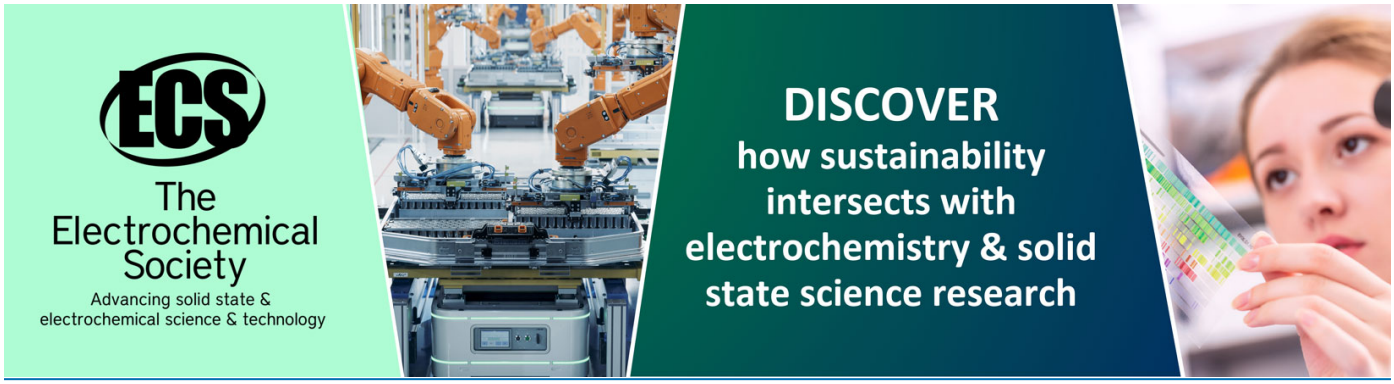
Mobile Edge Computing Based Video Surveillance System Using Special Hardware and Virtualization Technology



Mobile Edge Computing Based Video Surveillance System Using Special Hardware and Virtualization Technology

Shaowei Liu, Tingting Zhang, Ningbo Cui, Hao Zhang* and Jiayuan Chen

Department of Network and IT Technology Research, China Mobile Research Institute, No.32 Xuanwumen west street, Xicheng District, Beijing, China
Email: {liushaowei, zhangtingtingyj, cuiningbo, zhanghao*, chenjiayuan}@chinamobile.com

Abstract. As technology evolution, the solution for society security system is a hot research topic. Video surveillance is an important guarantee for society security. To meet various application scenarios, quick end-to-end (E2E) response is the key of video surveillance system. Mobile edge computing (MEC) could reduce E2E delivery delay by processing data nearby the data source. This paper introduces a MEC based solution that uses special hardware, graphics processing units (GPU), to accelerate graphic data processing. Taking the limited resource of edge scenario into consideration, virtualization technology is used to share computing resource on demand. The solution also proposes deployment policy basing on functional component's requirement. Finally, this paper verifies the solution and puts forward recommendations.

1. Introduction

With the development of economy and national urbanization process, a large number of rural residents migrate to cities. Therefore, the population density of city increases [1]. Even more, with modern vehicle, people become more mobile. Convenient transportation and large population mobility bring lots of challenge to social security system, and make it more difficult to crack down on and arrest criminal suspects. Hence, it is urgent to introduce advanced technology and effective methods to strengthen crime prevention and control.

Currently, the combination of video service and mobile edge computing (MEC) is regarded as a promising solutions. With the rapid development of IP network, each device connected to the network could communicate with each other. As one of the beneficiaries, video service has bloomed and realized the large-scale application in culture, education, entertainment, medical health, intelligent transportation, industrial manufacturing, video surveillance, and other fields. In the past few years, video service has evolved from high-definition (HD) to ultra-high-definition (UHD), which contributes to improve quality of user experience and expand industrial ecology. In video surveillance, UHD image provides more detailed information which supports and optimizes intelligent processing, such as, face detection along with attributes, and tracing tracking, etc.

Generally video surveillance system is composed of video terminal and video processor. Video terminal collects video data and receives instruction. According to mobility, video terminal has different existences, such as fixed-camera and portable camera. Corresponding, video terminal communicates with video processor by either Internet or cellular network. Video processor analyzes and processes video data received from video terminal, and sends instructions to video terminal. According to the requirement of delay, processing tasks divide into real-time tasks and background tasks. Real-time task, such as face detection and trace tracking in arrest, has strict requirements on response delay including delivery delay and processing delay.

Although UHD video brings more detailed information, it produces huge data. As a result, UHD brings heavy traffic to network and requires more computing capacity. As an emerging technology, MEC has great potential to deal with these problems [2].

MEC [3] brings server and data center closer to users, at the edge of the network. By placing resources physically near the source of data, instead of in data centers hundreds or thousands of miles away, MEC can reduce traffic on mobile network and delivery delay. In addition, comparing to core network, edge network has limited resource and computing capacity. MEC could cooperate with core network to complete complex task, especially in artificial intelligence (AI) area. For example, core network trains model, and edge network utilizes these abstract reasoning model to compute. Considering the limited resource of edge scenario, how to reduce processing delay is a huge challenge for MEC.

With the evolution of technology, video surveillance has experienced the video processor shifts from core to edge of network. This paper will introduce a solution that uses graphics processing units (GPU) and virtualization to reduce video data processing delay of real-time task, considering the features of video data processing and limited resource of edge scenario.

2. Related Work

In the past few years, several solutions for video surveillance have been proposed. In these solutions, video terminal usually consists of video collector and instruction receiver. Sometimes, video collector and instruction receiver can merge into one device. Video collector collects video data, and instruction receiver receives instruction. According to mobility, video collector can divided into fixed camera and portable camera. Fixed camera does not move at all, portable camera may be carried by law enforcement officers and law enforcement vehicles. Video processor could be located in either core network or edge network.

Early on, video processor was located in core network, as shown in Fig 1.

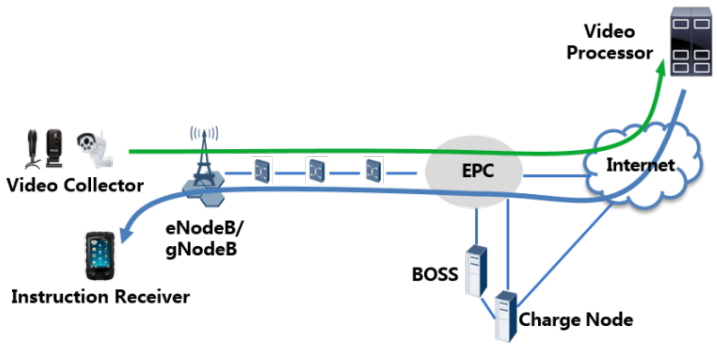


Figure 1. Video processor locates in core network.

The distance between video collector (instruction receiver) and video processor may be hundreds or thousands of miles. All workload of data processing is conducted by video processor. The solution has several limitations.

1) Real-time requirement. Based on the traditional computing and network model, video data is sent to core network and processed by video processor. Long delivery distance and traditional computing architecture will lead to high delivery latency and processing delay, respectively. High delay can hardly meet scenarios with real-time nature of the data processing. For example, in the action of real time capture, the policeman do not receive the trace of criminals until they are out of sight.

2) Bandwidth pressure. Because data is processed at core network, video collector has to send all video data from edge network to core network. Video data delivery, especially UHD video data, brings huge traffic to network, which will cause network bandwidth pressure and high cost.

3) Data security and privacy. Data delivery will pass through multiple network devices which increases the risk of data leakage and code hijacking.

Later, the location of video processor shifts from core network to edge network, as shown in Fig 2. Edge computing infrastructure provides virtual resource, such as compute, storage, and network. Edge computing platform (ECP) provides networking and vertical industry capabilities for applications. Functions of video processor become edge computing applications (ECAs) to provide service. Edge network cooperates with core network to process video data. Core network is responsible for training model which can be done background, and distributing trained model to ECA, which helps ECA reduce calculation complexity. ECA processes video data near the source of video collector reducing the data delivery latency.

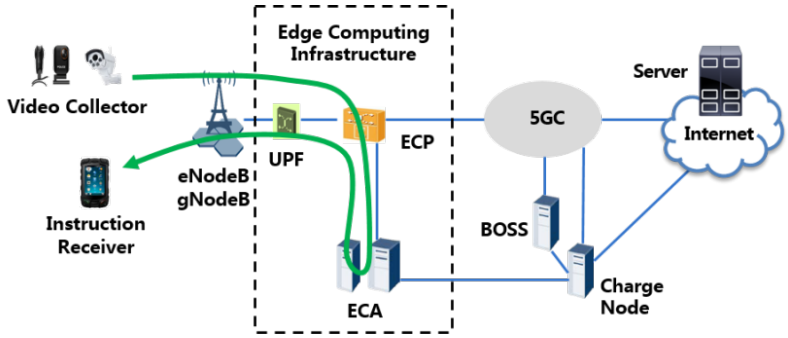


Figure 2. Video processor locates in edge network.

In order to achieve the purpose of real-time response, both delivery delay and processing delay are very important. Basing on MEC, how to reduce processing latency is another challenge for video surveillance system.

3. Special Hardware and Virtualization

This section first introduces the functional architecture of video surveillance system, then proposes a solution to accelerate video processing, and finally discusses the deployment policy and mode of the proposed solution.

3.1. Functional Architecture

Simplified functional architecture of video surveillance system includes terminal device, video processing platform, and application platform, as shown in Fig 3.

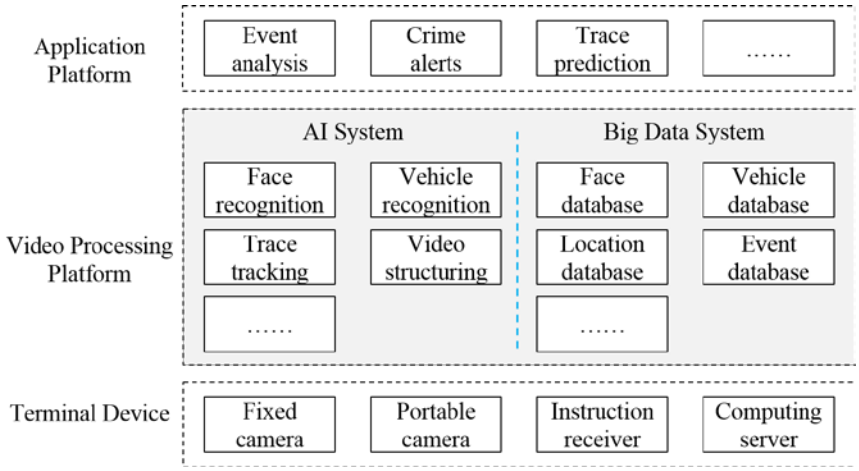


Figure 3. Simplified functional architecture of video surveillance system.

Terminal device includes different forms of camera, instruction receivers, and computing device.

Video processing platform is the key to reduce processing latency. It consists of AI system and big data system. AI system, including face recognition, vehicle recognition, video structuring, trace tracking, etc., provides structured data to the upper application platform on demand and assists core network analysis model training of new scenes. Big data system, including different kinds of database, analysis structured data and returns result to the upper application platform.

Application platform supports various kinds of video based applications, such as video monitoring, location management, trace analysis, etc.

3.2. GPU, Virtual Machine, and Container

Video processing platform is the most important part of video surveillance system and the key to reduce processing delay. Workload of video processing platform can be divided into graphics-related task and computing-related task. Modern GPU [4] is very efficient at manipulating computer graphics and image processing. It's highly parallel structure makes it more efficient than general-purpose CPUs for algorithms that process large blocks of data in parallel. It is crucial to adopt GPU to accelerate graphics-related processing.

From the perspective of limited resource in edge scenario, it is absolutely necessary to share resource among functional components to improve resource utilization. Furthermore, resource sharing need to consider the following aspects.

1) Real-time elasticity. Video processing platform should have the ability to fit the resources needed to copy with dynamical workloads. When the workload increases it scales by adding more resources, and then the demand wanes it shrinks back and removes unneeded resources. In order to reduce processing delay, resource elasticity should be quick enough.

2) Resource isolation. Each functional component should have logically independent resource to protect data security and minimize impact on others.

Container and virtual machine (VM) are typical realizations of virtualization technology that increases IT agility, flexibility and scalability. Although they have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware [5]. Comparing to VM, container is lightweight and quick to boot, but it is not isolated enough. Hence, secure container is proposed. Secure container with lightweight virtual machines feels and performs like containers, but provides stronger workload isolation using hardware virtualization technology. At present, Kata container is the best practice of secure container.

3.3. Deployment Policy and Mode

Considering computing type, resource sharing and isolation, deployment policy in different scenarios are proposed.

1) GPU and CPU. Graphics-related tasks adopt GPU to accelerate processing. Other tasks uses CPU (general server).

2) Resource granularity. GPU virtualization plug-in allows multiple guest VMs or containers to effectively share the physical compute resources. Functional components could own or share underlying compute resource. If functional component requires exclusive hardware, it should be deployed either on bare-metal directly or by pass-though. Otherwise, it is deployed on virtual layer.

3) Isolation and security. Container doesn't provide as strong a security boundary as a VM. Isolation capacity of secure container is between VM and container. If a functional component requires high security boundary, it should be deployed in VM. If a functional component requires quick boot, it could be deployed in container [6]. If a functional component requires both security boundary and quick boot, secure container may be the best choice.

Fig 4 displays five deployment modes, including bare-metal, container, secure container, and VM. With limited computing resource, all functional components prefer to work over virtualization technology to enhance the resource sharing. The 1st and 5th deployment mode only used in high performance and exclusive scenario. The 4th deployment mode is suitable for multi-tenant scenario by vGPU support.

In practice, AI system and big data system uses GPU and CPU, respectively. Each functional component of AI system works in an independent secure container (Kata) to ensure security and

agility, liking 3rd deployment mode. As shown by 2nd deployment mode, each functional component of big data system is deployed in an independent container to achieve higher performance and flexibility.

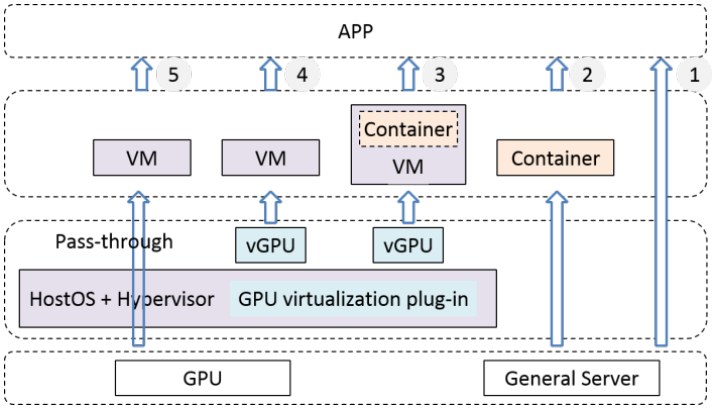


Figure 4. Functional component deployment mode.

4. Performance Evaluation

This section evaluates the performance of the proposed deployment modes by practical experiments and provides some recommendations.

4.1. Container on Bare-metal

To understand the performance impact of image recognition with different number of container, the experiment conducts two cases.

Case 1: Only one container is deployed on a bare-metal.

Case 2: Two containers are deployed on a bare-metal.

A physical GPU, Nvidia Tesla T4, is installed on the bare metal that has 8 CPUs, 60GB RAM, and 96GB SSD storage. The scheduling policy of GPU is best effort. The result shows that case 1 and case 2 require 1 ms and 1.77566 ms to recognize an image, respectively. The theoretical value of case 2 is 2 ms. From the comparison, we see that competition between containers on bare-metal contributes to improve the utilization of GPU.

4.2. Secure Container vs. VM

To compare the performance of secure container and VM, we test them with image classification. Both secure container and VM have one vGPU (a quarter of a physical GPU, Nvidia Tesla T4).

VM completes the task using 5.1656 ms, while secure container requires 5.1846 ms. The result indicates that the comparative performance of VM achieves slightly better performance than secure container by 0.369%.

4.3. Virtualization Overhead

To understand the overhead of GPU virtualization (using GPU, Nvidia Tesla T4), we illustrate it in table 1 by doing image classification in three cases.

Table 1. Delay/GPU utilization with different scenarios.

	Case 1	Case 2	Case 3
Bare-metal	0.98ms/90%	-	-
Secure Container	-	1.36ms/66%	1.91ms/96%

Case 1: Only one container is deployed on a bare-metal, the container monopolizes GPU.

Case 2: Only one secure container is deployed on a bare-metal, the secure container monopolizes the virtualized GPU.

Case 3: Two secure container are deployed on a bare-metal. These secure containers share the virtualized GPU. GPU adopts the scheduling policy of best effort.

Comparing the results of case 1 and case 2, we can conclude that virtualization has negative impact on utilization of GPU. From the results of case 1 and case 3, we can see that overhead of virtualization is about 3.796%, and the utilization of GPU sharply reaches to 96% which is higher than that of case 1. Above all, we can see that virtualization will introduce a slight overhead and improve sharing utilization of resource.

5. Conclusion

The social security is highly concerned by all sectors of society. Video service provides a foundation to building video surveillance system which is an important guarantee for social security. In order to achieve quick response, video surveillance system has try to minimize end-to-end data delivery delay and data processing latency. MEC is adopted to reduce end-to-end delivery delay by processing data nearby data source. Special hardware, GPU, is used to accelerate graphics processing. With limited resource in edge scenario, functional components share computing resource by virtualization technology.

Considering requirement in computing type, resource sharing and isolation, multiple deployment modes are proposed and tested. Although GPU virtualization introduces slightly overhead, it benefits the resource utilization which is very important in edge scenario with limited resource.

6. References

[1] https://www.macrotrends.net/countries/CHN/china/populationdensity.
[2] R. Buyya and S. N. Srirama, Smart Surveillance Video Stream Processing at the Edge for RealTime Human Objects Tracking, Fog and Edge Computing: Principles and Paradigms, Wiley, 2019, pp.319-346.
[3] M. Satyanarayanan, The Emergence of Edge Computing, Computer, vol. 50, no. 1, pp. 30-39, Jan. 2017.
[4] M. Xue et al., Scalable GPU Virtualization with Dynamic Sharing of Graphics Memory Space, IEEE Transactions on Parallel and Distributed Systems, vol. 29, no. 8, pp. 1823-1836, 1 Aug. 2018.
[5] W. Felter, A. Ferreira, R. Rajamony and J. Rubio, An updated performance comparison of virtual machines and Linux containers, 2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Philadelphia, PA, 2015, pp. 171-172.
[6] A. Lingayat, R. R. Badre and A. Kumar Gupta, Performance Evaluation for Deploying Docker Containers On Baremetal and Virtual Machine,2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 1019-1023.