Future Generation Computer Systems 106 (2020) 67–76

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

# Heterogeneous edge computing open platforms and tools for internet of things

Huansheng Ning [a,b,c,*], Yunfei Li [a,b,c], Feifei Shi [a,b,c], Laurence T. Yang [d]

[a] The School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[b] Beijing Engineering Research Center for Cyberspace Data Analysis and Applications, Beijing 100083, China
[c] Cybermatics and Cyberspace International Science and Technology Cooperation Base, Beijing 100083, China
[d] The Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada

ARTICLE INFO

*Article history:*
Received 15 May 2019
Received in revised form 28 November 2019
Accepted 25 December 2019
Available online 2 January 2020

*Keywords:*
Edge computing
Cloud computing
Platform
Internet of Things(ioT)
Architecture
Application

ABSTRACT

With the continuous development of Internet of Things (IoT) and the overwhelming explosion of Big Data, edge computing serves as an efficient computing mode for time stringent data processing, which can bypass the constraints of network bandwidth and delay, and has been one of the foundation of interconnected applications. Although edge computing has gradually become one of bridges between cloud computing centers and mobile terminals, the literature still lacks a thorough review on the recent advances in edge computing platforms. In this paper, we firstly introduce the definition of edge computing and advantages of edge computing platform. And then, we summarize the key technologies of constructing an edge computing platform, and propose a general framework for edge computing platform. The role of distributed storage management systems in building edge computing platform is elaborated in detail. Furthermore, we give some applications to illustrate how to use third-party edge computing platforms to build specific applications. Finally, we briefly outline current open issues of edge computing platform based on our literature survey.



## 1. Introduction

With the explosion of Internet of Things (IoT), mobile devices and terminals coupled with huge amount of users produce large volume of data, which has become a typical feature in the era of big data. As we have seen, the speed of data growth has far exceeded the load limitations of network bandwidth. Despite the strong processing power and storage capacity of cloud computing, there still exists defects in the mode of centralized data processing. Firstly, the traditional centralized processing mode is to upload the massive data collected from terminals to the central cloud server for unified processing, and then transmit the processed results or feedbacks to the terminals. Nevertheless with the emerging interconnections of all things [1], bottleneck of network bandwidth and delay is greatly highlighted [2]. Therefore, we should pay attention to not only the delay of large-scale data transmission, but also the demanding requirements for the processing and load capacity of central cloud servers. Secondly, time-sensitive applications have higher strict for latency, such as interactive voice systems in smart home and low-latency feedbacks upon user actions in VR games. In addition, the traditional centralized mode rarely takes data storage environment into account especially under offline circumstances.

Therefore, researchers and engineers have proposed a model for deploying data storage and computing at the edge, which is named as MEC (mobile edge computing [3]). As a matter of fact, MEC is located between the mobile devices and the cloud computing center, and provides storage, pre-computation and even complete computing services. It is similar to the architecture of edge computing server and model [4]. Actually, MEC can be regarded as a scenario where a large number of heterogeneous devices could communicate and collaborate with each other to deal with storage and process tasks [5]. In addition, MEC can use the wireless access network to provide functions and computing services from cloud required by users, creating a service environment with high performance, low latency and high bandwidth. Thus the download of various content and applications in the network will be speeded up and consumers can enjoy an uninterrupted and high quality web experience.

Studies usually focus more on concepts and characteristics of edge computing, while in our paper the main contribution is to research applications and frameworks of edge computing platforms. The contents of each chapter are arranged as follows. Section 2 outlines the definition of edge computing. Section 3 briefly

* Corresponding author at: The School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China.
*E-mail addresses:* ninghuansheng@ustb.edu.cn (H. Ning), g20188802@xs.ustb.edu.cn (Y. Li), msshifeifei@gmail.com (F. Shi), lyang@stfx.ca (L.T. Yang).

https://doi.org/10.1016/j.future.2019.12.036
0167-739X/

introduces edge computing platforms and analyzes the commonalities of these platforms. Besides, the framework of building edge computing platforms is summarized from the aspects of actual production and specific applications, and the core technologies involved in building edge computing platforms are proposed. Furthermore, we discuss distributed storage management systems serve as core functional modules in edge computing platform specifically. Section 4 lists some specific application cases and schemes to construct application by using the third-party edge computing platform. Section 5 points out challenges and open issues of edge computing platform. Finally, we conclude this paper in Section 6.

## 2. Definition of edge computing

ETSI (European Telecommunications Standards Institute) defines MEC as follows: Mobile edge computing provides Internet service environments and cloud computing functions at the edge of mobile networks, near radio access networks (RANs) and mobile users [6]. Actually, the core concept of MEC and edge computing is same: the location of computing is close to the source of data and closer to users. From the perspective of bionics, it is easier to describe the definition of edge computing: octopus makes full use of its tentacles when hunting, and each tentacle cooperates synchronously, never knotting. Reasons for this phenomenon are mainly in that the octopus' nervous system is similar to a simple distributed system: a brain in the center is responsible for scheduling, and multiple cerebellums are located at the tentacles for coordination. Edge computing is also a kind of distributed computing pattern: collected data can be processed at the smart gateway which is closer to users, without uploading a large amount of data to the cloud center.

The MEC is located between the cloud layer and the mobile devices layer [7]. It is a three-layer hierarchy of cloud-MEC-mobile devices [8]. MEC emphasizes edge servers complete computing tasks, while mobile devices in MEC are nearly considered to have no computational power. In contrast, terminal devices in the edge computing model usually have strong computing power.

Besides MEC, there exist other models proposed in recent years, for example, Mobile Cloud Computing (MCC). Actually, MCC, as its name implies, refers to cloud computing on the mobile side provides information services for electronic devices [9], such as mobile phones, laptops and so on. It is not only confined to person computer, but can also be acquired through mobile network links according to their own needs [10]. Table 1 [11] shows the comparison between MEC, MCC and edge computing.

Before edge computing proposed, the concept of fog computing has been put forward. The main goal of edge computing and fog computing is similar in that they both bring some of the capabilities of cloud computing to the edge of the network. Fog computing, edge computing and other computing models have been proposed as an extension of cloud computing, rather than a replacement of cloud computing. Even with the same goal, there are still some potential differences between edge computing and fog computing. Table 2 [12–15] shows the comparison between cloud computing, fog computing and edge computing. In a word, cloud computing, fog computing, edge computing and other computing models work together to form a computing model beneficial to each other. This new computing model can be better adapted to various application scenarios of Internet of Things.

Edge computing refers to a new type of computing mode that performs computing at the edge of the network. The specific data process consists of two parts: IoT services delivered upward and cloud services that passes down. The "edge" in edge computing is relative to cloud center computing, specifically referring to resources on the path from cloud to edge, including computing, storage, and routing, etc. And the edge can be regarded as one or more resource nodes on the path according to the specific needs of applications and actual scenarios.

**Table 1**
Comparison of MEC, MCC and Edge computing.

| | MEC | MCC | Edge computing |
|---|---|---|---|
| Deployment location | Between terminals and data center, it can be co-addressed with gateways | Between terminals and data center, it can be co-addressed with gateways | Located at the edge of network's edge devices |
| Real-time interaction | Provides mobility management only in the case of a terminal moving from one edge node to another | Provides support for cloud image switching from one edge node to another | Fully supports communication between edge node distributed applications |
| Application scenarios | Suitable for IoT and IoV scenarios | Mobile enhanced Apps and IoT scenarios devices | IoT scenarios that require distributed computing and storage |
| Architecture | Three-layer hierarchy of cloud-MEC-mobile devices | Three-layer hierarchy of cloud-MCC-mobile devices | A continuum from data source to cloud center |
| Computational power of devices | Nearly no computing power | Strong computing power | Strong computing power |
| Nature | Server | Small data center | Computing pattern |

**Table 2**
Comparison between cloud computing, fog computing and edge computing.

| | Cloud computing | Fog computing | Edge computing |
|---|---|---|---|
| Location of data collection, processing, storage | Cluster of data center servers hosted on the internet | Near-edge and core networking, network edge devices and core networking devices | Network edge, edge devices |
| Computing power | Strong (depend on server cluster) | Weak (depend on network edge device network) | Common (depend on edge device) |
| Responsible for the type of task | Large computation, or long-term storage task | Preprocessing | Real-time processing |
| Focus | Clusters level | Infrastructures level | Things level |
| Handling multiple IoT applications | Supported | Supported | Unsupported |
| Resource contention | Slight | Slight | Serious |

The generation of edge computing is due to the increase of data volume and the need for real-time data processing, which means edge computing is a favorable complement and effective extension of cloud computing. Cloud computing needs edge computing nodes to pre-process data, while edge computing needs massive storage and strong computing capacity of cloud computing as its support to break the bottleneck of network bandwidth and latency. In the traditional cloud-centric processing framework, cloud-centric nodes have powerful computing power and nearly unlimited storage capacity, but they cannot provide context-aware and low-latency computing and processing, while edge computing nodes have these capabilities. Edge computing nodes and cloud computing center implement intermittent data transmission and update data by wireless network, which greatly reduces the data transmission cost from devices to the cloud. Compared to cloud computing, edge computing can better support IoT applications:

- Relieve pressure of network bandwidth and data center: it is the development of IoT that leads to global network access devices generate more than 847ZB of data in 2021 [16]. However, only a small amount of data contains real information, and most of the data is temporary or even useless. These temporary data do not need to be stored for a long time. Actually, the amount of data generated by the devices as producer is two orders of magnitude higher than the amount of data used as consumer. Edge computing can process large amounts of temporary data at the edge of the network, and then upload the processed data.
- Enhance the response speed of user request services: mobile devices are relatively weak in computing and storage capabilities, while cloud computing can serve as support services for mobile devices in both aspects. However, the speed of network transmission is easily affected by many factors, such as the bottleneck of the development of communication technology, the instability of network links and forwarding routes in complex environments. Therefore, the response speed of cloud services is greatly constrained. Edge computing provides close-range services near users, so it can guarantee lower network latency and enhance the response speed of users' requests services.
- Improve the users' authority to control private data: in cloud computing mode, all data and applications are managed in the data center, and users do not have enough authority for access. Edge computing can not only provide infrastructure for user access, but localize the operation of private data.

Edge computing platform introduced in this paper is aimed to deal with heterogeneous data generated by access devices, which is essentially an acceleration scheme similar to the combination of software and hardware, and also a important medium for data interaction between heterogeneous IoT platforms.

## 3. Framework of edge computing platforms

Users can run local computing, message forwarding, data caching, synchronization and other various services on edge devices safely and conveniently through edge computing platform.

### 3.1. Existing edge computing platforms

According to their object-oriented characteristics, we classify edge computing platforms on the market into the following two categories: edge computing platform for internet users and Industrial Internet of Things (IIoT).

#### 3.1.1. Edge computing platform for internet users

Edge computing platforms for internet users include AWS Greengrass, Microsoft Azure IoT, Baidu BIE, etc. These open platforms are mainly based on the advantages of their own operators, and their main business targets are internet users.

AWS Greengrass [17] is a representative edge computing platform in this category. AWS Greengrass is an edge computing platform from Amazon with services of native computing, messaging, data caching, synchronization and ML Inference for connected devices of users. As a matter of fact, ML Inference is a module that includes model built and trained by machine learning reasoning in the cloud computing center. AWS Greengrass mainly consists of AWS Lambda and AWS IoT Core. AWS Lambda is a function module related to function computing, and AWS IoT Core is a cloud computing processing center. AWS Greengrass inherits the functions of "Device Shadow" in previous product AWS IoT: "Device Shadow" acts as virtual backup of real devices, caches the state of local devices, tracks and records the state of edge devices, and synchronizes state information to cloud computing centers when transferring data to the cloud. With AWS Greengrass, edge devices can run reliably and safely even when they are offline or only intermittently connected to the cloud. After the device is reconnected, AWS Greengrass device management automatically synchronizes the latest status of the device to ensure seamless operation. In fact, only a small part of the massive data acquired through IoT devices is meaningful for subsequent analysis. By using services such as AWS Greengrass flow analysis or stereotyped models to process data locally, only the required data is sent to the cloud for subsequent analysis. This scheme avoids the expensive cost of sending all data to the cloud, and ensuring high quality of data transmission.

The most striking of AWS Greengrass is the machine learning inference module. AWS Greengrass can be used to deploy complex event processing, machine learning, image recognition and other high-value artificial intelligence without writing code internally. Users can run Azure services such as Functions, flow analysis and machine learning locally. The cloud-built machine learning module can be quickly deployed on local edge devices, so that the trained machine learning program does not incur additional data transfer costs, while controlling latency and ensuring real-time performance well. In addition, as Amazon's industrial ecosphere continues to expand gradually, many ideas have become reality and used in various industrial parks, this advantage also lays the foundation for users to provide more reasonable and efficient edge computing solutions.

#### 3.1.2. Edge computing platform for IIoT

GE, Siemens, ABB, Schneider, Bosch and other industrial automation giants develop corresponding edge computing platforms for Industrial Internet of Things. The business objects of these open edge computing platforms are mainly industrial equipment providers and service providers cooperating with their enterprises.

Edge computing is a critical part of IIoT solutions. GE Predix [18] is the first edge computing platform on the market, and it is a scalable edge computing platform for IIoT. As an extensible edge computing platform for IIoT, Predix Edge provides powerful connection and management capabilities to meet any edge computing needs. Predix Edge delivers connectivity, applications, and intelligence from plant floor to data center by supporting deployment options from small embedded gateways to servers. Industrial equipment providers and service providers can use Predix Edge to enhance the functionality and usability of their devices. In addition, GE has also launched Predix and associated products, such as Predix Cloud and Predix Private Cloud. These products can provide a model of deployment, asset performance management and other functions to optimize the performance of capital.

### 3.2. Key technologies for building edge computing platforms

Through the comparison of different edge computing platforms, it is found that most platforms adopt the following technologies and tools to realize computing migration, resource discovery, management and other platform-level functions.

#### 3.2.1. Container technology and tools

Docker [19] is the most representative container technology, which decouples from the host through container isolation, so user service doesn't need to rely on a specific host. Mainstream edge computing platforms currently regard containerization as their major strength: Docker's merit lies in cross-platform support and low standardization requirements of platform environment. Docker isolates resources and precisely allocates CPU, memory, and other resources for each running instances to improve resource utilization. Actually, this feature meets the requirement of "one-button deployment" of edge computing platform on various operating systems.

Kubernetes [20] is a tool for organizing containers that manages the entire life cycle of applications, with the abilities to create clusters and deploy, publish, extend, update various applications. Kubernetes and OpenStack host VM-based applications and container-based applications. Kubernetes is commonly used in open source platforms for automated deployment (e.g. HELM), expansion and maintenance of container clusters. By using Kubernetes, users can be responded quickly and effectively of their needs, and private applications can be deployed quickly and expectantly. In addition, Kubernetes has the function of expanding applications rapidly and joining new applications seamlessly, which optimizes the resources usage greatly.

#### 3.2.2. Tools to deploy and maintain cloud platforms

OpenStack provides an operating toolset for clouds deployment. It aims to help users in running virtual computing or storage services, providing scalable and flexible cloud computing for public and private clouds, as well as big or small clouds [2]. OpenStack is an open source project, which is maintained by community, including a components collection of OpenStack Computing (Nova), OpenStack Object Storage (Swift) and OpenStack Mirror Service (Glance).

Keystone is a basic service that supports OpenStack, its main functions are to manage users rights and maintain endpoints, authentication, authorization [21,22] of OpenStack services. In general, there are many edge computing platforms use KOLLA as their assistive tool to deploy OpenStack, in this way the functional components could be modularized, so as to reduce the coupling of sub-functional modules. OpenStack not only meets the requirements of modularization of edge computing platforms, but meets the needs of "use on demand, deployment on demand".

#### 3.2.3. Virtualization technology

The most representative virtualization technology is NFV [23] (network functions virtualization) in edge computing platform. Its principle is as follows: Heterogeneous network devices (e.g. servers, storages and switches) are constructed into a data center network firstly, and then virtualized them into a virtual machine, deploy the specific business on this virtual machine finally. NFV redefines the architecture of network devices essentially. For example, computing virtualization in NFV is a virtual machine that creates multiple virtual systems on a single server. Storage virtualization in NFV refers to the virtualization of multiple storage devices into a logical storage device. Network virtualization, that is, control plane of the network device is separated from underlying hardware. The control plane of the device is installed on the server virtual machine, so various service softwares can be installed at the virtualized device level.

The control page or plane of the device is separated from the specific device by the NFV: the control plane of the different devices is based on the virtual machine, and the virtual machine is based on the cloud operating system. In this way, when deploying a new service, it is only to install a software package of the necessary functional modules in the virtual machine to iteratively update, thereby avoiding duplication of investment.

#### 3.2.4. Distributed storage management system

In the architecture of edge computing platform, functional components of middle layer are based on basic modules of the bottom layer to provide services for the higher functional components. Among them, distributed storage management system is located on the bottom of the distributed operating system, which is the core open source component of the entire edge computing platform, and also a basis to provide independent solutions for practical application needs of edge computing platform [24]. Starting from the distributed storage management system in the framework of edge computing platform, we illustrate the scheme of constructing an independently customized edge computing platform. We divide the existing distributed storage systems into the following categories:

- Edge Computing Platform for Big Data: Distributed file systems for big data mainly include Hadoop HDFS [25] and IBM GPFS [26]. The former is suitable for high-data throughput access to large data sets in the commercial field, and has the characteristics of scalability, efficiency and reliability. The core of Hadoop is that this system adopts Hadoop MapReduce, which can migrate processing tasks to physical nodes without migrating data and reduce network I/O latency. The latter is a proprietary high-performance cluster file system developed by IBM, which can be deployed in shared disk or non-shared distributed parallel mode. Both distributed file storage systems are used in collaboration with their large data processing architectures, and can be applied to the construction of personal or business edge computing platforms.
- Edge Computing Platform for Pattern Recognition and Image Processing: Facebook Haystack adopts storing each image as a file to replace the NFS file system. It is suitable for object storage of images and widely used in pattern recognition and image processing [27]. Facebook Haystack focuses on long tail services and reduces the effective replication factor from 3.6 to 2.8 or 2.1 [28]. Its characters of low latency, disk recoverability and high throughput also guarantee the construction of edge computing platform for pattern recognition and image processing.
- Edge Computing Platform for Large Scale Cluster: Lustre [29] provides high-performing file management for heterogeneous computer clusters with high performance and open authentication, so it is usually used in supercomputers. Weather, virtualization, oil and gas, life sciences, multifunctional media and financial industries can apply Lustre as their file management module to build edge computing platform.
- Edge Computing Platform for Multi-integration: Ceph, as an open source software, has great scalability and there are many new commercial servers developed based on Ceph [30,31]. Ceph supports object storage, block storage and file storage at the same time on a single distributed node [32]. Ceph nodes are supported by common nodes and intelligent daemons. Ceph storage cluster organizes a large number of nodes, which is a good choice for individuals or enterprises to build edge computing platform.
- Edge Computing Platform for Virtual Block Storage: OpenStack Cinder [33], Amazon EBS [34], Azure [35], Google Persistent Storage [36] and Cluster FS all provide block storage as a Linux-accessed file system: virtualized block storage device pools provide self-service APIs to users who need and consume those resources without understanding the actual location of those storage deployment or types of storage devices. These are also the commonly adopted distributed file system for the edge computing platform on the market.

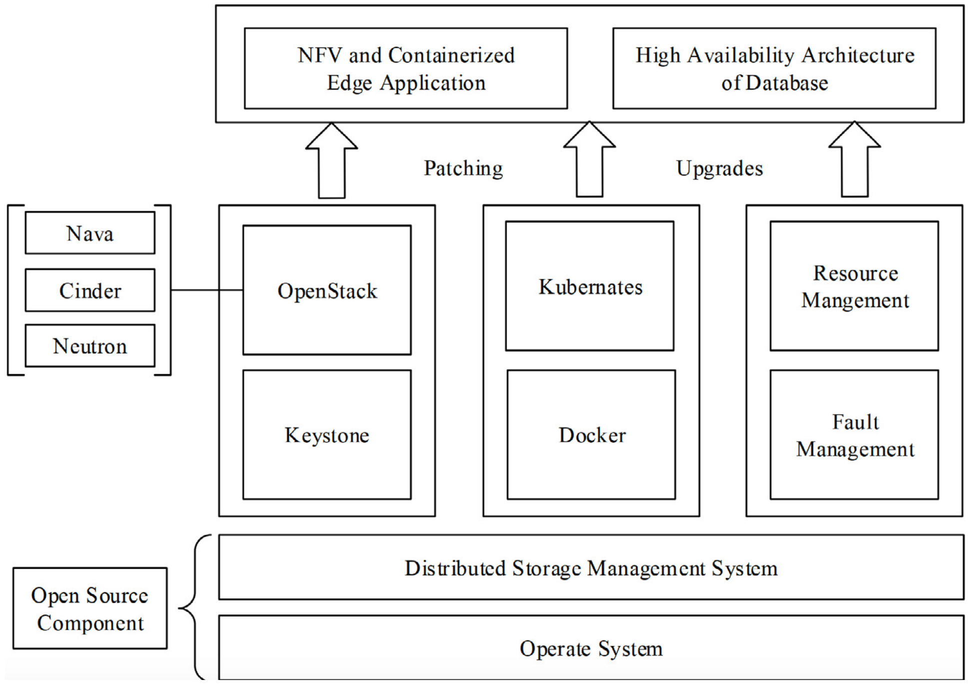


**Fig. 1.** Framework of edge computing platform.

Actually, the key technologies involved in building edge computing platforms are not static, such as docker, which can be replaced with other containerization technologies. For example, containerization technology can also be used to realize cross-platform migration with virtualization technology. Similarly, there are many options for managing virtualization platform technology. For example, Libvirt can provide a convenient way to manage the collection of virtual machines and other virtualization functions of software to realize storage and network interface management.

### *3.3. Framework of edge computing platform*

After a detailed study of popular edge computing platforms and pertinent technologies, we summarized the basic framework of building an edge computing platform in Fig. 1.

The whole framework can be divided into five parts:

- Open Source Components: distributed operating system (e.g. Linux) and distributed storage management system constitute open source components.
- Platform Deployment Components: including Keystone and containerized OpenStack (includes Nova, Cinder and Neutron). Keystone and OpenStack are responsible for the layout and deployment of cloud platform, as well as supporting modularization of edge platform components.
- Cross-platform Support Components: provides various real-time conversion of running services, so functional models written in arbitrary languages can be compatible in different develop environment.
- Edge Computing Enhancement Components: including system management, fault management and resource management, etc.;
- Edge Computing Application Components: including database based on high availability architecture, NFV and container edge computing applications.

In the framework of computing platform above, open source components is the basis of the entire edge computing framework, on which the cost performance and response time of the edge computing platform are better than the mainframe system, and support the collaborative requirements of scattered edge users; In the platform deployment components, Keystone and OpenStack are responsible for the layout and deployment of cloud platform, as well as supporting modularization of edge platform components. Open source components and platform deployment components coordinate and adapt to various application scenarios; Open source components and platform deployment components coordinate to support rapid deployment and seamless upgrade of edge applications; Edge Computing Enhancement Components and Edge Computing application Components support the whole life cycle management of edge applications.

In the framework of the entire edge computing platform, the five layers of components work together to realize the end-edge-cloud coordination.

## 4. Applications of edge computing platform

### *4.1. Construction of edge applications*

The prominent feature of edge computing nodes is that they are closer to devices, but the applications of edge computing close to devices is quite different from that of traditional applications. The following is a reference of the third-party edge computing platform for users to build user-defined edge computing applications.

#### *4.1.1. Microsoft Azure IoT*

Microsoft Azure IoT is an open edge computing platform on where enterprises can build their own cloud services and develop edge applications. As well as they can improve the concurrent processing ability and data maintenance ability of their systems by using the data service function of Azure. Here is an typical case combining with machine vision, showing how to use Azure to build an edge computing application for enterprises.

Azure provides PaaS and SaaS services [35]. Users can send data produced by authenticated terminals to the edge(an embedded edge computing system and located in the camera essentially) of Azure through network, such as adding cameras and

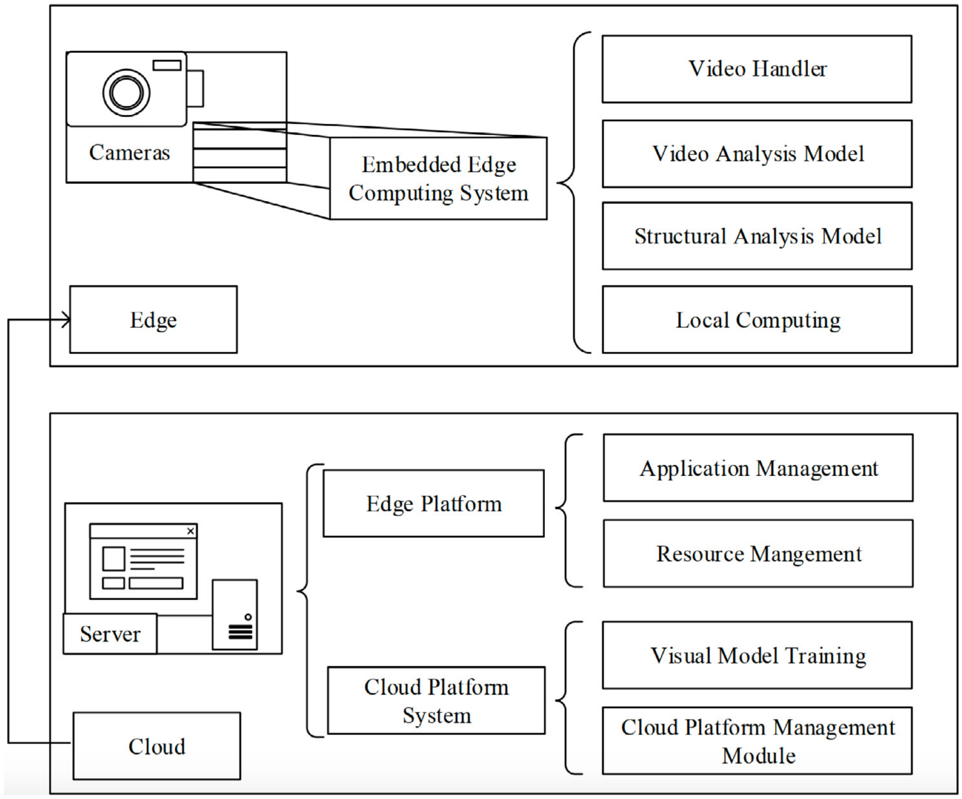


**Fig. 2.** Precise medication.

other access equipments. The data will be pre-processed at the edge, and then uploaded to the platform on the cloud. This system is shown in Fig. 2.

Azure IoT Edge consists of three modules: cloud platform, embedded edge computing system and access devices. Embedded edge computing system carries functions of data acquisition, data synchronization, local computing and configuration synchronization. Embedded system in edge machine vision mainly includes video processing function, video analysis model and result analysis function. Video analysis model here is derived from cloud training. Access devices are cluster composed of multiple cameras, responsible for collecting data such as pictures and videos. Azure is a third-party cloud platform with high performance computing capability, carrying the machine training in the system. The machine vision model for cloud-based training can be applied to face recognition, industrial quality inspection, urban management and public safety, etc. Azure provides services to deploy these visual models to local devices, providing fast recognition response on the one hand, and reducing the cost of video or picture transmission bandwidth on the other hand.

*4.1.2. Linux foundation edge Baetyl*

Baetyl [37] is an open edge computing framework for the Linux Foundation edge, originating from the baidu open source project. The combination of Baetyl and BIE(Baidu Intelligent Edge [38]) cloud management suite will enable cloud management and application distribution, enabling applications to run on edge devices to meet a variety of edge computing scenarios. During the design of architectures, Baetyl adopts a modular and containerized design pattern where Baetyl divides the product into modules and ensures that each module is a separate one. In addition, Baetyl employs a containerized design pattern to construct system images. Due to the cross-platform nature of docker, the operating environment of each operating system is guaranteed to be consistent. Besides, Baetyl isolates and limits the containers' resources, and accurately allocates CPU, memory, and other resources for each running instance to improve resource utilization. In general, Baetyl can fully meet the conscious needs of users to deploy on demand.

A complete Baetyl system consists of Master, Service, Volume and system resources. Among them, Master is the core part of Baetyl that manages Volume and Service, built-in Engine, external RESTful API, and the command line. Service is a set of running programs managed by Baetyl that provide specific functions, such as message routing services, functional computing services, micro-services, and so on. Instance refers to the specific runtime or container that is started by Service. A Service can start multiple instances, or be started dynamically by other services. For example, instances of a runtime function service are dynamically started and stopped by a function manager service. Volume is the directory used by Service, which can be a read-only directory, such as a directory for placing resources including configuration, certificates, scripts, and so on, or a writeable directory for persisting data, such as logs and databases. Engine refers to the operational abstraction and concrete implementation of various operational modes of Service, such as docker container mode and native process mode.

Edge computing platforms, such as Azure and Baetyl, rely on the advantages of those own operators, so users need not worry about bandwidth and computing power of cloud platform. Uncontrollable data, such as user data and control instructions, can be handed over to the platform. Faced with the different edge needs of different users, Azure has successfully cooperated with Schneider Electric, Tetra Park, Drone Works, Rockwell Automation and other large companies to produce a series of solutions for agriculture, pattern recognition, unmanned aerial vehicles maintenance, oil and gas, etc. Users can adopt these solutions to deploy edge computing applications that meet their own needs.

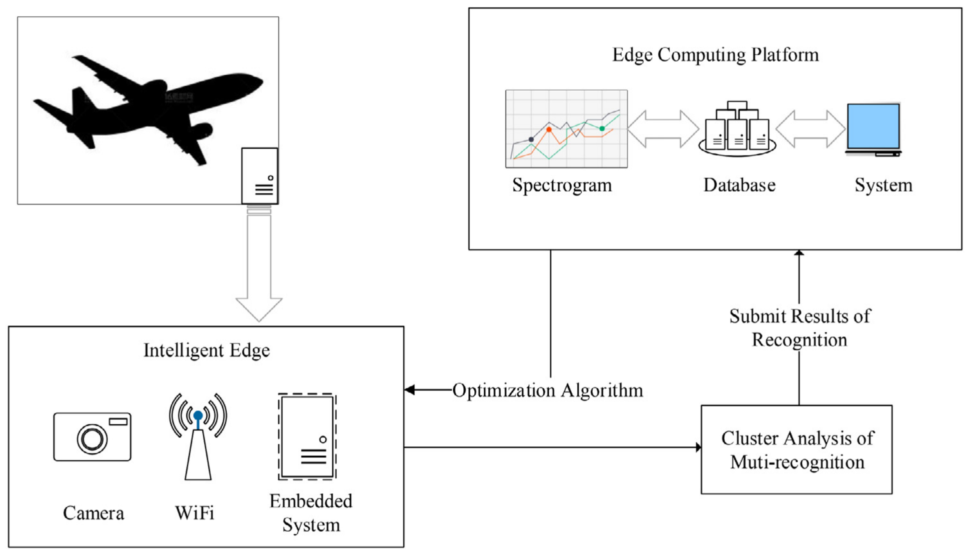


**Fig. 3.** Precise agriculture medication.

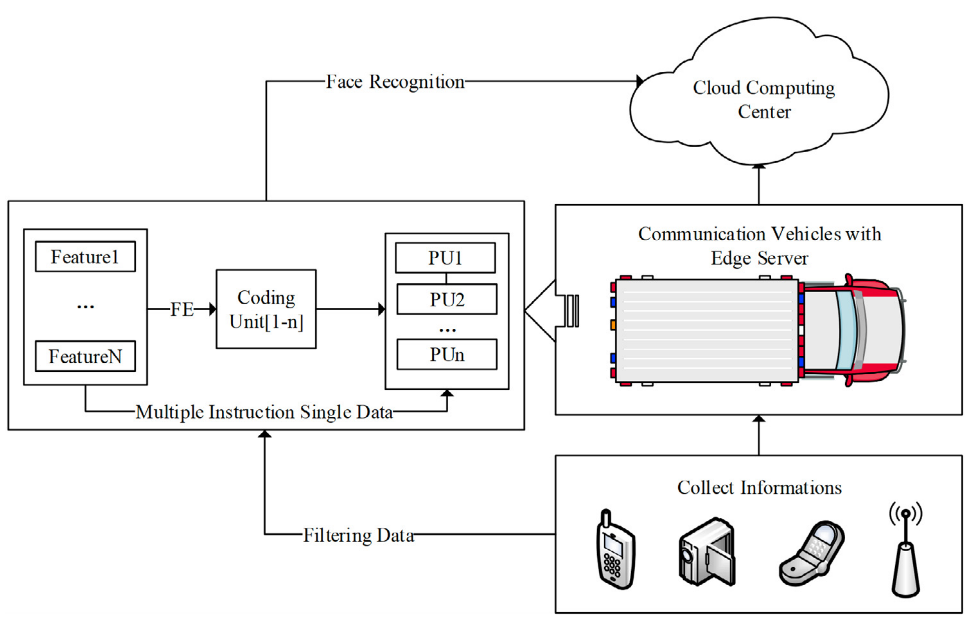


**Fig. 4.** Image search framework based on edge computing.

*4.2. Specific applications*

It is a basic task for edge computing platform to support applications, so we introduce some cases of the combination of edge computing platform and specific practice.

*4.2.1. Pesticide spraying by UAV and precise medication in agriculture*

Pests and diseases are the most direct factors related to crop yield and farmers income in agricultural. Due to the features of sudden outbreak and huge destruction, pests and diseases have become one of the major agricultural disasters all over the world. In addition, widely used and highly efficient methods are spraying pesticides in large areas, and nowadays pesticide spraying on large areas of crops is basically carried out by modern unmanned aerial vehicles, which takes effect and saves time. However, the large-scale indiscriminate spraying of pesticides has great shortcomings: on one hand, the cost of grain planting increases and farmers income is affected; on the other hand, excessive spraying of pesticides leads to pesticide residues, which does not meet the requirements of safety and health [39].

Therefore, it is necessary to combine agricultural with edge computing: the intelligent edge computing module, which is essentially an embedded processing unit, is deployed on unmanned aerial vehicles. And next, the infrared camera is deployed on the unmanned aerial vehicles to collect spectral information data of crops. Then the data would be uploaded to the artificial intelligence processing module of the edge computing platform. The real-time monitoring model is loaded to identify the location and severity of insect pests, the color and wilting degree of the disease. Finally, the module execute the pesticide spraying on unmanned aerial vehicles in real time by monitoring results. In this way, the detector has some intelligent functions to realize intelligent detection and precise medication. Fig. 3 shows the edge computing application for the entire precision agriculture.

#### 4.2.2. Disaster rescue and intelligent fire fighting based on edge computing

Once natural disasters such as earthquakes, fires, tsunamis and landslides erupt, they will not only engulf countless lives, but also bring substantial trauma and long-term mental disorders to the local people. Even if modern science and technologies develop rapidly, the occurrence of disasters cannot be accurately predicted. Therefore, when a disaster occurs, facing with complex and dangerous environment of the disaster scene, it is particularly significant for unmanned aerial vehicles, image sensor and other means to obtain real-time information of the disaster scene. Image and video information returned from the disaster scene could use to perceive and determine the extent of damage caused by the disaster, determine the distribution of the affected population and the congestion of roads. As fact, the local network environment has been destroyed: the wireless network is heavily congested, the fixed network is ineffective, and the network transmission environment is unstable. At this time, if traditional data processing method is adopted to upload all the collected information(without any processing) to the cloud computing center, the transmission condition of the network will be the biggest constraint. In the field of disaster relief, if the collected information cannot be quickly and accurately processed into seasonable and effective rescue programs, it will increase the possibility of secondary damage to the lives and property of the affected population.

Cloudlet proposes a solution based on edge computing for rapid resource awareness and disaster panoramic construction [40]. The scheme aims to overcome network congestion and low transmission quality in complex network environment, support delay-sensitive applications, and ultimately release the pressure of data center. Wayne State University scholars have presented a highly efficient image search framework in the event of disaster [41], which is mainly used for fast search of missing persons. The model is shown in Fig. 4.

Also in the field of fire protection, intelligent fire service is realized through the combination of fire field edge equipment and cloud. The edge devices, edge nodes in the fire scenes and cloud servers constitute the whole edge computing framework [42]. Edge nodes mainly include routers, base stations, switches and their corresponding data and computing nodes. These edge nodes can perform data processing and analysis for delay-sensitive applications. Cloud computing center servers are responsible for storing the past fire data and applications with low real-time requirements. The model is shown in Fig. 5.

## 5. Challenges and open issues

The edge computing platform provides an architecture and software support for edge-based computing applications that reduces latency and improves efficiency significantly. In fact, in edge computing practical applications, about 90 percent of edge complexity is software related. Many third-party edge computing platforms provide hardware support, but ignoring software is the key to distinguish the quality and complexity of edge computing. At present, the edge computing platform is still in a period of rapid development. The following points are our analysis of the future development trend about edge computing platform:

(1) The dedicated edge computing platform for specific application scenarios will grow rapidly. In addition to the specific application scenarios of several edge computing platforms proposed in Section 4.2, smart home, smart factory and smart city are currently the application scenarios of concern, and how to systematically support the functional requirements of these scenarios is one of the development trends of edge computing platforms.

(2) The data security and privacy protection issues of the edge computing platform will become more prominent. How to ensure the data security when the edge devices cooperate with each other, how to avoid exposing users' location privacy information, especially under the limited computing resources and storage capacity of edge computing devices, which makes the security defense of the edge computing platform more challenging.

(3) The edge computing platform will incorporate more artificial intelligence technologies. At present, the artificial intelligence technologies represented by deep learning are developing rapidly, but the research of "intelligent edge" has just started. The future edge computing platform will introduce artificial intelligence technologies to realize intelligent processing of edge data. For example, knowledge graph, a hot topic in both academia and industry, how to implement knowledge graph in the edge computing environment, is becoming a hot research direction and also the focus of our team's future research.

(4) Multiple edge platforms cooperate and eventually land on a comprehensive computing platform. When a large number of IoT access devices are involved, it is easy for edge computing platforms to complicate data storage. For example, when the data resides on one edge computing platform or migrates from one platform to another, what resident algorithm and data migration algorithm should be adopted to ensure the data will not be damaged. In addition, how to cooperate with each other to improve efficiency and maximize resource utilization among edge computing platforms has posed challenges to the platform and interface design of edge computing.

## 6. Conclusions

Edge computing has a great potential to manage millions of heterogeneous sensors and devices under the great developments of IoT. It is noteworthy that edge computing platform has become an indispensable part of edge computing, where users can deploy and migrate practical applications in different scenarios effectively. As a matter of fact, the idea of building an edge computing platform is not a fixed idea, but rather determines the type of application, the size of the data being processed, and so on. For the selection of scenarios and construction of edge computing platform, it must be based on actual productions and specific applications. Final solutions to build edge computing platform ultimately depends on the following points: the practical business requirements, the type of generated solutions, the skills the generated solutions needed to organized and the long-term maintenance of these solutions. This article does not elaborate on the specific aspects of the edge computing platform (e.g. hardware selection, network topology and security issues), but gives a detailed illustration of the functional components of each layer in the edge computing platform. Depending on the detailed classification of distributed storage management systems, we put forward personal opinions on the construction of edge computing platform. We also discuss how to build an edge computing application with the help of third-party open edge computing platform and cases of the combination of edge computing platform as well as the specific practice of the entire edge computing platform. In short, the following works mainly include task scheduling of end-edge-cloud coordinately and applications migration between different IoT platforms.

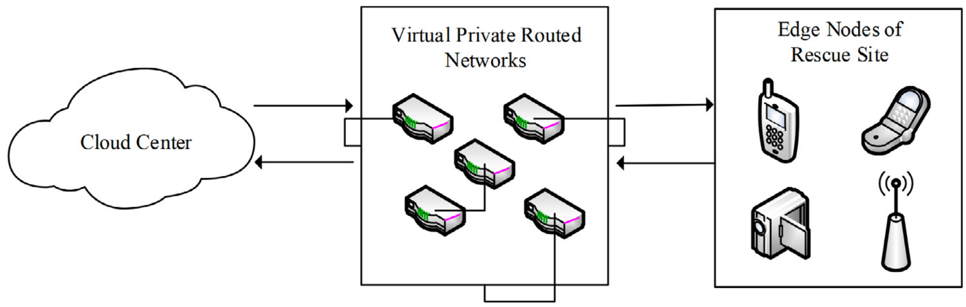


**Fig. 5.** Intelligent fire protection based on edge computing.

## 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.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61872038, 61811530335, and in part by the Fundamental Research Funds for the Central Universities under Grant FRF-BD-18-016A.

## References

[1] D.E. Culler, The once and future internet of everything, GetMobile: Mob. Comput. Commun. 20 (2017) 5–11.
[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, A view of cloud computing, Int. J. Comput. Technol. 4 (2013) 50–58.
[3] European telecommunication standards institute. executive briefing-mobile edge computing initiative, 2015, URL https://www.etsi.org/images/files/ETSIWhitePapers.
[4] W. Shi, F. Liu, H. Sun, Q. Pei, Edge Computing. Volume 1, Beijing Science Press, 2018.
[5] M. Patel, Mobile-Edge Computing—Introductory Technical White Paper, Mobile-Edge Computing (MEC) Industry Initiative, 2014.
[6] Yifan, Mobile edge computing towards 5g: Vision, recent progress, and open challenges, China Commun. 13 (2017) 89–99.
[7] Y. Jararweh, A. Doulat, O. Alqudah, E. Ahmed, M. Al-Ayyoub, E. Benkhelifa, The future of mobile cloud computing: Integrating cloudlets and mobile edge computing, in: International Conference on Telecommunications, 2016.
[8] T.H. Luan, L. Gao, L. Zhi, X. Yang, L. Sun, Fog computing: Focusing on mobile users at the edge, Comput. Sci. (2015).
[9] Fernando, Niroshinie, Loke, W. Seng, Rahayu, Wenny, Mobile cloud computing: A survey, Future Gener. Comput. Syst. 29 (2013) 84–106.
[10] M.R. Rahimi, R. Jian, H.L. Chi, A.V. Vasilakos, N. Venkatasubramanian, Mobile cloud computing: A survey, state of art and future directions, Mob. Netw. Appl. 19 (2014) 133–143.
[11] A.N. Khan, M.L.M. Kiah, S.U. Khan, S.A. Madani, Towards secure mobile cloud computing: A survey, Future Gener. Comput. Syst. 29 (2013) 1278–1299.
[12] P. Hu, S. Dhelim, H. Ning, T. Qiu, Survey on fog computing: architecture, key technologies, applications and open issues, J. Netw. Comput. Appl. 98 (2017) 27–42.
[13] S. Wang, Z. Xing, Z. Yan, W. Lin, W. Wang, A survey on mobile edge networks: Convergence of computing, caching and communications, IEEE Access 5 (2017) 6757–6779.
[14] R. Roman, J. Lopez, M. Mambo, Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges, Future Gener. Comput. Syst. 78 (2016) S0167739X16305635.
[15] E. Ahmed, M.H. Rehmani, Mobile edge computing: Opportunities, solutions, and challenges, Future Gener. Comput. Syst. 70 (2016).
[16] Cisco virtual networking:cisco globle cloud index:forecast and methodology 2015-2020 [eb/ol], 2019, URL https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html?dtid=osscdc000283.
[17] A. Kurniawan, Learning AWS IoT: Effectively Manage Connected Devices on the AWS Cloud using Services Such As AWS Greengrass, AWS Button, Predictive Analytics and Machine Learning, Packt Publishing Ltd, 2018.
[18] GE. Predix, URL https://www.ge.com/digital/iiot-platform/.
[19] Docker, URL https://www.docker.com.
[20] Kubernetes, URL https://kubernetes.io.
[21] K. Jackson, Openstack Cloud Computing Cookbook, Ehu Es, 2013.
[22] R.H. Khan, J. Ylitalo, A.S. Ahmed, Openid authentication as a service in openstack, in: International Conference on Information Assurance and Security, 2012.
[23] NFV, URL https://www.etsi.org/technologies/nfv.
[24] W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, X. Yang, A survey on the edge computing for the internet of things, IEEE Access 6 (2018) 6900–6919.
[25] K. Shvachko, S. Radia, A.L. Cox, The hadoop distributed file system, in: IEEE Symposium on Mass Storage Systems and Technologies, 2010.
[26] Marcelo Barrios, Terry Jones, Scott Kinnane, Mathis LandzettelSafran Al-Safran, Jerry Stevens, Christopher Stone, Chris Thomas, Ulf Troppens, Sizing and Tuning Gpfs, IBM Redbook, 1999.
[27] D. Beaver, S. Kumar, H.C. Li, J. Sobel, P. Vajgel, Finding a needle in haystack: Facebook's photo storage, in: Usenix Conference on Operating Systems Design and Implementation, 2010.
[28] S. Muralidhar, W. Lloyd, S. Roy, C. Hill, E. Lin, W. Liu, S. Pan, S. Shankar, V. Sivakumar, L. Tang, F4: Facebook's warm blob storage system, in: Usenix Conference on Operating Systems Design and Implementation, 2014.
[29] Lustre, 2019, URL http://wiki.lustre.org.
[30] S.A. Weil, S.A. Brandt, E.L. Miller, D.D.E. Long, C. Maltzahn, Ceph: A scalable, high-performance distributed file system. n/a, 2010, 307–320.
[31] S.A. Weil, S.A. Brandt, E.L. Miller, C. Maltzahn, Crush: Controlled, scalable, decentralized placement of replicated data, in: IEEE Sc Conference, 2006.
[32] Carlos Maltzahn, Esteban Molina-Estolano, Amandeep Khurana, Alex Nelson, Scott Brandt, Sage Weil, Ceph as a scalable alternative to the hadoop distributed file system, The USENIX Mag. 35 (2010) 38–49.
[33] T. Rosado, J. Bernardino, An overview of openstack architecture, in: Proceedings of the 18th International Database Engineering & Applications Symposium, ACM, 2014, pp. 366–367.
[34] A.E.C. Cloud, Amazon web services, 2011, (Retrieved November 9, 2011).
[35] Azure files, 2019, URL https://www.azure.cn.
[36] Google persiste storage, 2019, URL https://www.actifio.com/technology/integrations/google-cloud.
[37] Baetyl edge computing platform, 2019, URL https://baetyl.io/en/.
[38] Baidu Intelligent Edge, URL https://cloud.baidu.com/product/bie.html?track=cp:nsem|pf:pc|pp:nsem-chanpin-BIE|pu:BIE-pinpaici|ci:|kw:2143978.
[39] Diseases and Insect Pests, URL https://baike.baidu.com.
[40] K. Gai, M. Qiu, H. Zhao, L. Tao, Z. Zong, Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing, J. Netw. Comput. Appl. 59 (2016) 46–54.
[41] X. Wu, R. Dunne, Q. Zhang, W. Shi, Edge computing enabled smart firefighting: opportunities and challenges, in: Proceedings of the Fifth ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies, ACM, 2017, p. 11.
[42] S. Hui, L. Xu, W. Shi, Vu: Video usefulness and its application in large-scale video surveillance systems: An early experience, in: SmartIoT Workshop, 2017.

**Huansheng Ning** received his B.S. degree from Anhui University in 1996 and his Ph.D. degree from Beihang University in 2001. Now, he is a professor and vice dean of the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research focuses on the Internet of Things and general cyberspace. He is the founder of the Cyberspace and Cybermatics International Science and Technology Cooperation Base.

**Yunfei Li** receives his B.S. degree from Qingdao University in 2017. He is currently working toward the M.S. degree from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests include NLP and Edge computing.

**Feifei Shi** received her B.S. degree from China University of Petroleum and her M.S. degree from University of Science and Technology Beijing. She is currently a Ph.D. student in the School of Computer and Communication Engineering, University of Science and Technology Beijing. Her current research interests include Internet of Things and Artificial Intelligence.

**Laurence Tianruo Yang** received PhD degree in computer science from the University of Victoria, Canada. He is a profess or in the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous computing, and big data. His research had been supported by the National Sciences and Engineering Research Council and Canada Foundation for Innovation. E-mail: lyang@stfx.ca