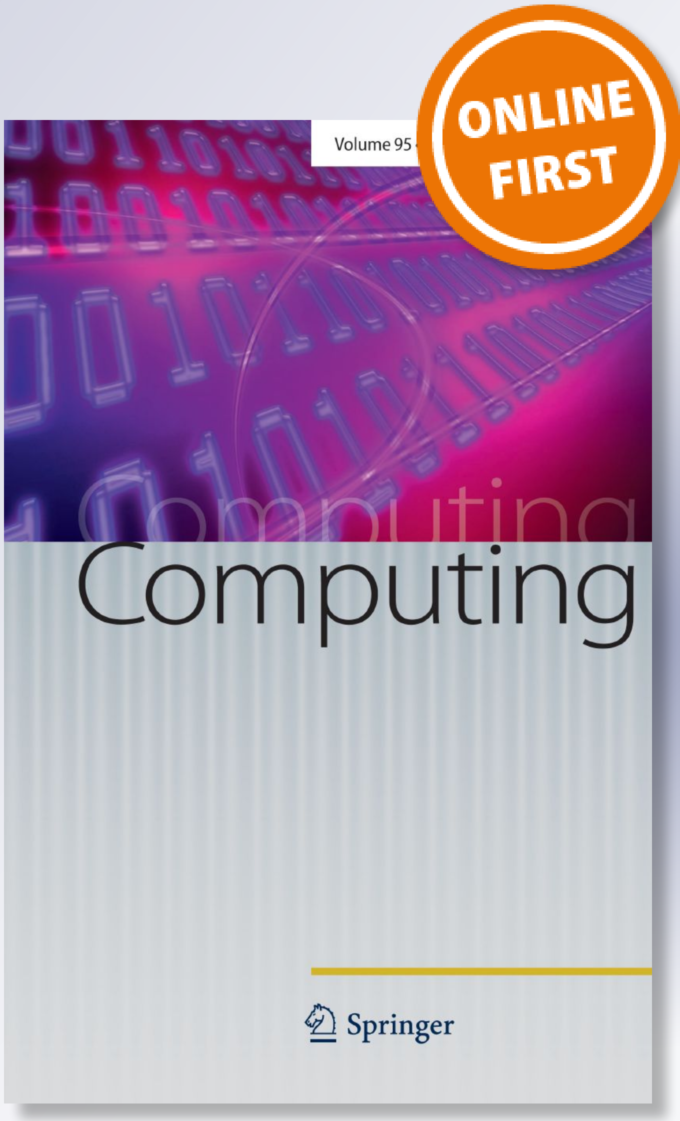
Edge-based personal computing services: fall detection as a pilot study

Lingmei Ren, Qingyang Zhang, Weisong Shi & Yanjun Peng

Computing

ISSN 0010-485X

Computing
DOI 10.1007/s00607-018-00697-x



Springer

Computing
https://doi.org/10.1007/s00607-018-00697-x

Edge-based personal computing services: fall detection as a pilot study

Lingmei Ren[1] · Qingyang Zhang[2] · Weisong Shi[3] · Yanjun Peng[4]

Received: 15 August 2018 / Accepted: 22 December 2018

Abstract
Current developments in information and electronic technologies have pushed a tremendous amount of applications to meet the demands of personal computing services. Various kinds of smart devices have been launched and applied in our daily lives to provide services for individuals; however, the existing computing frameworks including local silo-based and cloud-based architectures, are not quite fit for personal computing services. Meanwhile, personal computing applications exhibit special features, they are latency-sensitive, energy efficient, highly reliable, mobile, etc, which further indicates that a new computing architecture is urgently needed to support such services. Thanks to the emerging edge computing paradigm, we were inspired to apply the distributed cooperative computing idea at the data source, which perfectly solves issues occurring among existing computing paradigms while meeting the requirements of personal computing services. Therefore, we explore personal computing services utilizing the edge computing paradigm, discuss the overall edge-based system architecture for personal computing services, and design the conceptual framework for an edge-based personal computing system. We analyze the functionalities in detail. To validate the feasibility of the proposed architecture, a fall detection application is simulated in our preliminary evaluation as an example service in which three Support Vector Machine based fall detection algorithms with different kernel functions are implemented. Experimental results show edge computing architecture can improve the performance of the system in terms of total latency, with about 22.75% reduction on average in the case of applying 4G at the second hop even when the data and computing stream of the application is small.

Keywords Edge computing · Personal computing service · Fall detection · Computing paradigm

Mathematics Subject Classification 68M14

✉ Lingmei Ren
renlingmei11@163.com

Extended author information available on the last page of the article

Published online: 03 January 2019

1 Introduction

In the rapid development of the information era, the Internet of Things (IoT), a structured framework whereby various physical objects are connected to the internet, has revolutionized and made a qualitative leap in many application services. It changes our lives and work by connecting everything around us. It has also gained wide acceptance and adoption in many aspects, not only including home appliances, environmental protection, public security, intelligent transportation, industry automation, but also referring to various personal computing services. Generally, IoT delivers positive effects throughout almost all aspects of the society.

Personal computing services represent one of the most attractive and common application fields for IoT, involving a broad range of applications for personal smart healthcare, such as rehabilitation training, chronic disease monitoring, health monitoring and elderly care, etc [1,2], as well as other personal services, e.g. physical training, sports monitoring, entertainment, navigation and other applications that can serve individuals. Taking smart health-care monitoring as an example, it is one of the most common application areas of personal computing service. Most of these applications employ smart wearable, implantable, and/or embedded devices that can sense and collect the physiology data of the person to monitor their health in their everyday lives. Therefore, various smart devices can be viewed as the core part of these services. According to a recent report [3], each person is likely to have an average of 5.1 connected devices by 2020, while those devices will permeate different industries for different personal computing purposes to serve mankind. Furthermore, it is forecasted that the total number of smart health-care devices will reach to 808.9 million in 2020 [4,5]. However, no matter how many smart devices there are, it is expected that the service requester (individuals who launch the service request) can be monitored and served continuously and efficiently. Nevertheless, low-level smart devices always have limited computing and storage resources, as well as battery capacity, but they are required to be more lightweight with longer usage time to provide a super user experience [6]. Obviously, only one device used in existing systems cannot support the goal of real "smart" or "intelligent". To deal with the limitations and contradictions mentioned above, many personal computing services leverage IoT architecture to integrate all related data to achieve "smart". Two paradigms have been used widely, including the local silo-based method and the centralized cloud computing solution as shown in Fig. 1. For local silo-based system, low-level end devices working with a local server or not respond for all the computing tasks, such as sensing, processing and even action. However, the existing local silo-based system always works in a silo-based manner without communication or connection with others for comprehensive personal services. On the other hand, for the cloud computing-based framework, various low-level end devices only respond to signal sensing and simple preprocessing, while the centralized cloud server responds for most of the computing tasks in which more complex computations and storage can be done. Related action instructions or alarms will serve as feedback to guide actuators accordingly in case of an emergency. Generally, they apply a centralization idea for processing purposes, utilizing the abundant computing and storage resources of the cloud. In such a centralized computing framework, most of the computing and storage tasks are placed in the cloud. However,

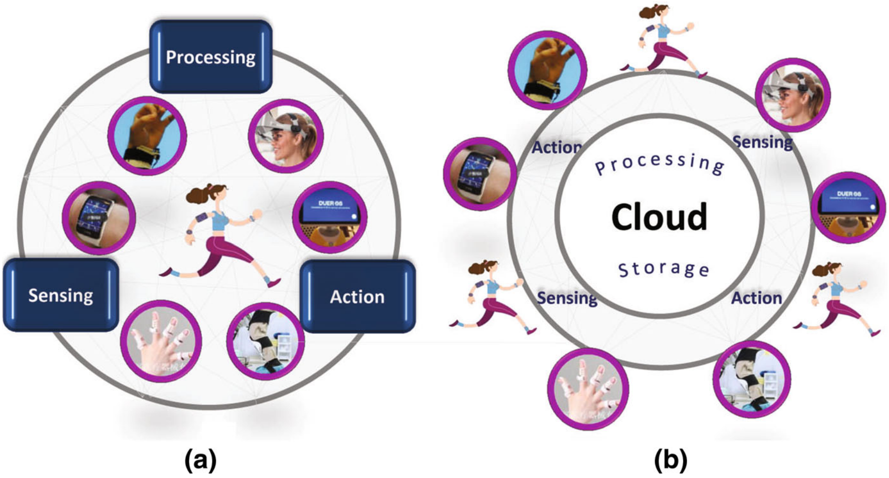


Fig. 1 High-level overview of existing computing paradigms: **a** local silo-based and **b** centralized cloud-based personal computing services

the reliability and real-time of such architecture are seriously affected by network quality and throughput.

Except for above issues in existing two frameworks, personal computing services also show unique features compared with other IoT-based applications: (1) latency-sensitivity [7], (2) energy efficiency, (3) high reliability, (4) mobility, (5) scalability, 6) privacy and security. A detailed analysis of these features will be discussed in the following section. Due to the characteristic of such services, it is obvious that the two mentioned frameworks studied by current researchers might not be good solutions for personal computing services. For the local silo-based framework, it is difficult to get the goal of "smart" or "intelligent" as the system with limited resources always works in a silo-based manner. Obviously, it is also the main reason for the tough implementation of features 2 and 3 of personal computing services, which finally bring trouble for developers with a tradeoff between "smart" and user experience. Furthermore, the remaining features of personal computing services are difficult to be realized in centralized cloud computing frameworks.

Recently, a novel and booming computing paradigm was introduced, named edge computing (also known as fog computing [8], cloudlet [9], mobile edge computing [10]). It takes the computation migration from the cloud to the edge of the network as the ultimate objective [6,11–14]. In other words, edge computing will be an efficient architecture for computation and storage in the proximity of the data source [15]. Accordingly, it is predicted that to be an effective framework for personal computing services in our paper. Currently, several edge computing based methods using a hybrid approach in which such as gateways or foglets are used as a middle layer to orchestrate the process between the end devices and the cloud have been proposed, which refers to multiple fields, such as healthcare [16–18], smart cities [19], smart homes [20], etc. The hybrid-based approach focuses on the interaction between the edge and the cloud, and it has been proved to improve the performance of system. Nevertheless, few researchers focus on pervasive personal computing services from the edge computing viewpoint. In this paper, we analyze the special features of personal computing, design the overall system architecture and the conceptual framework of personal computing services, and give a comprehensive summary of design challenges and opportunities. The main contributions of this paper are:

(1) We have discussed the differences between services based on the local silo-based framework and that those based on the centralized cloud computing framework, as well as the main special features of personal computing to show the advantages of edge computing in this service domain.
(2) A comprehensive edge-based system architecture for personal computing services is proposed, while a conceptual framework for edge-based personal computing system is designed, in which various functionalities of the edge node are analyzed in detail.
(3) Relevant hotspot research fields about various edge-based applications that have been implemented, as well as task management are concluded respectively.
(4) Current challenges and opportunities for edge-based personal computing applications are pointed out.

The remainder of this paper is organized as follows. Section 2 shows the background of this paper, including the discussion of existing computing paradigms and the main special features of personal computing services. Then, we discuss the advantages of edge computing in Sect. 3. Based on the conclusion of Sects. 2 and 3, we introduce a novel edge computing paradigm-based solution for personal computing services. The overall architecture and detailed conceptual framework are designed and presented in Sect. 4. In Sect. 5, fall detection application is simulated as example services to demonstrate the advantages of the edge computing-based framework in personal computing services. Section 6 mainly discusses and concludes the existing research hotspots in edge computing. We analyze the open issues and potential opportunities that the edge-based personal computing system offers in Sect. 7. The last section concludes this paper.

2 Background

We define personal computing services as those that users can access directly and/or must be available to users whenever they want to access or interact with them, which is similar to the definition of the personal service environment as shown in [21]. They can also be implemented or finished by the means of electronic or information technologies. Large amounts of personal computing services have been researched or developed. As discussed in the former section, the computing frameworks for such services are mainly two styles; however, each computing framework still meets different kinds of challenges or issues due to different workflows. In this section, a detailed comparison of two current paradigms is given. Furthermore, we also discuss and conclude the special features of personal computing services which further indicate the motivation of our paper.

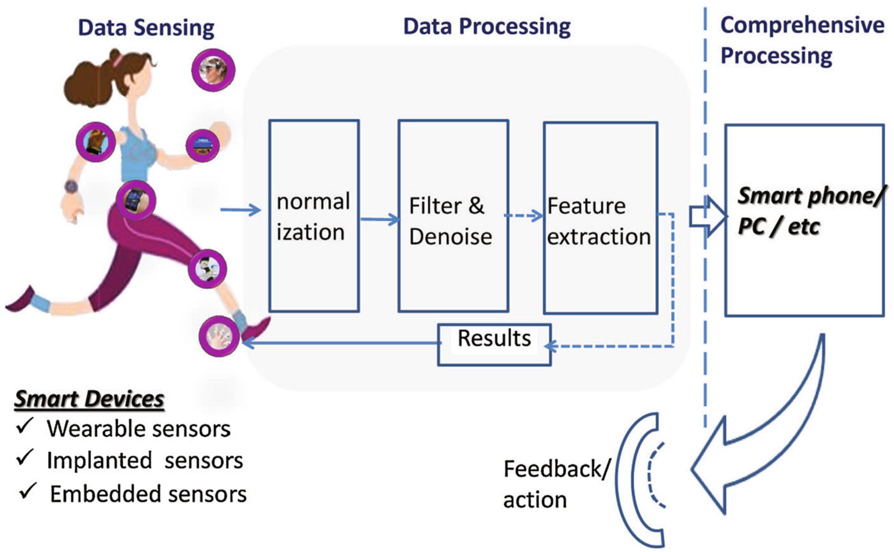


Fig. 2 Overall architecture of local silo-based paradigm

2.1 Existing computing paradigms

Although local silo-based paradigms and centralized cloud computing are commonly used in most IOT scenarios, these two independent computing methods have different characteristics.

Local silo-based paradigms Smart devices have become the norm in daily life, allowing users to enjoy personal computing services for personal convenience. For example, an elderly person can use a fall detector to detect a fall event to prevent further emergency events [22]. An individual at home can use a smart home app on their smart phone to control the lights in their home [20]. A human sitting in a car can enjoy music or air conditioning through systems in the vehicle. Most people can use existing resources in physical local devices to serve them although they work in a local silo-based manner. The functional composition of such a paradigm is shown in Fig. 2. The front-end sensors can sense and acquire signals at the proximity of the user, preprocess data, e.g. analog to digital (A/D) conversion, filter, denoise, data normalization, feature extraction and analysis, etc. In some simple applications, the smart front-end senor analyzes the features to evaluate or conclude results directly as shown on the left of the dashed line. However, due to the limitation of resources, for others, the alarm or final results are only acquired with the help of another resource-rich device, such as a smart phone or PC. Once an alert or emergency event occurs, it can inform a caregiver or rescue center accordingly through the communication interface on the devices. However, in such a local silo-based system which works in a silo-based manner, the application always has an independent system to achieve its functions, while there is no connection or communication with other systems due to multiple reasons, such as limited resources or limited battery capacity of smart devices. Therefore it is difficult for the individual to achieve real "smart" due to such a special paradigm. To solve this

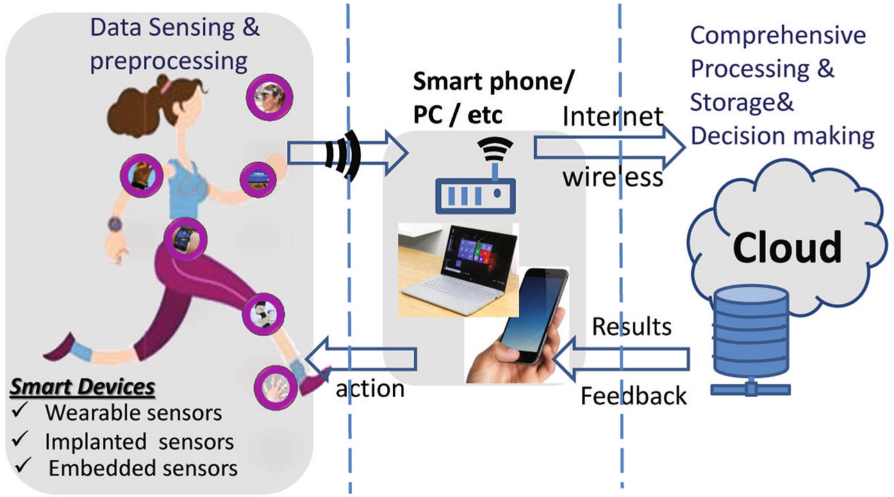


Fig. 3 Overall architecture of cloud computing paradigm

problem, the centralized cloud-based method is widely studied and applied, which is described in detail as follows.

Centralized cloud computing As defined in [23], cloud computing is a model enabling on-demand network access to a public shared computing resource pool that can be configured, where those resources can be rapidly provisioned without too much management efforts. Obviously, the cloud has abundant resources which can perfectly cover the drawbacks of the local silo-based architecture mentioned above. Accordingly, researchers prefer to use the centralization idea to improve the quality of various personal computing services. They push more comprehensive computing such as machine learning or Artificial Intelligence (AI) to the cloud server for "smart". Therefore, personal computing services can make use of resources in the cloud for further intelligence and interaction processing.

Today, many cloud-based services have been used and sprung up among our daily lives. For example, an individual at home can enjoy music or interact with Little Fish which have been launched by Baidu [24], a user can dispose of garbage by interacting with Sensor Can which can be opened by a person‘s voice (such as a product developed by Simplehuman) [25]. The overall architecture of cloud computing-based personal computing services is shown in Fig. 3. The front-end device collects significant information and does simple preprocessing. After that, communication and network transmission will be undertaken to send out the collected data or features to the cloud server for advanced computing. In the cloud layer, the cloud server will analyze, process and store the transmitted data. Furthermore, decision making can also be done on the cloud to guide the front-end device or the related person or institution to act. In all, cloud computing can benefit computation-intensive applications in an effective way, especially for those using complex algorithms. However, due to the insufficiencies of such a paradigm as concluded in Table 1, the cloud computing paradigm is not a perfect method for IoT applications, especially for personal comput-

Table 1 Comparison of existing computing framework and edge computing

Characteristics	Local silo-based framework	Cloud computing framework	Edge computing
Working mechanism	MCU processes data directly, or local device nearby provides assistant processing	Remote server centralized processes data from distributed applications	Edge node located closer to users collaborates with other edge nodes for further comprehensive processing. Meanwhile, it can communicate with the cloud for further mining
Size	MCU or/and local device, which is small enough	Tens of thousands of integrated servers	Edge node in each location which can be small or big according to the application
Geographical coverage	Local	Global	Local
Sever locations	No sever but with a simple MCU or smart device	Centralized server located in remote area	Distributed edge servers located in many locations or geographical areas
Bandwidth requirements	Local processing without bandwidth requirements. Or total data further processed can be sent to local smart device	Grow with the total amount of data generated by all the clients	Raw data or processed data (eg. data processing of optimized or adaptive data dissemination [32–35] to reduce network overhead) will be sent to edge node, which can be elastic adjusted based on the processing method, while further processed data need depth mining can be send to cloud
Internet connectivity	No	To upload large amount data, clients need to be connected to the cloud server by internet connection	Edge node can provide services even there is no or intermittent connectivity
Hardware resource	Extremely limited computing and storage resources	Large amount computing and storage resources	Limited computing and storage resources
Information type	Special information for the special application gotten from special device	Global information that is highly abstracted from many regions	Limited localized information and types gotten from near the data source
Data analytics	Simple but intensive analytics	Big data analytics	Simple but distributed collaboration analytics
Response time	Milliseconds to seconds	Seconds to weeks	Milliseconds to Minutes
Deployment	Localized	Centralized	Centralized or semi-distributed

ing applications with a real-time requirement. Cloud computing framework is heavily affected by internet connectivity. Large amounts of data simultaneously sent to the cloud will cause network congestion, further bringing out the latency issue. Moreover, network unreliability will lead to data missing, further triggering reliability issues. Security and privacy are also two major issues when transmitting data via the internet, which is easily attacked. Storing data on the public cloud also introduces high security and privacy problems. Therefore, we can conclude that the cloud computing paradigm still suffers from many issues when used in personal computing services. To sum up, a new computing paradigm needs to be proposed to solve issues that the two current computing frameworks are facing.

2.2 Special features of personal computing services

Personal computing services bring convenience to users by providing real-time feedbacks according to the users requirements, which is the ultimate objective of the system. However, personal computing services present a series of characteristics shown as follows:

(1) Latency-sensitive

Most personal computing services, such as fall detection, rehabilitation training, chronic diseases monitoring, health monitoring, physical training, sport monitoring, etc, have latency-sensitive characteristics to respond to emergencies or improve user experiences. All of those services or applications monitor the related status continually according to the requirements of the person, so a real-time cognitive response should be launched when there is a warning, alarm or even emergency. That is to say, a timely response plays an important role in getting a better user experience, as humans are sensitive to delays when interacting with smart devices. Take fall detection as an example, various fall detection methods are always used among the elderly for daily life monitoring to avoid dangerous fall events, a prompt forecast that an elderly person has fallen will prevent a disastrous injury, or even save the life of the person. Therefore, low latency is important, while the operations on cognitive response must be fast and accurate [26]. As it is mentioned in [27], in such a scenario, the end-to-end latency of a few tens of milliseconds is safe enough. Obviously, the centralized cloud computing paradigm is not efficient enough for those latency-sensitive applications [2] due to long data transmission time and the network delay.

(2) Low power consumption requirement

For most personal computing services, large amounts of data are collected and preprocessed by sensors, which are then transmitted to a device like a smart phone or PC or even the cloud via wireless transmission for further computing. However, above operations consume a lot of energy, especially for wireless transmission, which is verified in our former work [28]. High power consumption means frequent battery recharging or replacement, which is troublesome for users. As it is indicated in [28], more than 95% of volunteers consider that the battery life of a system will affect whether they will use the system. Beyond fall detection applications, energy-efficiency is also strongly required by other personal computing services. As users are the directly experiencing use of the product, they want a device that works a long time. For example, if a high power consumption heart rate device is used to monitor sports and physical activities, and it requires athletes to replace or recharge the battery when the device exhausts, this will interrupt the athlete's training, which further affects the quality of training. Therefore, the successful condition of a personal computing service first is low power consumption. Many approaches have been proposed to reduce the energy consumption of the a device. Truncated Multiplier is one way to reduce energy consumption at the expense of signal degradation. For example, Solaz et al. [29] employed a programmable truncated DSP to adaptively truncate multiplication within the algorithm. Data abstraction is another method for reducing energy consumption. In [30], the authors proposed a method to reduce the traffic load by constructing higher-level abstractions of data. Reducing the transmission power has also been proposed by several authors to save energy; for example, a low power transmission protocol was proposed in [31]. Those methods reduce the power consumption of front-end devices; however, more energy saving methods should be considered throughout the whole computing paradigm, especially for the cloud computing framework.

(3) Demand for high reliability demand

High reliability is the basic requirement of personal computing services, as it will guide users to act according to the decisions made by systems, which is usually affected by the procedures of data processing and the related algorithm. Therefore, it is important for personal computing services to provide high accuracy to ensure the reliability of the services. A low reliability service will bring about disaster; conversely, a high reliability system is useful. However, the local silo-based framework has limited resources in front-end devices which cannot support a complex algorithm and will further affect the accuracy of the final results. Transmitting related data to the cloud server for further computing is one effective idea to improve the reliability of the system, yet the cloud-based system brings out long latency and high power consumption issues.

(4) Mobility

For personal computing services, an individual is the main body to be served. However, the user is free to move anywhere he/she wants. Ideally, he/she can reach services autonomously within the coverage area wherever the individual is without interruption. When the user stays at home, he/she can access all the services deployed in the house; for example, he/she can tell lights to be brighter, open the trash can with his/her voice, etc. Once he/she moves into a new environment, the user can interact with new devices in the surroundings. Therefore, personal computing services should support mobility by accompanying the user as he/she moves around in the network. In light of this, mobility is one of the significant features of personal computing services, and is the pre-condition of "smart" personal computing services. Mobility of personal computing services requires flexibility to enable inclusion and exclusion devices. Obviously, it has not been implemented in local silo-based or cloud-based manner.

(5) Scalability

Due to the increasing numbers and types of devices that will be used in the future, scalability is one of the most important requirements for the IoT system, as well as for personal computing applications. It is believed that as more smart sensors are produced or new requirements emerge, individuals will apply new devices to provide rich personal computing services in the surrounding environment, which lays claim to the scalability of personal computing services. Moreover, due to the mobility of personal computing services, system will add devices to provide services when the individual comes in, which also expresses the scalability of the system. However, no matter whether a local silo-based or cloud-based system is used, scalability has not been considered by researcher to date. Especially for local silo-based application, it is difficult to add different kinds of devices without re-development.

(6) Privacy and security

Personal computing services use front-end devices to collect data from users, which is analyzed to provide feedbacks accordingly. Each step in data streaming refers to sensitive information about the user. Once unauthorized or malicious access successfully occurs during any data stream, negative impacts or even threats may act upon users. For example, the user might share his/her health information with a doctor; however, the information may be intercepted and modified during network transmission. The data acquired by doctor maybe a sheet of black squares or gibberish, which prevents the doctor from making the proper diagnosis. Therefore, privacy and security are two important concerns in personal computing services, as a large amount of the data collected refer to individuals. For the cloud-based framework, due to the fact that the cloud is public, security and privacy are two difficulties that need to be solved, as they occur commonly and frequently.

Though two computing frameworks including local silo-based architecture and centralized cloud computing-based architecture, have been commonly used in personal computing services, each computing paradigm has pros and cons. Meanwhile, personal computing services have unique characteristics, which further indicate that the two existing computing frameworks are not quite fit for personal computing services to achieve the goal of real “smart”.

3 Benefits of edge computing framework

Personal computing services have drawn a great attention, since convenient services have become more and more popular. The flourish of edge computing provides a new computing paradigm that can decompose complex computing tasks into small events or elements, which can be further processed on the local edge node to achieve distributed computing. That is to say, edge computing allows for decentralizing the computing flowing to the edge of the network, so that data processing and storage can be done close to data sources. According to the idea of edge computing, a huge amount of data generated at the source of the personal computing system can be transferred to edge or collaborative edge devices nearby to do computing, storage, communication, etc, instead of the remote cloud server. This operation will reduce the burdens of the network, and also solve problems such as the latency and high power consumption caused by the centralized cloud computing paradigm. Meanwhile, the edge computing paradigm can also better address all the issues met by the local silo-based system. Furthermore, edge computing also easily meets the features and requirements of personal computing applications in terms of latency, security, energy efficiency, reliability and mobility. In total, the edge computing paradigm could be a good method for personal computing services by allowing computation to be performed at the source of the data, which can cover the bottleneck problems of the two existing solutions. To show the advantage of edge computing used in personal computing services, we compare the three computing frameworks mentioned above in Table 1.

We have selected as many features as possible for Table 1. From the table, it is clear that edge computing is very different from the local silo-based framework and cloud computing. Edge computing aims to carry out the related operations such as data processing and storage on the edge side closer to data sources, which are similar to the local silo-based framework; meanwhile it also emphasizes edge collaboration with other edge nodes to share data and resources. Based on the comparison list in Table 1, the edge computing paradigm perfectly complements and compensates for the disadvantages of the two existing computing frameworks. In addition, as described in [5], the edge computing paradigm also shows advantages in terms of response time, energy consumption, mobility, scalability, security, cost, etc, which are identical to the features of personal computing services. Consequently, the edge computing paradigm is quite fit for personal computing services, which inspires us to penetrate edge computing into personal computing services. Finally, the expectation of edge computing-based personal computing services is shown in Fig. 4. As many computing and storage tasks as possible are migrated to the edge network, while the cloud processes and finishes only few tasks that is not strict in time.

4 Edge computing model for personal computing services

Since some critical issues continue to occur when personal computing services work with the two existing computing paradigms, we envision integrating edge computing into personal computing services. In this section, we first propose the overall architecture of edge computing based personal computing services. Then, the conceptual diagram of the proposed system is discussed.

4.1 Overview of system architecture

In personal computing applications, data is produced by various sensors or smart devices at a range of places around the individual, while the data must be consumed to serve the user for convenient services in real-time. Edge computing commits to compute and store data at the proximity of data source, which is appropriate for personal computing services as both data production and consumption are at the data source. Therefore, we attempt to utilize this new computing paradigm for personal

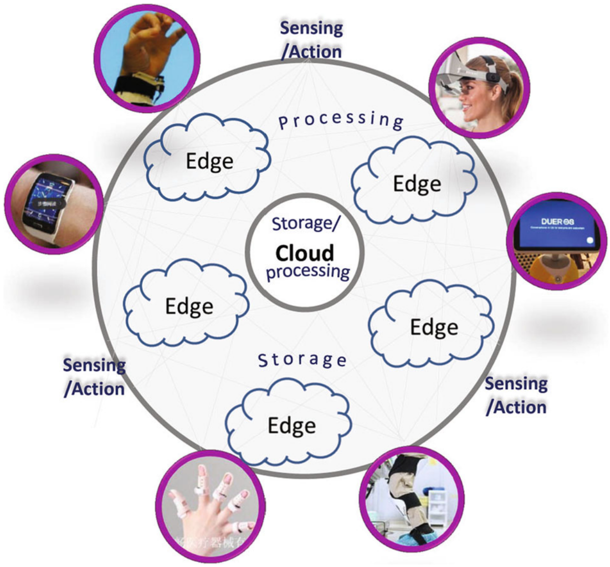


Fig. 4 High-level overview of edge-based personal computing services

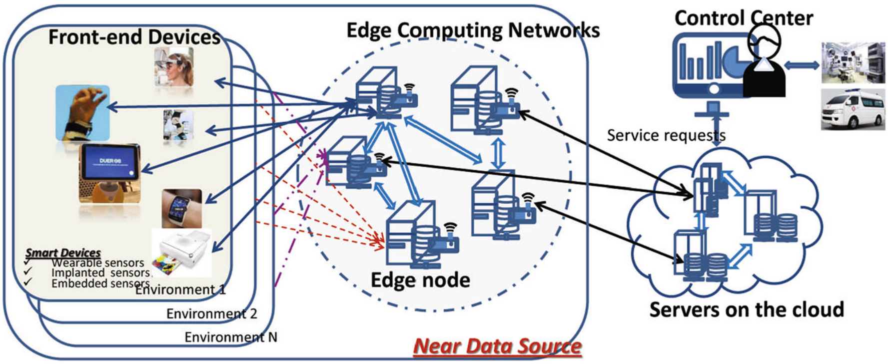


Fig. 5 Overall system architecture of edge computing-based personal computing services

computing services. The overall system architecture of edge computing-based personal computing services is illustrated in Fig. 5. The system is composed of three main components, including front-end devices, edge computing networks and the remote cloud. Both the front-end devices and edge computing networks work at the data source, while the cloud provides services far away. A brief description of the related devices in the system, i.e. front-end devices, edge node, and cloud server, is shown below.

(1) *Front-end devices* Front-end devices have the capacity of sensing, which can be located in/on/around the body of an individual as shown in Fig. 5. Clearly, front-end devices work as data producers generating raw data. However, as shown in the figure, the row between the front-end devices and edge computing networks is bi-directional, which means that front-end devices not only act as data producers, but they also

consume data, unlike the conventional cloud computing paradigm. In other words, front-end devices themselves can sense data, but also request services and content or perform tasks from the edge/cloud. Devices in home environments, such as smart can, camera, little fish, etc. are some examples of front-end devices. Other examples are EKG sensors [36], fall detectors, thermometers, etc. All front-end devices in different environments can interact with edge computing networks.

(2) *Edge node* Since the authors in [14] defined "edge" as any computing and network resources between data sources and the centralized cloud, we consider edge nodes as devices that can connect to either front-end devices or the cloud, while they can also do computing tasks, such as machine learning, making decision, etc,. Those devices cloud be smart phones, routers, gateways, switches or even a new device that can be connected to front-end devices for computing tasks. In the framework, they mainly do complex data processing, analysis and storage, especially for latency sensitive applications. Edge nodes have more rich computing and storage capacities than front-end devices, but resources are still limited in edge nodes. Therefore, most urgent tasks can be finished by edge collaboration [14], while other tasks can be transmitted to the cloud through internet connectivity for further data processing.

(3) *Cloud server* The cloud server has remarkable features of large storage space and powerful processing capacity, which allows it to support rich applications and services. Thus, it has frequently been used in many traditional applications, such as finance, education. In edge computing mode, the cloud server is still important and cannot be replaced although most computing tasks have been performed at the edge. As the resources of the edge node are limited as discussed above, large scale data/features should be transferred to the cloud for long term storage, data modeling, and history analysis. Furthermore, some remote access or services can be implemented by connecting to the cloud but not edge node or the front-end devices.

To summarize, in such a system architecture, related computing and storage tasks will be processed on either front-end devices or edge computing networks deployed close to the individual, so as to immediately react to the person. In other words, the objective of edge computing-based personal computing services is to use any available computing and storage resources close to the data sources including data producers, which can be different types of sensors, smart devices, or edge nodes in edge computing networks, while the remote cloud is used to store processed results for those latency tolerant services.

4.2 Conceptual edge framework for personal computing system

By shifting many operation tasks to the edge side, a personal computing system is expected to enhance more functionality in terms of smart collaboration, low-latency, security, energy efficiency, reliability and mobility compared to existing system architectures as concluded above. Therefore, we propose a conceptual design of edge-based personal computing system shown in Fig. 6.

The proposed conceptual framework consists of three layers, including a sensor network, a smart edge node and an edge-side services/remote cloud. The sensor network is constructed with different smart devices in a person's immediate environment. For

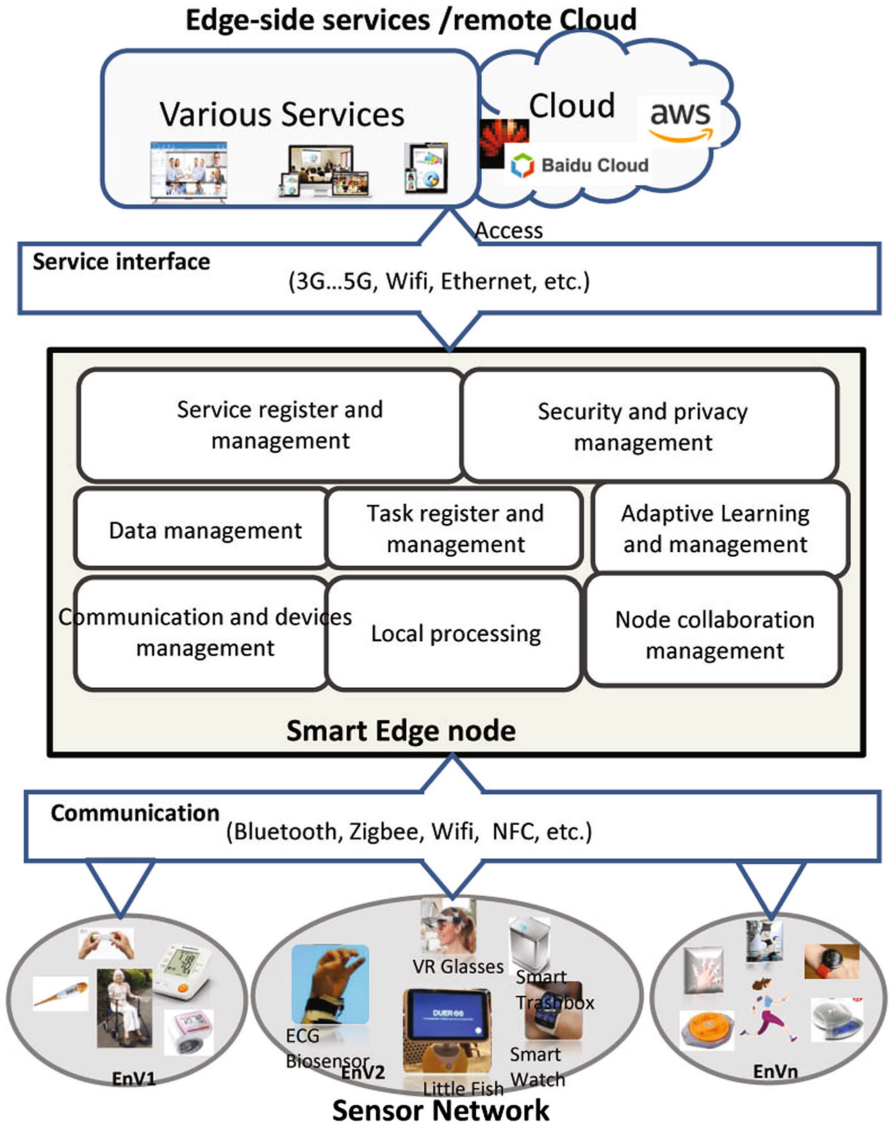


Fig. 6 Conceptual framework of edge node for personal computing services

personal computing services, the environment can be a house, a fitness center, a nursing room, or even in a car. Sensors in the network can sense data, do simple computing, and act according to results from the edge/cloud, while preprocessed data will be sent to an edge node for further computing. Edge-side services/remote cloud as labeled in this figure are proposed from the service perspective, in which various services can initialize a service request to access data in the edge side. Conversely, data/results produced in the edge node can also be sent to edge-side services/remote cloud. A smart edge node is the core of this proposed framework and have been mainly designed in detail, including communication and device management, local processing, node collaboration management, task register and management, data management, adaptive learning and management, service register and management, and finally, security and privacy management. Due to the design of these functionalities, the system supports connection and communication of various front-end devices, local processing and decision making while allowing for edge collaboration, storage, etc. Components corresponding to the edge node are explained as follows.

(1) *Communication and device management* Personal computing services enable various devices with different communication protocols to provide services, such as commonly used Bluetooth, Zigbee, WiFi, 4G/5G, etc. Thanks to the design of communication and device management, those devices with different interfaces can be connected and communicated with edge node. Once devices are connected to the system, they need to be registered for further device management. Only in this way can the edge node dynamically add or delete or even renew devices to support the extendibility of the system and the mobility of personal computing services.

(2) *Task register and management* Edge nodes can collaborate with each other for one function; that is to say, not all computing tasks are done in one edge node. Therefore, a function can be divided into multiple subtasks which can then be distributed among different edge nodes. To manage those subtasks, a task register and management module has been designed. All tasks should be registered first, so as to easily manage them, and then, it responds to do the task allocation and scheduling.

(3) *Node collaboration management* As concluded above, edge nodes still have limited resources and capacities, so each edge node should share data and computing tasks with each other for more complex functionalities. Therefore, node collaboration management has been designed in this framework to implement complex computing or make decision with edge collaboration. This module will decide which edge nodes should interact to achieve one function. When necessary, it can also make decisions about which front-end devices need to sense data.

(4) *Local processing* Since edge nodes respond to complex computing near the data source, a local processing module has been designed. Unlike other computing paradigms, for a function, each related edge node will only finish some subtask computing that is allocated by the task register and management module. A subtask such as pre-processing should be finished by the local processing module.

(5) *Data management* Front-end devices will transfer all related data to the edge node. Due to different data sources, there are differences in data formats, which should be processed first for further data processing. A data management module designed to respond to all data operations can solve this. Meanwhile, it can also provide other data operations such as data filtering, data compression, data standardization, data fusion, data storage, etc.

(6) *Adaptive learning and management* Adaptive is one typical component that needs to be considered to provide better user services, especially when critical abnormal events occur. For example, data exhibits periodicity in some services, so adaptive learning can be used to find abnormal data, which can improve the accuracy and reliability of personal computing services. Adaptive learning can also be used to adjust transmission rate while considering both energy efficiency and accuracy [37]. For normal events, low transmission rates can be used to reduce power consumption due to

long term wireless transmission; once an emergency event happens, the system should adaptively learn and increase the transmission rate to improve algorithm accuracy. Of course, the functions of adaptive learning and management are not limited to two situations, but these are just two examples.

(7) *Service register and management* There will be many applications or services at the edge-side or cloud end accessing the edge node network to request data. This module responds to manage services that launch requests, including a service register, service release, service distribution and migration.

(8) *Security and privacy management* Since all the information along the path of this framework refers to individuals, this information is critical for users, but it is easy for an attacker to intrude. Once the system is attacked, privacy information will be revealed for some purpose usage, meanwhile, related sensing data can be malevolent distorted, which may bring serious consequences for the individual. Therefore, security and privacy management is necessary to be included in our system, which can ensure the system has high reliability while runs in a security environment.

5 Analysis of preliminary experiments

In this section, accelerometer-based fall detection algorithm, which shows low data stream and computing stream, will be used as a personal computing service example to prove the feasibility of the proposed edge-based personal computing framework. For such an application, smart sensors designed with various transmission interfaces, such as Bluetooth, WiFi, etc, have been used by the elderly for fall detections. These front-end devices can sense information of the body or environment to detect if a fall is occurring or not, while they also have the capability of data transmission. However, due to the resource limitation of front-end devices, smart sensors can only do simple computing and analysis.

5.1 Experiment setup

To verify the credibility of using edge computing for personal computing services, we simulated fall detection scenes using two Raspberry Pi platforms. One Raspberry Pi is used to substitute for a the smart front-end device to sense body movement. Due to the mobility of personal computing services, obviously, a mobile phone will be an adaptive edge server platform. Here, for easy evaluation, the other Raspberry Pi is chosen as an edge server located nearby due to its similar computing and storage capacities. For the cloud server, Baidu Cloud Compute located in Suzhou, Jiangsu is used.

Here, we assume an elderly person in his/her house in Hefei, Anhui Province, where is about 400 km from the cloud, needs to be monitored to avoid falls, the edge node is physically located in the room nearby. Wearing the smart sensor, the person can act freely in his/her own house. Based on this description, an accelerometer-based fall detection scenario with multiple network topologies have been considered and tested as shown in Table 2, respectively. In this table, edge only means that the sensing data

Table 2 Experimental network topologies for accelerometer-based scene

Classification	Topology	Description
Edge only	WiFi only	Sensor collects data which is transmitted directly through Wifi to the edge node for further processing and feedback, while edge server and the cloud is only connected through wired
	BLE-Wired	Sensor collects data which is transmitted directly through Bluetooth to the edge node for further processing and feedback, while edge server and the cloud is only connected through wired
	BLE-WiFi	Sensor collects data which is transmitted directly through Bluetooth to the edge node for further processing and feedback, while edge server and the cloud is only connected through WiFi
	BLE-4G	Sensor collects data which is transmitted directly through Bluetooth to the edge node for further processing and feedback, while edge server and the cloud is only connected through 4G
Edge processing	WiFi only	Sensor collects data which is transmitted directly through WiFi to the edge node for further processing, then final results are send to the cloud through WiFi
	BLE-Wired	Sensor collects data which is transmitted directly through Bluetooth to the edge node for further processing, then final results are send to the cloud through wired
	BLE-WiFi	Sensor collects data which is transmitted directly through Bluetooth to the edge node for further processing, then final results are send to the cloud through WiFi
	BLE-4G	Sensor collects data which is transmitted directly through Bluetooth to the edge node for further processing, then final results are send to the cloud through 4G
Cloud processing	WiFi only	Sensor collects data which is transmitted directly through WiFi to the cloud for further processing
	BLE-Wired	Sensor collects data which is transmitted directly through Bluetooth to the edge node firstly, and then is sent to the cloud for further processing through wired
	BLE-WiFi	Sensor collects data which is transmitted directly through Bluetooth to the edge node firstly, and then is sent to the cloud for further processing through WiFi
	BLE-4G	Sensor collects data which is transmitted directly through Bluetooth to the edge node firstly, and then is sent to the cloud for further processing through 4G

is only transmitted and processed on the edge server; edge processing means that the sensing data is first sent to the edge server for processing, and then the results are further sent to the cloud; cloud processing means that sensing data is transmitted to the edge server first, which is then sent to the cloud again for further processing. WiFi only means the whole topology uses WiFi, while BLE-wired means the first hop is Bluetooth and the second hop uses wired transmission, similar to BLE-4G and BLE-WiFi. Simply, the acceleration data used in our experiments was acquired from [38] directly.

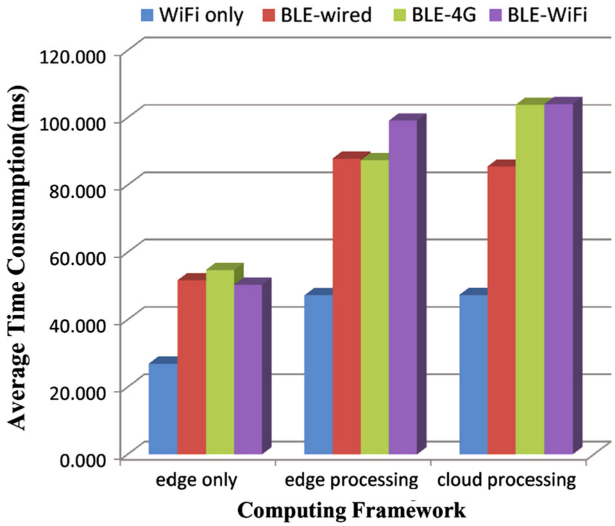


Fig. 7 Average time consumption under different network topologies of accelerometer-based fall detection algorithm

5.2 Preliminary experiment results

During the above experiments, we mainly measured the performance of each topology in terms of the average delay time under the described scenario. Figure 7 illustrates the average time consumption of the accelerometer-based fall detection algorithm for edge only, edge computing and cloud computing under different wireless communication topologies as shown in Table 2. During the experiments, each experiment was repeated 990 times. From this figure, it is clear that the performance of the system is better with WiFi as the first hop with around 49.69% total time consumption reduction compared to Bluetooth on average. Therefore, although the front-end devices of personal computing services always have various transmission interfaces, a WiFi-based smart device is a better choice. Also as expected, edge processing shows better performance than cloud processing in terms of the average time consumption in general, especially in the case of applying 4G at the second hop. That is to say, the edge computing framework is quite fit for this application scenario. However, the cloud computing framework with a network topology of BLE-wired showed better performance than the edge computing with the same topology, so it is estimated that this is because the data and computing of accelerometer-based fall detection is too small. We believe that if more complex personal computing services are added, the advantage of the edge computing framework will be demonstrated. Except for this, using the edge alone with any network shows better behavior than the other two classifications, with about 42.84% average time consumption reduction compared to edge processing, while it saves an average of 46.07% more time compared to cloud processing. Actually, for this special fall monitoring application, the elderly are in normal status most of the time, with only few falls occurring. Therefore, we should consider task offloading on the sensor and edge to take advantage of edge only and edge processing to further enhance the edge computing framework.

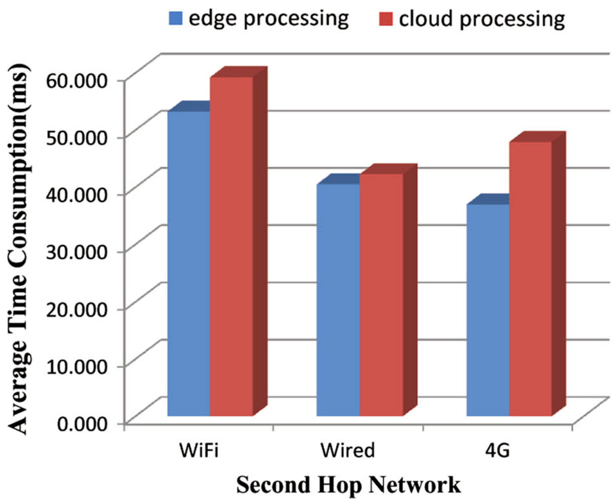


Fig. 8 Average time consumption of second hop

Furthermore, topologies with the same first hop network but different second hop were test in our experiment, and the average time consumption of the second topology is illustrated in Fig. 8. Obviously, this figure also shows the advantage of the edge computing framework in terms of average time consumption compare to the cloud computing framework under the ideal situation of the same average time consumption of the first hop. Especially, edge computing framework with 4G as the second hop has more obvious performance, with about 22.75% latency saving, which is about 47.912 ms consumed by cloud computing framework, while 37.012 ms for the edge processing framework, which is consistent with the results concluded in Fig. 7.

Here, the average time consumption for various operations has also been explored, those operations mainly include data transmission, data caching, SVM-based fall detection algorithm with lin kernel function, SVM-based fall detection algorithm with rbf kernel function and SVM-based fall detection algorithm with sig kernel function as shown in Fig. 9. Obviously, data transmission accounts for the most time of the event, with nearly 96%, and also data transmitted to the edge will consume less time compare to sending it to the cloud, both of which further reveals that edge computing is adaptive for fall detection applications as little data will be uploaded to the cloud in such a personal computing service.

6 Related work

The flourish of edge computing has attracted large attentions of different field researchers to do relevant research, including application framework, energy-efficiency for edge computing system, security, and also task management of the edge-based model. Here we only conclude studies from the following two perspectives: application implementation based on the edge computing framework, and task management in the edge computing paradigm. These two topics are the focus of current research.

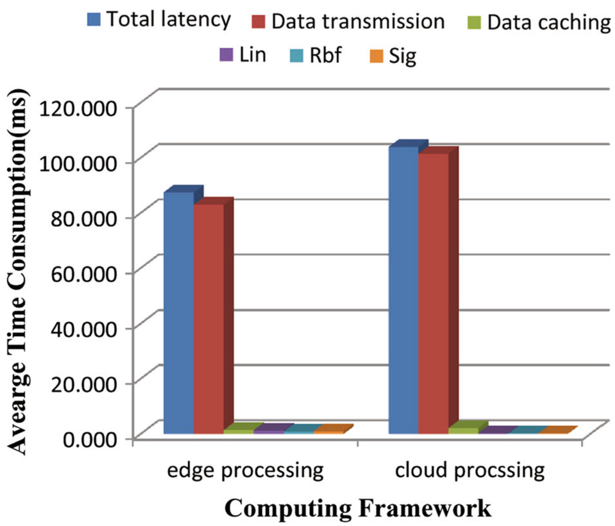


Fig. 9 Average time consumption comparison of operations under BLE-4G topology

6.1 Services based on edge computing frameworks

From the time that edge computing was recently proposed, it has been thought to be quite fit for IoT scenario. Many IoT researchers compete to integrate edge computing into their research. Edge computing was applied to fall detection monitoring as shown in [19]. They successfully designed a real-time fall detection system based on the edge computing paradigm. Similarly, [37,39] are two papers proposing edge computing architecture for healthcare. Smart home is another typical application that can use edge computing, Cao et al. [20] proposed the overall vision of EdgeOS_H for a smart home. In addition, they also discussed functional challenges for the implementation of EdgeOS_H. Work [40] stated next generation firefighting is based on edge computing paradigm, the overall system architecture of which was designed in detail. Meanwhile, authors also discussed the opportunities, challenges and technical suggestions on constructing such edge-based firefighting. Edge computing has also been found fit for AMBER Alert, Zhang et al. [41] proposed an enhanced AMBER alert system using edge collaborative, which can be used to search for suspect vehicles in real-time. Connected and autonomous vehicles (CAVs) are computation-intensive services, and are often equipped with various sensors, which will generate about 4TB data per day as reported by Intel [42] to monitor themselves and the surroundings to support autonomous driving and other services. Obviously, this is a typical edge computing application scenario. Accordingly, Zhang et al. [43] proposed a full-stack edge based platform for CAVs, named OpenVDAP, which will be an open-source platform for other research and evaluations.

6.2 Task management in the edge-based model

In this subsection, we summarize computing/task offloading in the edge computing paradigm. Lin et al. [6] decomposed applications into several function modules, which are considered as the basic unit for migration. However, the decisions regarding which nodes should host which function module, and which service requests should serve for each function module instances are important, because reasonable function module allocation will minimize the total network bandwidth. The authors proposed a heuristic algorithm to solve this, which has been evaluated to reduce bandwidth consumption by up to 40%. In research reported in [7], the authors provide video analytic services based on the edge computing paradigm for latency-sensitive applications, and they have built LAVEA, which offloads computing between clients and edge nodes to provide low-latency analytics. They used edge-first thought and formulated an optimization problem for offloading task selection and prioritized task request queues to minimize the response time. In [2], the authors modeled application, network, data and cost functions, and then, task allocation was formulated based on data distribution, task dependency, embedded device constraint, and device heterogeneity simultaneously. Sundar and Liang [44] considered the scheduling decision in a network of heterogeneous local processors and a remote cloud server, while formulating the optimization problem to minimize the overall application execution cost subject to an application completion deadline. Wang et al. [45] considered that different health tasks have different priorities to be processed, they concluded that the existing task management approach based on factors such as energy, bandwidth consumption, etc, is insufficiently efficient in such conditions. Therefore, they proposed a task scheduling approach that considers human health status to decide task processing priorities and where the tasks should run. The edge computing paradigm applies distributed computing to decentralize the data to the edge of the network. Clearly, a simple and effective distributed computing algorithm is essential in different edge-based applications, which presents many opportunities.

7 Challenges and opportunities

Edge computing is quite fit for IoT applications due to its distributed computing near the data sources; however, there are still some challenges as well as opportunities for us when it is used in latency-sensitive personal computing services, including device management and communication, data management, task management, and security and privacy. We discuss these challenges in this section in hopes of informing potential researchers or developers.

(1) Device management and communication

As personal computing applications are vary, different types of devices will be connected and used to provide services. Accordingly, the designed system architecture should allow heterogeneous devices with different types of transmission protocols to be connected to the system and managed by the system simultaneously. For example, as discussed above, mobility is one feature of personal computing services, individuals can move anywhere he/she wants. In such situations, he/she may want to launch services in a new environment, so the edge node should automatically connect and support related devices no matter which type of transmission protocol they belongs to, without any interruption. The edge-based personal computing system also supports communication among different types of devices, such as front-end device, edge node and cloud [2], which means data from one layer should not only be shared among them, they should also be understood and recognized by other layers for distributed computing. All the challenges discussed above require good pre-design reasonably.

(2) Data management

As the edge-based personal computing system is a heterogeneous data-oriented architecture, how to unify data so that all the edge nodes can understand and interact with each other is one of essential challenges that need to be solved. Otherwise, there will be interoperability issues among heterogeneous device. Furthermore, different service data has different degrees of urgency in personal computing applications, so the definition or measurement of data priority is necessary. However, how to predefine the priority of data is worth further consideration. There are also many other data management issues to resolve, such as, how to share data from different types of devices and how ensure the quality of data and the function of the system.

(3) Task management

Since edge computing is supposed to be a distributed computing paradigm, it requires applications or computing tasks to be divided into slices and distributed to related edge nodes. However, where the tasks should be placed (also described as task allocation and placement) is a major topic of discussion, which also brings out some other challenges, such as which device responds to task division, how to divide the application into subtasks, and what size those subtasks should be, as either large or small size will bring out different results. If the subtask is cut into a slice that is too small, the package or interconnection due to task scheduling will consume extra energy and time. On the other hand, if the subtask is cut too big, it will not be a commonly used module by other services; in other words, the usability of such a subtask will be low. Subtasks are frequently distributed among edge nodes; actually, the cloud, which is rich in various resources, can also do computing tasks, so subtasks can also be shared to cloud. In this situation, many related challenges need to be solved, such as how to collaborate tasks between edge nodes and the cloud to achieve a joint resource optimization of whole system.

(4) Security and privacy

Personal computing services have feature of high security and privacy, however, due to the distributed architecture of edge computing, security and privacy become more complicated [46], which creates a big challenge. First, data streams among the whole system carry sensitive data or information; however, unauthorized access, multiple data mangers, and data sharing by different devices will affect the security of the data. On one hand, data may be distorted for some negative purposes, so the accuracy of the service may be reduced or even be abnormal. On the other hand, personal data about services may be stolen for unreasonable purposes. Then, the reliability issue of data, service or network also causes security problems. Any unreliable event will cause the system to work abnormally. As data streams contain sensitive data or personal information, once it is accessed and used illegally, it will refer to privacy issues. Providing effective security and privacy remains a big challenge for us.

8 Conclusion and future works

In this paper, we compared existing computing paradigms used in personal computing applications and analyzed features of personal computing services. Through the analysis concluded above, a new computing framework for personal computing services is inevitable. The emerging edge computing framework focusing on distributed computing at the edge of the network is quite fit for personal computing services. Therefore, we designed the overall system architecture for personal computing services, including front-end devices, edge node networks and the cloud. Meanwhile, a conceptual framework for an edge-based personal computing system was introduced, in which 8 functionalities were designed and described in detail. Then we reviewed several applications implemented based on the edge computing framework and also one research hotspot of task management in edge computing. A fall detection method was implemented as a personal computing service to evaluate the performance of the proposed system architecture in terms of average time consumption. The preliminary evaluation shows our system framework is efficient for personal computing services. Finally, we put forward challenges and opportunities worthy of further research, as we hope this paper can provide useful information to inspire researchers in related fields to consider relevant issues to contribute the topic.

Acknowledgements The authors would like to thank for Qiangyang Zhang for his help in experiments. The author also would like to thanks the anonymous reviewers for their valuable comments and suggestions. This research was supported by Natural Science Foundation of Shandong Province(ZR2018BF014), Scientific Research Foundation of Shandong University of Science and Technology for Recruited Talents(2017RCJJ042), and Key Laboratory for wisdom mine information technology of Shandong Province, Shandong University of Science and Technology.

References

1. Islam SMR, Kwak D, Kabir MH, Hossain M, Kwak KS (2015) The Internet of Things for health care: a comprehensive survey. IEEE Access 3:678–708
2. Sahni Y, Cao J, Zhang S, Yang L (2017) Edge mesh: a new paradigm to enable distributed intelligence in Internet of Things. IEEE Access 5:16441–16458
3. Frost Sullivan (2015) Power management in Internet of Things (IoT) and connected devices. https://www.frost.com/sublib/display-market-insight.do?id=294383010. Accessed 22 Jan 2018 (Online)
4. Meola A (2016) Internet of things in healthcare: information technology in health. http://www.businessinsider.com/internet-of-things-in-healthcare-2016-8. Accessed 22 Jan 2018 (Online)
5. Akmandor AO, Jha NK (2017) Smart health care: an edge-side computing perspective. IEEE Consum Electron Mag 7(1):29–37
6. Lin H, Shih Y, Pang A, Lou Y (2016) A virtual local-hub solution with function module sharing for wearable devices. In: MSWiM '16 proceedings of the 19th ACM international conference on modeling, analysis and simulation of wireless and mobile system, NY, USA, pp 278–286

7. Yi S, Hao Z, Zhang Q, Zhang Q, Shi W, Li Q (2017) LAVEA: latency-aware video analytics on edge computing platform, Sec17, CA, USA
8. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the Internet of Things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing, pp 13–16
9. Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for vm-based cloudlets in mobile computing. IEEE Pervasive Comput 8(4):14–23
10. Patel M et al (2014) Mobile-edge computing introductory technical white paper. White Paper, mobile-edge computing (MEC) industry initiative
11. CISCO (2015) White paper: fog computing and the Internet of Things: extend the cloud to where the things are. http://www.cisco.com/c/dam/enus/solutions/trends/iot/docs/computing-overview.pdf. Accessed 22 Jan 2018 (Online)
12. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protoclos, and applications. IEEE Commun Surv Tutor 17(4):2347–2376
13. Zhanikeev M (2015) A cloud visistation platform to facilitate cloud federation and fog computing. Computer 48(5):80–83
14. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: vision and challenges. IEEE Internet Things J 3(5):637–646
15. Zhang Q, Zhang X, Shi W, Zhang Q, Zhong H (2016) Firework: big data sharing and processing in collaborative edge environment. In: Proceedings of 4th IEEE workshop on hot topics in web systems and technologies (HotWeb), Washington DC, October, pp 24–25
16. Constant N, Borthakur D, Abtahi M, Dubey H, Mankodiya K (2017) Fog-assisted wIoT: a smart fog gateway for end-to-end analytics in wearable Internet of Things. In: 23rd IEEE symposium on high performance computer architecture HPCA 2017. Texas, USA
17. Dubey H, Monteiro A, Constant N, Abtahi M, Borthakur D, Mahler L, Sun Y, Yang Q, Akbar U, Mankodiya K (2017) Fog computing in medical internet-ofthings: architecture, implementation, and applications. In: Khan SU, Zomaya AY (eds) Handbook of large-scale distributed computing in smart healthcare. Springer, Berlin
18. Cao Y, Hou P, Chen S (2015) Distributed analytics and edge intelligence: pervasive health monitoring at the era of fog computing, Mobidata 15, Hangzhou, China, pp 43–48
19. Barik RK, Dubey H, Samaddar AB, Gupta RD, Ray PK (2016) Foggis: fog computing for geospatial big data analytics. In: 3rd IEEE Uttar Pradesh section international conference on electrical, computer and electronics engineering, Varanasi, India
20. Cao J, Xu L, Abdallah R, Shi W (2017) EdgeOS_H: a home operating system for internet of everything. In: Proceedings of the 37th IEEE international conference on distributed computing systems (ICDCS), Vision/Blue Sky Track, Atlanta, USA
21. Bylund M, Waern A (2001) Personal service environment-openness and user control in user-service interaction, SICS research report
22. Ren L, Zhang Q, Shi W (2012) Low-power fall detection in home-based environments, Mobile-Health12, South Carolina, USA
23. Mell P, Grance T (2011) The NIST definition of cloud computing. National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, USA, technical report, pp 50-50
24. TIME.COM (2017) This new computer wants to be the ultimate AI assistant for the home. https://www.tuicool.com/articles/ZraMjuz. Accessed 22 Jan 2018 (Online)
25. Lee N (2017) Simplehuman made a trashcan you can open with your voice. https://www.engadget.com/2017/01/05/simplehuman-made-a-trashcan-you-can-open-with-your-voice/. Accessed 22 Jan 2018 (Online)
26. Ha K, Chen Z, Hu W, Richter W, Pillai P, Satyanarayanan M (2014) Towards wearable cognitive assistance. In: International conference on mobile systems, NY, USA, pp 68–88
27. Satyanarayanan M (2017) The emergence of edge computing. Computer 50(1):30–39
28. Ren L, Shi W, Yu Z, Liu Z (2016) Real-time energy-efficient fall detection based on SSR enery efficiency strategy. Int J Sens Netw 20(4):243–251
29. Solaz M, Bourke A, Conway R, Nelson J, OLaighin G, (2010) Real-time low-energy fall detection algorithm with a programmable truncated MAC. In: 32nd annual international conference of the IEEE EMBS. Buenos Aires, Argentina, pp 2423–2426
30. Ganz F, Barnaghi P, Carrez F (2013) Information abstraction for heterogeneous real world internet data. IEEE Sens J 13(10):3793–3805

31. Gia TN, Thanigaivelan NK, Rahmani AM, Westerlund T, Liljeberg P, Tenhunen H (2014) Customizing 6LoWPAN networks towards Internet-of-Things based ubiquitous healthcare systems. In: Proceeding of NORCHIP, Tampere, Finland, pp 1–6
32. Bai Y, Hao P, Zhang Y (2018) A case for web service bandwidth reduction on mobile devices with edge-hosted personal services. In: IEEE INFOCOM 2018, Honolulu, USA
33. Traub J, Breß RT, Katsifodimos A, Markl V (2017) Optimized on-demand data streaming from sensor nodes. In: SoCC'17, Santa Clara, USA
34. Trihinas D, Pallis G, Dikaiakos MD (2018) Low-cost adaptive monitoring techniques for the Internet of Things. IEEE Trans Serv Comput. https://doi.org/10.1109/TSC.2018.2808956
35. Trihinas D, Pallis G, Dikaiakos MD (2017) ADMin: adaptive monitoring dissemination for the Internet of Things. In: IEEE INFOCOM 2017, Atlanta, GA, USA
36. Murtha J (2018) How edge computing can advance healthcre. http://www.hcanews.com/news/how-edge-computing-can-advance-healthcare. Accessed 22 Jan 2018 (Online)
37. Rahmani AM, Gia TN, Negash B, Anzanpour A, Azimi I, Jiang M, Liljeberg P (2017) Exploiting smart e-health gateways at the edge of healthcare Internet-of Things: a fog computing approach. Future Gener Comput Syst 78(2018):641–658
38. Apexcz (2018) Fall activity detection. https://github.com/apexcz/FallActivityDetection. Accessed 12 June 2018 (Online)
39. Barik RK, Dubey H, Mankodiya K (2017) SOA-FOG: secure service-oriented edge computing architecture for smart health big data analytics. In: 5th IEEE global conference on signal and information processing, GlobalSIP 2017, Montreal, Canada
40. Wu X, Dunne R, Zhang Q, Shi W (2017) Edge computing enabled smart firefighting: opportunities and challenges. In: HotWeb17, CA, USA
41. Zhang Q, Zhang Q, Zhong H (2017) Poster: enhancing AMBER alert using collaborative edges. In: SEC17, CA, USA
42. Nelson P (2016) Just one autonomous car will use 4000 gb of data/day. http://www.networkworld.com/article/3147892/internet/one-autonomous-car-will-use-4000-gb-of-dataday.html. Accessed Mar 2018 (Online)
43. Zhang Q, Wang Y, Zhang X, Liu L, Wu X, Shi W, Zhong H (2018) OpenVDAP: an open vehicular data analytics platform for CAVs. In: Proceedings of the 38th IEEE international conference on distributed computing systems (ICDCS), Vision/Blue Sky Track, Vienna, Austria
44. Sundar S, Liang B (2018) Offloading dependent tasks with communication delay and deadline constraint. In: IEEE INFOCOM 2018, Honolulu, USA
45. Wang H, Gong J, Zhuang Y, Shen H, Lach J (2017) HealthEdge: task scheduling for edge computing with health emergency and human behavior consideration in smart homes. In: International conference on networking, architecture, and storage (NAS), Shenzhen, China, pp 1–2
46. Roman R, Zhou J, Lopez J (2013) On the features and challenges of security and privacy in distributed Internet of Things. Comput Netw 57(10):2266–2279

Affiliations

Lingmei Ren[1] · Qingyang Zhang[2] · Weisong Shi[3] · Yanjun Peng[4]

[1] Shandong University of Science and Technology, Qingdao, Shandong, China

[2] Anhui University, Hefei, Anhui, China

[3] Wayne State University, Detroit, MI, USA

[4] Key Laboratory for Wisdom Mine Information Technology of Shandong Province, Shandong University of Science and Technology, Qingdao, Shandong, China