in high operation cost and low performance stability. We design EC-MASS, a video analytics system

leveraging mapping scheduling algorithm to dynamically adjust the mapping relationship between multi-cameras and multi-edge data centers according to the configuration dynamics. Compared with Optimal, EC-MASS shows its reliability of strategy and low control overheads. Compared with Traditional, EC-MASS achieves better performance stability (theoretically does not exceed the system resource limit, while the former exceeds three times in 1.5 h), and lower cost (about 10.9%).

Besides load balancing and system cost, which are the focus of this paper, latency is also an important performance metric of video analytics. The topology in the real-world network environment is complex, and different network topologies will affect the connection establishment and video stream transmission latency of video analytics. While EC-MASS optimizes the load balancing and system cost of video analytics, its techniques adjusting the mapping relationship between cameras and edge data centers according to the configuration dynamics will likely carry over when considering the different network topologies. However, to provide better performance of video analytics system, optimizing the latency caused by network topology in EC-MASS will be an important work going forward.

CRediT authorship contribution statement

Shu Yang: Conceptualization, Methodology, Writing – original draft. **Qingzhen Dong:** Methodology, Software, Writing – original draft. **Laizhong Cui:** Supervision, Writing – review & editing. **Xun Chen:** Formal analysis, Writing – original draft. **Siyu Lei:** Writing – original draft. **Yulei Wu:** Validation. **Chengwen Luo:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C.-C. Hung, G. Ananthanarayanan, P. Bodik, L. Golubchik, M. Yu, P. Bahl, M. Philipose, Videoedge: Processing camera streams using hierarchical clusters, in: 2018 IEEE/ACM Symposium on Edge Computing, SEC, IEEE, 2018, pp. 115–131.

[2] J. Jiang, G. Ananthanarayanan, P. Bodík, S. Sen, I. Stoica, Chameleon: scalable adaptation of video analytics, in: S. Gorinsky, J. Tapolcai (Eds.), Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 2018, Budapest, Hungary, August 20-25, 2018, ACM, 2018, pp. 253–266, http://dx.doi.org/10.1145/3230543.3230574.

[3] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, M.J. Freedman, Live video analytics at scale with approximation and delay-tolerance, in: 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17), 2017, pp. 377–392.

[4] X. Zeng, B. Fang, H. Shen, M. Zhang, Distream: scaling live video analytics with workload-adaptive distributed edge intelligence, in: Proceedings of the 18th Conference on Embedded Networked Sensor Systems, 2020, pp. 409–421.

[5] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, 2018, arXiv preprint arXiv:1804.02767.

[6] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Comput. Sci. (2014).

[7] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2) (2012).

[8] H. Zhang, G. Ananthanarayanan, P. Bodík, M. Philipose, P. Bahl, M.J. Freedman, Live video analytics at scale with approximation and delay-tolerance, in: A. Akella, J. Howell (Eds.), 14th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2017, Boston, MA, USA, March 27-29, 2017, USENIX Association, 2017, pp. 377–392, URL https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/zhang.

[9] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, M. Zaharia, Optimizing deep CNN-based queries over video streams at scale, 2017, CoRR abs/1703.02529arXiv:1703.02529.

[10] S. Han, H. Shen, M. Philipose, S. Agarwal, A. Wolman, A. Krishnamurthy, MCDNN: an approximation-based execution framework for deep stream processing under resource constraints, in: R.K. Balan, A. Misra, S. Agarwal, C. Mascolo (Eds.), Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2016, Singapore, June 26-30, 2016, ACM, 2016, pp. 123–136, http://dx.doi.org/10.1145/2906388.2906396.

[11] K. Hsieh, G. Ananthanarayanan, P. Bodík, S. Venkataraman, P. Bahl, M. Philipose, P.B. Gibbons, O. Mutlu, Focus: Querying large video datasets with low latency and low cost, in: A.C. Arpaci-Dusseau, G. Voelker (Eds.), 13th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2018, Carlsbad, CA, USA, October 8-10, 2018, USENIX Association, 2018, pp. 269–286, URL https://www.usenix.org/conference/osdi18/presentation/hsieh.

[12] M.R. Garey, D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, 1983, W.H. Freeman.

[13] S. Sebastio, G. Gnecco, A. Bemporad, Optimal distributed task scheduling in volunteer clouds, Comput. Oper. Res. 81 (2017) 231–246.

[14] R.T. Rockafellar, Lagrange multipliers and optimality, SIAM Rev. 35 (2) (1993) 183–238.

[15] K. Park, C.A. Felippa, U. Gumaste, A localized version of the method of Lagrange multipliers and its applications, Comput. Mech. 24 (6) (2000) 476–490.

[16] M.L. Fisher, Optimal solution of scheduling problems using Lagrange multipliers: Part I, Oper. Res. 21 (5) (1973) 1114–1127.

[17] L. Huang, M.J. Neely, Delay reduction via Lagrange multipliers in stochastic network optimization, in: 2009 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, IEEE, 2009, pp. 1–10.

[18] P. Hennequin, Combinatorial analysis of quicksort algorithm, RAIRO - Theor. Inf. Appl. - Inform. Théor. Appl. 23 (3) (1989) 317–333, URL http://eudml.org/doc/92337.

[19] Y. Lu, A. Chowdhery, S. Kandula, Optasia: A relational platform for efficient large-scale video analytics, in: M.K. Aguilera, B. Cooper, Y. Diao (Eds.), Proceedings of the Seventh ACM Symposium on Cloud Computing, Santa Clara, CA, USA, October 5-7, 2016, ACM, 2016, pp. 57–70, http://dx.doi.org/10.1145/2987550.2987564.

[20] H.B. Pasandi, T. Nadeem, CONVINCE: collaborative cross-camera video analytics at the edge, in: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2020, Austin, TX, USA, March 23-27, 2020, IEEE, 2020, pp. 1–5, http://dx.doi.org/10.1109/PerComWorkshops48775.2020.9156251.

[21] H. Shen, S. Han, M. Philipose, A. Krishnamurthy, Fast video classification via adaptive cascading of deep models, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017, pp. 2197–2205, http://dx.doi.org/10.1109/CVPR.2017.236.

[22] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal, et al., Mobile-edge computing introductory technical white paper, White Paper, Mobile-Edge Comput. (MEC) Ind. Initiative 29 (2014) 854–864.

[23] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 2012, pp. 13–16.

[24] M. Satyanarayanan, Pervasive computing: Vision and challenges, IEEE Pers. Commun. 8 (4) (2001) 10–17.

[25] N. Hassan, K.-L.A. Yau, C. Wu, Edge computing in 5G: A review, IEEE Access 7 (2019) 127276–127289.

[26] E. Ahmed, A. Ahmed, I. Yaqoob, J. Shuja, A. Gani, M. Imran, M. Shoaib, Bringing computation closer toward the user network: Is edge computing the solution? IEEE Commun. Mag. 55 (11) (2017) 138–144.

[27] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, J. Zhang, Edge intelligence: Paving the last mile of artificial intelligence with edge computing, Proc. IEEE 107 (8) (2019) 1738–1762, http://dx.doi.org/10.1109/JPROC.2019.2918951.

[28] X. Shan, H. Zhi, P. Li, Z. Han, A survey on computation offloading for mobile edge computing information, in: 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing,(HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), IEEE, 2018, pp. 248–251.

[29] A.A. Al-Habob, O.A. Dobre, A.G. Armada, S. Muhaidat, Task scheduling for mobile edge computing using genetic algorithm and conflict graphs, IEEE Trans. Veh. Technol. 69 (8) (2020) 8805–8819.

[30] Y. Wang, Z.-Y. Ru, K. Wang, P.-Q. Huang, Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing, IEEE Trans. Cybern. 50 (9) (2019) 3984–3997.

[31] Z. Ning, J. Huang, X. Wang, J.J. Rodrigues, L. Guo, Mobile edge computing-enabled Internet of vehicles: Toward energy-efficient scheduling, IEEE Netw. 33 (5) (2019) 198–205.

[32] H. Wang, J. Gong, Y. Zhuang, H. Shen, J. Lach, Healthedge: Task scheduling for edge computing with health emergency and human behavior consideration in smart homes, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 1213–1222.

[33] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, Q. Li, Lavea: Latency-aware video analytics on edge computing platform, in: Proceedings of the Second ACM/IEEE Symposium on Edge Computing, 2017, pp. 1–13.

[34] C. Wang, S. Zhang, Y. Chen, Z. Qian, J. Wu, M. Xiao, Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 257–266.