WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2013; **13**:1587–1611
Published online 11 October 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.1203

RESEARCH ARTICLE

A survey of mobile cloud computing: architecture, applications, and approaches

Hoang T. Dinh, Chonho Lee, Dusit Niyato* and Ping Wang

School of Computer Engineering, Nanyang Technological University (NTU), Singapore

ABSTRACT

Together with an explosive growth of the mobile applications and emerging of cloud computing concept, mobile cloud computing (MCC) has been introduced to be a potential technology for mobile services. MCC integrates the cloud computing into the mobile environment and overcomes obstacles related to the performance (e.g., battery life, storage, and bandwidth), environment (e.g., heterogeneity, scalability, and availability), and security (e.g., reliability and privacy) discussed in mobile computing. This paper gives a survey of MCC, which helps general readers have an overview of the MCC including the definition, architecture, and applications. The issues, existing solutions, and approaches are presented. In addition, the future research directions of MCC are discussed. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS

mobile cloud computing; offloading; mobile services

***Correspondence**

Dusit Niyato, School of Computer Engineering, Nanyang Technological University (NTU), Singapore.
E-mail: dniyato@ntu.edu.sg

1. INTRODUCTION

Mobile devices (e.g., smartphone and tablet PC) are increasingly becoming an essential part of human life as the most effective and convenient communication tools not bounded by time and place. Mobile users accumulate rich experience of various services from mobile applications (e.g., iPhone apps and Google apps), which run on the devices and/or on remote servers via wireless networks. The rapid progress of mobile computing (MC) [1] becomes a powerful trend in the development of IT technology as well as commerce and industry fields. However, the mobile devices are facing many challenges in their resources (e.g., battery life, storage, and bandwidth) and communications (e.g., mobility and security) [2]. The limited resources significantly impede the improvement of service qualities.

Cloud computing (CC) has been widely recognized as the next generation computing infrastructure. CC offers some advantages by allowing users to use infrastructure (e.g., servers, networks, and storages), platforms (e.g., middleware services and operating systems), and softwares (e.g., application programs) provided by cloud providers (e.g., Google, Amazon, and Salesforce) at low cost. In addition, CC enables users to elastically utilize resources in an on-demand fashion. As a result, mobile applications can be rapidly provisioned and released with the minimal management efforts or service provider's interactions. With the explosion of mobile applications and the support of CC for a variety of services for mobile users, mobile cloud computing (MCC) is introduced as an integration of CC into the mobile environment. MCC brings new types of services and facilities mobile users to take full advantages of CC.

This paper presents a comprehensive survey on MCC. Section 2 provides a brief overview of MCC including definition, architecture, and its advantages. Section 3 discusses the use of MCC in various applications. Then, Section 4 presents several issues that arise in MCC and approaches to address the issues. Next, the future research directions are outlined in Section 5. Finally, we summarize and conclude the survey in Section 6. The list of acronyms appeared in this paper is given in Table I.

2. OVERVIEW OF MOBILE CLOUD COMPUTING

The term 'mobile cloud computing' was introduced not long after the concept of 'cloud computing'. It has been attracting the attentions of entrepreneurs as a profitable business option that reduces the development and running cost of mobile applications, of mobile users as a

Table I. Acronyms

4G	Fourth Generation
AAA	Authentication, Authorization, Accounting
APDV	Application Protocol Data Unit
API	Application Programing Interface
ARM	Advanced RISC Machine
AV	Antivirus
B2B	Business to Business
B2C	Business to Customer
BTS	Base Transceiver Station
CC	Cloud Computing
CSP	Cloud Service Provider
EC2	Elastic Compute Cloud
GPS	Global Positioning System
HA	Home Agent
IaaS	Infrastructure as a Service
IA	Integrated Authenticated
ID	Identifier
IMERA	French acronym for Mobile Interaction in Augmented Reality Environment
ISP	Internet Service Provider
IRNA	Intelligent Radio Network Access
JME	Java ME, a Java platform
LBS	Location Base Service
LTE	Long Term Evolution
LTS	Location Trusted Server
MAUI	Memory Arithmetic Unit and Interface
MC	Mobile Computing
MCC	Mobile Cloud Computing
MDP	Markov Decision Process
MSC	Mobile Service Cloud
P2P	Peer-to-Peer
PaaS	Platform as a Service
QoS	Quality of Service
RACE	Resource-Aware Collaborative Execution
REST	Repretational State Transfer
RFS	Random File System
RTP	Real-time Transport Protocol
S3	Simple Storage Service
SaaS	Software as a Service
TCC	Truster Crypto Coprocessor
URI	Uniform Resource Identifier

new technology to achieve rich experience of a variety of mobile services at low cost, and of researchers as a promising solution for green IT [3]. This section provides an overview of MCC including definition, architecture, and advantages of MCC.

2.1. What is mobile cloud computing?

The MCC forum defines MCC as follows [4]:

'Mobile cloud computing at its simplest, refers to an infrastructure where both the data storage and data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and MC to not just smartphone users but a much broader range of mobile subscribers'.

Aepona [5] describes MCC as a new paradigm for mobile applications whereby the data processing and storage are moved from the mobile device to powerful and centralized computing platforms located in clouds. These centralized applications are then accessed over the wireless connection based on a thin native client or web browser on the mobile devices.

Alternatively, MCC can be defined as a combination of mobile web and CC [6,7], which is the most popular tool for mobile users to access applications and services on the Internet.

Briefly, MCC provides mobile users with the data processing and storage services in clouds. The mobile devices do not need a powerful configuration (e.g., CPU speed and memory capacity) because all the complicated computing modules can be processed in the clouds.

2.2. Architectures of mobile cloud computing

From the concept of MCC, the general architecture of MCC can be shown in Figure 1. In Figure 1, mobile devices are connected to the mobile networks via base stations (e.g., base transceiver station, access point, or satellite) that establish and control the connections (air links) and functional interfaces between the networks and mobile devices. Mobile users' requests and information (e.g., ID and location) are transmitted to the central processors that are connected to servers providing mobile network services. Here, mobile network operators can provide services to mobile users as authentication, authorization, and accounting based on the home agent and subscribers' data stored in databases. After that, the subscribers' requests are delivered to a cloud through the Internet. In the cloud, cloud controllers process the requests to provide mobile users with the corresponding cloud services. These services are developed with the concepts of utility computing, virtualization, and service-oriented architecture (e.g., web, application, and database servers).

The details of cloud architecture could be different in different contexts. For example, a four-layer architecture is explained in [8] to compare cloud computing with grid computing. Alternatively, a service-oriented architecture, called Aneka, is introduced to enable developers to build. Microsoft .NET applications with the supports of application programming interfaces (APIs) and multiple programming models [9]. [10] presents an architecture for creating market-oriented clouds and [11] proposes an architecture for web-delivered business services. In this paper, we focus on a layered architecture of CC (Figure 2). This architecture is commonly used to demonstrate the effectiveness of the CC model in terms of meeting the user's requirements [12].

Generally, a CC is a large-scale distributed network system implemented based on a number of servers in data centers. The cloud services are generally classified based on a layer concept (Figure 2). In the upper layers of this paradigm, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) are stacked.

- *Data centers layer.* This layer provides the hardware facility and infrastructure for clouds. In data center layer, a number of servers are linked with high-speed networks to provide services for customers. Typically, data centers are built in less populated places, with a high power supply stability and a low risk of disaster.
- *IaaS.* Infrastructure as a Service is built on top of the data center layer. IaaS enables the provision of storage, hardware, servers, and networking components. The client typically pays on a per-use basis. Thus, clients can save cost as the payment is only based on how much resource they really use. Infrastructure can be expanded or shrunk dynamically as needed. The examples of IaaS are Amazon Elastic Cloud Computing and Simple Storage Service (S3).
- *PaaS.* Platform as a Service offers an advanced integrated environment for building, testing, and deploying custom applications. The examples of PaaS are

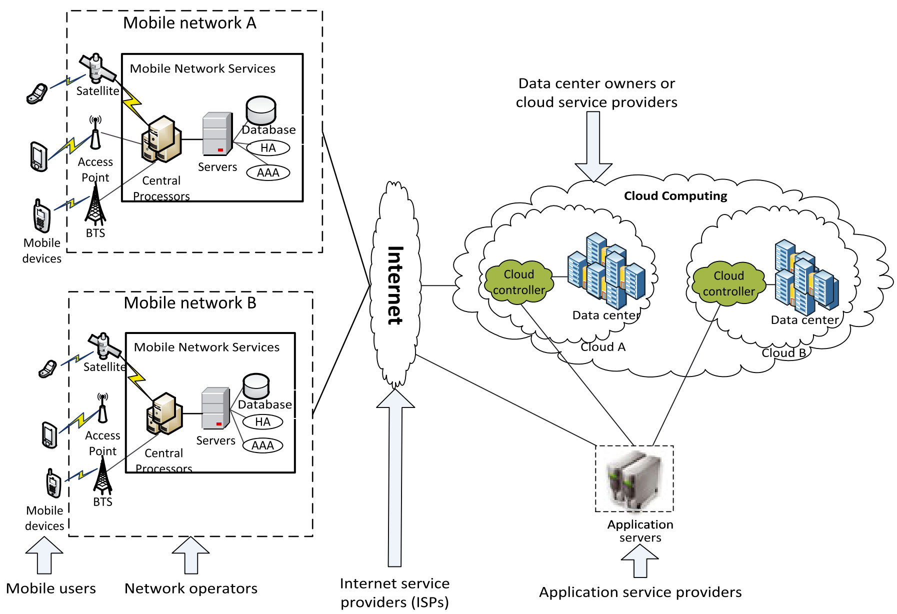

Figure 1. Mobile cloud computing architecture.

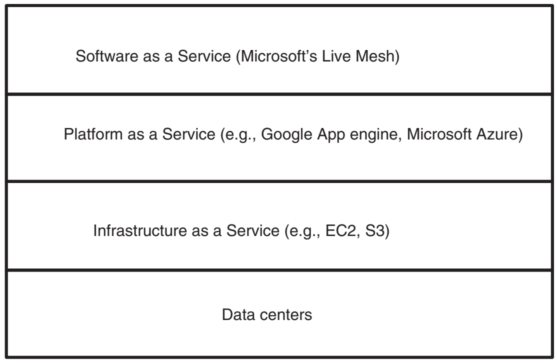

Figure 2. Service-oriented cloud computing architecture.

Google App Engine, Microsoft Azure, and Amazon Map Reduce/Simple Storage Service.

- *SaaS.* Software as a Service supports a software distribution with specific requirements. In this layer, the users can access an application and information remotely via the Internet and pay only for that they use. Salesforce is one of the pioneers in providing this service model. Microsoft's Live Mesh also allows sharing files and folders across multiple devices simultaneously.

Although the CC architecture can be divided into four layers as shown in Figure 2, it does not mean that the top layer must be built on the layer directly below it. For example, the SaaS application can be deployed directly on IaaS, instead of PaaS. Also, some services can be considered as a part of more than one layer. For example, data storage service can be viewed as either in IaaS or PaaS. Given this architectural model, the users can use the services flexibly and efficiently.

2.3. Advantages of mobile cloud computing

Cloud computing is known to be a promising solution for MC because of many reasons (e.g., mobility, communication, and portability [13]). In the following, we describe how the cloud can be used to overcome obstacles in MC, thereby pointing out advantages of MCC.

(1) *Extending battery lifetime.* Battery is one of the main concerns for mobile devices. Several solutions have been proposed to enhance the CPU performance [14,15] and to manage the disk and screen in an intelligent manner [16,17] to reduce power consumption. However, these solutions require changes in the structure of mobile devices, or they require a new hardware that results in an increase of cost and may not be feasible for all mobile devices. Computation offloading technique is proposed with the objective to migrate the large computations and complex processing from resource-limited devices (i.e., mobile devices) to resourceful machines (i.e., servers in clouds). This avoids taking a long application execution time on mobile devices which results in large amount of power consumption.

Rudenko *et al.* [18] and Smailagic and Ettus [19] evaluate the effectiveness of offloading techniques through several experiments. The results demonstrate that the remote application execution can save energy significantly. Especially, Rudenko *et al.* [18] evaluates large-scale numerical computations and shows that up to 45% of energy consumption can be reduced for large matrix calculation. In addition, many mobile applications take advantages from task migration and remote processing. For example, offloading a compiler optimization for image processing [20] can reduce 41% for energy consumption of a mobile device. Also, using memory arithmetic unit and interface (MAUI) to migrate mobile game components [21] to servers in the cloud can save 27% of energy consumption for computer games and 45% for the chess game.

(2) *Improving data storage capacity and processing power.* Storage capacity is also a constraint for mobile devices. MCC is developed to enable mobile users to store/access the large data on the cloud through wireless networks. First example is the Amazon Simple Storage Service [22] which supports file storage service. Another example is Image Exchange which utilizes the large storage space in clouds for mobile users [23]. This mobile photo sharing service enables mobile users to upload images to the clouds immediately after capturing. Users may access all images from any devices. With the cloud, the users can save considerable amount of energy and storage space on their mobile devices because all images are sent and processed on the clouds. Flicker [24] and ShoZu [25] are also the successful mobile photo sharing applications based on MCC. Facebook [26] is the most successful social network application today, and it is also a typical example of using cloud in sharing images.

 Wirel. Commun. Mob. Comput. 2013; **13**:1587–1611 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

Mobile cloud computing also helps in reducing the running cost for compute-intensive applications that take long time and large amount of energy when performed on the limited-resource devices. CC can efficiently support various tasks for data warehousing, managing and synchronizing multiple documents online. For example, clouds can be used for transcoding [27], playing chess [21,28], or broadcasting multimedia services [29] to mobile devices. In these cases, all the complex calculations for transcoding or offering an optimal chess move that take a long time when perform on mobile devices will be processed efficiently on the cloud. Mobile applications also are not constrained by storage capacity on the devices because their data now is stored on the cloud.

(3) *Improving reliability.* Storing data or running applications on clouds is an effective way to improve the reliability because the data and application are stored and backed up on a number of computers. This reduces the chance of data and application lost on the mobile devices. In addition, MCC can be designed as a comprehensive data security model for both service providers and users. For example, the cloud can be used to protect copyrighted digital contents (e.g., video, clip, and music) from being abused and unauthorized distribution [30]. Also, the cloud can remotely provide to mobile users with security services such as virus scanning, malicious code detection, and authentication [31]. Also, such cloud-based security services can make efficient use of the collected record from different users to improve the effectiveness of the services.

In addition, MCC also inherits some advantages of clouds for mobile services as follows:

- *Dynamic provisioning*. Dynamic on-demand provisioning of resources on a fine-grained, self-service basis is a flexible way for service providers and mobile users to run their applications without advanced reservation of resources.
- *Scalability*. The deployment of mobile applications can be performed and scaled to meet the unpredictable user demands due to flexible resource provisioning. Service providers can easily add and expand an application and service without or with little constraint on the resource usage.
- *Multitenancy*. Service providers (e.g., network operator and data center owner) can share the resources and costs to support a variety of applications and large number of users.
- *Ease of integration*. Multiple services from different service providers can be integrated easily through the cloud and Internet to meet the user demand.

3. APPLICATIONS OF MOBILE CLOUD COMPUTING

Mobile applications gain increasing share in a global mobile market. Various mobile applications have taken the advantages of MCC. In this section, some typical MCC applications are introduced.

3.1. Mobile commerce

Mobile commerce (m-commerce) is a business model for commerce using mobile devices. The m-commerce applications generally fulfill some tasks that require mobility (e.g., mobile transactions and payments, mobile messaging, and mobile ticketing). The m-commerce applications can be classified into few classes including finance, advertising, and shopping (Table II).

The m-commerce applications have to face various challenges (e.g., low network bandwidth, high complexity of mobile device configurations, and security). Therefore, m-commerce applications are integrated into CC environment to address these issues. Yang *et al.*[32] proposes a 3G E-commerce platform based on CC. This paradigm combines the advantages of both third generation (3G) network and CC to increase data processing speed and security level [33] based on public key infrastructure (PKI). The PKI mechanism uses an encryption-based access control and an over-encryption to ensure privacy of user's access to the outsourced data. In [34], a 4PL-AVE trading platform utilizes CC technology to enhance the security for users and improve the customer satisfaction, customer intimacy, and cost competitiveness.

3.2. Mobile learning

Mobile learning (m-learning) is designed based on electronic learning (e-learning) and mobility. However, traditional m-learning applications have limitations in terms of high cost of devices and network, low network transmission rate, and limited educational resources [35–37].

Table II. Application classes of m-commerce.

Application classes	Type	Examples
Mobile financial applications	B2C, B2B	Banks, brokerage firms, mobile-user fees
Mobile advertising	B2C	Sending custom made advertisements according to user's physical location
Mobile shopping	B2C,B2B	Locate/order certain products from a mobile terminal

B2C, business to customer; B2B, business to business

Cloud-based m-learning applications are introduced to solve these limitations. For example, utilizing a cloud with the large storage capacity and powerful processing ability, the applications provide learners with much richer services in terms of data (information) size, faster processing speed, and longer battery life.

Zhao *et al.* [38] presents the benefits of combining m-learning and CC to enhance the communication quality between students and teachers. In this case, a smartphone software based on the open source JavaME UI framework and Jaber for clients is used. Through a web site built on Google Apps Engine, students communicate with their teachers at anytime. Also, the teachers can obtain the information about student's knowledge level of the course and can answer students' questions in a timely manner. In addition, a contextual m-learning system based on Mobile Interaction in Augmented Reality Environment platform [39] shows that a cloud-based m-learning system helps learners access learning resources remotely.

Another example of MCC applications in learning is 'Cornucopia' implemented for researches of undergraduate genetics students and 'plantations pathfinder' designed to supply information and provide a collaboration space for visitors when they visit the gardens [40]. The purpose of the deployment of these applications is to help the students enhance their understanding about the appropriate design of MCC in supporting field experiences. In [41], an education tool is developed based on CC to create a course about image/video processing. Through mobile phones, learners can understand and compare different algorithms used in mobile applications (e.g., deblurring, denoising, face detection, and image enhancement).

3.3. Mobile healthcare

The purpose of applying MCC in medical applications is to minimize the limitations of traditional medical treatment (e.g., small physical storage, security and privacy, and medical errors [42,43]). Mobile healthcare (m-healthcare) provides mobile users with convenient helps to access resources (e.g., patient health records) easily and efficiently. Besides, m-healthcare offers hospitals and healthcare organizations a variety of on-demand services on clouds rather than owning standalone applications on local servers.

There are a few schemes of MCC applications in healthcare. For example, [44] presents five main mobile healthcare applications in the pervasive environment.

- *Comprehensive health monitoring services* enable patients to be monitored at anytime and anywhere through broadband wireless communications.
- *Intelligent emergency management system* can manage and coordinate the fleet of emergency vehicles effectively and in time when receiving calls from accidents or incidents.
- *Health-aware mobile devices* detect pulse rate, blood pressure, and level of alcohol to alert healthcare emergency system.
- *Pervasive access to healthcare information* allows patients or healthcare providers to access the current and past medical information.
- *Pervasive lifestyle incentive management* can be used to pay healthcare expenses and manage other related charges automatically.

Similarly, [45] proposes @HealthCloud, a prototype implementation of m-healthcare information management system based on CC and a mobile client running Android operating system (OS). This prototype presents three services utilizing the Amazon's S3 Cloud Storage Service to manage patient health records and medical images.

- *Seamless connection to cloud storage* allows users to retrieve, modify, and upload medical contents (e.g., medical images, patient health records, and biosignals) utilizing web services and a set of available APIs called Repretational State Transfer.
- *Patient health record management system* displays the information regarding patients' status, related biosignals, and image contents through application's interface.
- *Image viewing support* allows the mobile users to decode the large image files at different resolution levels given different network availability and quality.

For practical system, a telemedicine homecare management system [46] is implemented in Taiwan to monitor participants, especially for patients with hypertension and diabetes. The system monitors 300 participants and stores more than 4736 records of blood pressure and sugar measurement data on the cloud. When a participant performs blood glucose/pressure measurement via specialized equipment, the equipment can send the measured parameters to the system automatically. Also, the participant can send parameters by SMS via their mobile devices. After that, the cloud will gather and analyze the information about the participant and return the results. The development of mobile healthcare clearly provides tremendous helps for the participants. However, the information to be collected and managed related to personal health is sensitive. Therefore, [47,48] propose solutions to protect the participant's health information, thereby, increasing the privacy of the services. Although [47] uses peer-to-peer paradigm to federate clouds to address security issue, data protection, and ownership, the model in [48] provides security as a service on the cloud to protect mobile applications. Therefore, mobile health application providers and users will not have to worry about security issue because it is ensured by the security vendor.

3.4. Mobile gaming

Mobile game (m-game) is a potential market generating revenues for service providers. M-game can completely offload game engine requiring large computing resource (e.g., graphic rendering) to the server in the cloud, and gamers only interact with the screen interface on their devices.

Li *et al.*[49] demonstrates that offloading (multimedia code) can save energy for mobile devices, thereby increasing game playing time on mobile devices. Cuervo *et al.*[21] proposes MAUI, a system that enables fine-grained energy-aware offloading of mobile codes to a cloud. Also, a number of experiments are conducted to evaluate the energy used for game applications with 3G network and WiFi network. It is found that instead of offloading all codes to the cloud for processing, MAUI partitions the application codes at a runtime based on the costs of network communication and CPU on the mobile device to maximize energy savings given network connectivity. The results demonstrate that MAUI not only helps energy reduction significantly for mobile devices (i.e., MAUI saves 27% of energy usage for the video game and 45% for chess), but also improves the performance of mobile applications (i.e., the game's refresh rate increases from 6 to 13 frames per second).

Wang and Dey [50] presents a new cloud-based m-game using a rendering adaptation technique to dynamically adjust the game rendering parameters according to communication constraints and gamers' demands. The rendering adaptation technique mainly bases on the idea to reduce the number of objects in the display list because not all objects in the display list created by game engine are necessary for playing the game and scale the complexity of rendering operations. The objective is to maximize the user experience given the communications and computing costs.

3.5. Other practical applications

A cloud becomes a useful tool to help mobile users share photos and video clips efficiently and tag their friends in popular social networks as Twitter and Facebook. MeLog [51] is an MCC application that enables mobile users to share real-time experience (e.g., travel, shopping, and event) over clouds through an automatic blogging. The mobile users (e.g., travelers) are supported by several cloud services such as guiding their trip, showing maps, recording itinerary, and storing images and video.

Ye *et al.*[52] introduces a mobile locationing service allowing users to capture a short video clip about the surrounding buildings. The matching algorithm run on a cloud can use a large amount of information to search for a location of these buildings. Also, One Hour Translation [53] provides an online translation service running on the cloud of Amazon Web Services. One Hour Translation helps mobile users, especially foreign visitors, receive the information translated in their language through their mobile devices.

A cloud becomes the most effective tool when mobile users require searching services (e.g., searching information, location, images, voices, or video clips).

- *Keyword-based searching.* Pendyala and Holliday [54] proposes an intelligent mobile search model using semantic in which searching tasks will be performed on servers in a cloud. This model can analyze the meaning of a word, a phrase, or a complex multiphase to produce the results efficiently and accurately. Lagerspetz and Tarkoma [55] presents an application using the cloud to perform data searching tasks for mobile users. Lagerspetz and Tarkoma [55] uses Dessy system [56] to find the users' data, metadata, and context information through desktop search (e.g., indexing, query, and index term stemming, and search relevance ranking), and synchronization techniques.
- *Voice-based searching.* Fabbrizio *et al.* [57] proposes a search service via a speech recognition in which mobile users just talk to microphone on their devices rather than typing on keypads or touchscreens. Fabbrizio *et al.* [57] introduces the AT&T speech mashup model that utilizes web services and CC environment to meet the speech service demands of customers. This model optimizes the data transmission in a mobile network, reduces latency, and is flexible in integrating with other services. Several examples are demonstrated (e.g., speak4it, iPizza, and JME local business search).
- *Tag-based searching.* Cai-Dong *et al.* [58] introduces a photo searching technique based on ontological semantic tags. Mobile users search only recall parameters that are tagged on images before such images are sent to a cloud. The cloud is used for storing and processing images for resource-limited devices. The current service is designed for the images stored on private CC environment. In the future, it is expected to expand for searching images in a public cloud environment.

In addition, there are a mobile-cloud collaborative application [59] to detect traffic lights for the blind, a CC framework [60] to monitor different corners in a house through a mobile device, and some efforts which integrate current services (e.g., BitTorrent, and Mobile Social Network) into the clouds as in [61,62]. Thereby, we can recognize that MCC is probably a prevailing technology trend with numerous applications in the near future.

4. ISSUES AND APPROACHES OF MOBILE CLOUD COMPUTING

As discussed in the previous section, MCC has many advantages for mobile users and service providers. However, because of the integration of two different fields, that is, CC and mobile networks, MCC has to face many technical challenges. This section lists several research issues in MCC, which are related to the mobile communication and CC. Then, the available solutions to address these issues are reviewed.

4.1. Issues in mobile communication side

(1) *Low bandwidth.* Bandwidth is one of the big issues in MCC because the radio resource for wireless networks is much scarce as compared with the traditional wired networks.

Jin and Kwok [63] proposes a solution to share the limited bandwidth among mobile users who are located in the same area (e.g., a workplace, a station, and a stadium) and involved in the same content (e.g., a video file). The authors model the interaction among the users as a coalitional game. For example, the users form a coalition where each member is responsible for a part of video files (e.g., sounds, images, and captions) and transmits/exchanges it to other coalition members. This results in the improvement of the video quality. However, the proposed solution is only applied in the case when the users in a certain area are interested in the same contents. Also, it does not consider a distribution policy (e.g., who receives how much and which part of contents) which leads to a lack of fairness about each user's contribution to a coalition.

Jung *et al.*[64] considers the data distribution policy which determines when and how much portions of available bandwidth are shared among users from which networks (e.g., WiFi and WiMAX). It collects user profiles (e.g., calling profile, signal strength profile, and power profile) periodically and creates decision tables by using Markov decision process algorithm. Based on the tables, the users decide whether or not to help other users download some contents that they cannot receive by themselves due to the bandwidth limitation and how much it should help (e.g., 10% of contents). The authors build a framework, named RACE (Resource-Aware Collaborative Execution), on the cloud to take advantages of the computing resources for maintaining the user profiles. This approach is suitable for the users who share the limited bandwidth to balance the trade-off between benefits of the assistance and energy costs.

(2) *Availability.* Service availability becomes a more important issue in MCC than that in the CC with wired networks. Mobile users may not be able to connect to the cloud to obtain a service due to traffic congestion, network failures, and the out-of-signal.

Huerta-Canepa and Lee [65] and Zhang *et al.*[66] propose solutions to help mobile users in the case of the disconnection from clouds. In [65], the authors describe a discovery mechanism to find the nodes in the vicinity of a user whose link to the cloud is unavailable. After detecting nearby nodes that are in a stable mode, the target provider for the application is changed. In this way, instead of having a link directly to the cloud, a mobile user can connect to the cloud through neighboring nodes in an ad hoc manner. However, it does not consider the mobility, capability of devices, and privacy of neighboring nodes.

Zhang *et al.* [66] tries to overcome the drawbacks of [65]. In particular, [66] proposes a WiFi based multihop networking system called MoNet and a distributed content sharing protocol for the situation without any infrastructure. Unlike [65], this solution considers moving nodes in the user's vicinity. Each node periodically broadcasts control messages to inform other nodes of its status (e.g., connectivity and system parameters) and local content updates. According to the messages, each node maintains a neighboring node list and a content list and estimates role levels of other nodes based on the disk space, bandwidth, and power supply. Then, the nodes with the shortest hop length path and the highest role level are selected as the intermediate nodes to receive contents. Besides, the authors also consider security issues for mobile clients when they share information by using an account key (to authenticate and encrypt the private content), a friend key (to secure channel between two friends), and a content key (to protect an access control). Two applications are introduced; that is, WiFace and WiMarket that are two colocated social networking. This approach is much more efficient than the current social networking systems, especially in the event of disconnection.

(3) *Heterogeneity.* Mobile cloud computing will be used in the highly heterogeneous networks in terms of wireless network interfaces. Different mobile nodes access to the cloud through different radio access technologies such as WCDMA, GPRS, WiMAX, CDMA2000, and WLAN. As a result, an issue of how to handle the wireless connectivity while satisfying MCC's requirements arises (e.g., always-on connectivity, on-demand scalability of wireless connectivity, and the energy efficiency of mobile devices).

Klein *et al.*[67] proposes an architecture to provide an intelligent network access strategy for mobile users to meet the application requirements. This architecture is built based on a concept of Intelligent Radio Network Access (IRNA [68]). IRNA is an effective model to deal with the dynamics and heterogeneity of available access networks. To apply IRNA in MCC environment, the authors propose a context management architecture with the purpose to acquire, manage, and distribute a context information. As shown in Figure 3, this architecture consists of three main components: context provider, context broker, and context consumer. However, the context quality enabler is also required to facilitate the operations of other components. In this architecture, when a context consumer wants to communicate with a context provider, the context consumer will request the Uniform Resource Identifier (URI) of context providers at

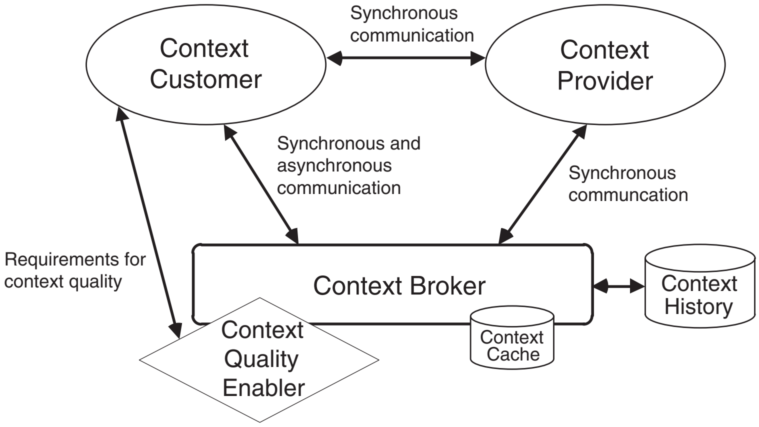

Figure 3. Context management architecture introduced in [67].

the context broker. Using this URI, the context consumer can communicate directly to the context provider and request the context data. Hence, this process increases the speed of context data delivery. Furthermore, when a context quality enabler receives the requirement about the context quality from the context consumer, the context quality enabler will filter out URIs of the context providers that are not suitable with the required quality level. Therefore, this architecture enables controlling context quality according to the demands of the context consumers.

4.2. Issues in computing side

(1) *Computing offloading.* As explained in the previous section, offloading is one of the main features of MCC to improve the battery lifetime for the mobile devices and to increase the performance of applications. However, there are many related issues including efficient and dynamic offloading under environment changes.

(a) *Offloading in the static environment.* Experiments in [18] show that offloading is not always the effective way to save energy. For a code compilation, offloading might consume more energy than that of local processing when the size of codes is small. For example, when the size of altered codes after compilation is 500 KB, offloading consumes about 5% of a device's battery for its communication, whereas the local processing consumes about 10% of the battery for its computation. In this case, the offloading can save the battery up to 50%. However, when the size of altered codes is 250 KB, the efficiency reduces to 30%. When the size of altered codes is small, the offloading consumes more battery than that of local processing. As another example, [18] shows the Gaussian application (to solve a system of linear algebraic equations) which offloads the entire matrix into the remote server. In terms of the energy efficiency, the cost of offloading is higher for small matrices (e.g., smaller than 500 x 500 in size), whereas the cost saving can be up to 45% for large matrices. Therefore, it is a critical problem for mobile devices to determine whether to offload and which portions of the application's codes need to be offloaded to improve the energy efficiency. In addition, different wireless access technologies consume different amount of energy and support different data transfer rates. These factors have to be taken into account.

Kumar and Lu [28] suggests a program partitioning based on the estimation of the energy consumption (communication energy and computation energy) before the program execution. The optimal program partitioning for offloading is calculated based on the trade-off between the communication and computation costs. The communication cost depends on the size of transmitted data and the network bandwidth, whereas the computation cost is impacted by the computation time. However, information such as the communication requirements or/and the computation workload may change in different execution instances. Thus, optimal decisions of a program partitioning must be made at a runtime dynamically.

Several solutions are proposed to find the optimal decision for partitioning applications before offloading. In [49], the authors present a partition scheme to offload computational tasks on mobile devices. The scheme is based on the profiling information about computation time and data sharing at the level of procedure calls. This scheme constructs a cost graph. Then branch-and-bound algorithm [69] is applied to the cost graph with an objective to minimize the total energy consumption of computation and the total data communication cost. The idea of this algorithm is to prune the search space to obtain an approximated solution. The experiment results indicate that the energy saving of partitioning obtained from this scheme is considerable in some programs (e.g., Mediabench programs and GNU Go for a Go game). However, the authors do not show the experiment results in a dynamic environment such as network disconnection and bandwidth changes (high to low bandwidth). Also, this approach considers only partition for tasks which are procedure calls.

In [70], the authors present an approach to decide which components of Java programs should be offloaded. This approach first divides a Java program into methods and uses input parameters (e.g., size of methods) to compute execution costs for these methods. Then, this approach compares the local execution costs of each method with the remote execution costs that are estimated based on status of the current wireless channel condition to make an optimal execution decision. Similar to [49], the approach in [70] lacks generality and cannot be applied for diverse applications.

Wang and Li [71] present a computation offloading scheme on mobile devices and proposes a polynomial time algorithm to find an optimal program partition. The proposed scheme partitions a program into the distributed subprograms (which run on a device and a server) by producing a program abstraction. In this case, all physical memory references are mapped into the references of abstract memory locations. The program abstraction is generated at a runtime based on pointer analysis techniques [72,73]. Then, the task allocation and data transfer of the abstract memory locations are determined subject to the control and data flow defined over the program abstraction. The program abstraction is divided into clusters by clustering analysis [74], and a heuristic algorithm [75] is applied into the clusters to find the optimal partition to minimize the execution cost of the program. Unlike [49,70], this scheme does not restrict the tasks to be partitioned at any specific level (e.g., a basic block, a loop, a function, or even a group of closely related functions).

Hunt and Scott [76] present an automatic distributed partitioning system called Coign, which automatically transforms a program into the distributed applications without accessing the source codes. As shown in Figure 4, Coign constructs a graph model of the application's intercomponent communication through the scenario-based profiling (e.g., network data) to find the best distribution. Coign applies the

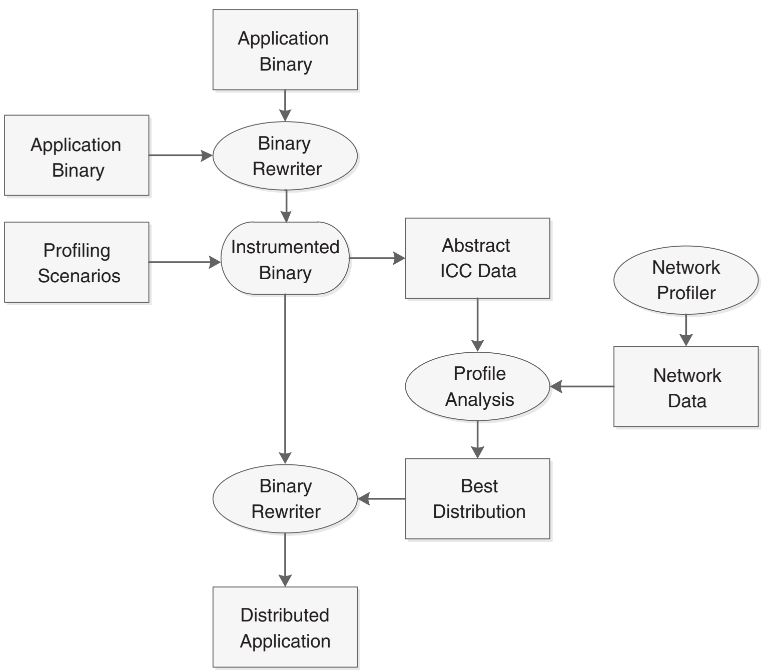

Figure 4. The Coign automatic distributed partitioning system: an application is transformed into a distributed application by inserting the Coign runtime, profiling the instrumented application, and analyzing the profiles to cut the network-based graph.

DOI: 10.1002/wcm

lift-to-front minimum-cut graph-cutting algorithm [77] to choose the distributed applications with the minimum communication time.

Most approaches use the data size and execution time of computations to find the optimal program partition for offloading and assume that such an information is known before the execution. However, it is difficult to obtain the accurate execution time of computations because the time varies in different instances of the computations, and the inaccurate information results in inefficient offloading performance. Therefore, [78] proposes an offloading method which does not require the estimation of execution time for each computation instance. Online statistics of the computation time are used to compute optimal timeout, and if the computation is not completed after the timeout, this computation will be offloaded to the server. Through experiments, it is shown that this approach not only addresses the inaccuracy in estimating the computing execution time but also saves up to 17% more energy than existing approaches.

(b) *Offloading in the dynamic environment.* This subsection introduces few approaches to deal with offloading in a dynamic network environment (e.g., changing connection status and bandwidth). The environment changes can cause additional problems. For example, the transmitted data may not reach the destination, or the data executed on the server will be lost when it has to be returned to the sender.

Ou *et al.* [79] analyzes the performance of offloading systems operating in wireless environments. In this work, the authors take into account three circumstances of executing an application, thereby estimating the efficiency of offloading. They are the cases when the application is performed locally (without offloading), performed in ideal offloading systems (without failures), and performed with the presence of offloading and failure recoveries. In the last case, when a failure occurs, the application will be re-offloaded. This approach only re-offloads the failed subtasks, thereby improving the execution time. However, this solution has some limitations. That is, the mobile environment is considered as a wireless ad hoc local area network (i.e., broadband connectivity is not supported). Also, during offloading execution, a disconnection of a mobile device is treated as a failure.

Tang and Cao [80] consider three common environmental changes shown in Table III and explains the suitable solutions for offloading in the different environments. For example, in the case of connection status (e.g., disconnection during the program execution) changes, the server will periodically check the connection status with the client and maintain the execution information about the particular running tasks. When the disconnection is recovered, the server will send the execution results for the client. If the server cannot reconnect to the client, the server will wait for the predefined time interval, and the tasks will be deleted. However, the drawback of these approaches is that they are only general solutions, and they do not mention a detailed method to address the dynamic partitioning issue; that is, how to partition application.

Chun and Maniatis [81] present a system to partition an application in dynamic environments. The proposed system follows three steps with different requirements related to the application structuring, partitioning choice, and security. In the application structuring step, the programs are structured to be seamlessly and dynamically executed between a mobile device and a cloud. To achieve this, both the client and cloud must have all parts of the application, and the application decides what modules to run at the client and at the server dynamically at a runtime. Secondly, in the partitioning choice step, the system will choose a suitable partitioning policy so that the total energy consumption is minimized. Finally, to address security issue, the authors point out that modules containing sensitive data will be executed locally. The sensitive data is marked based on the programmer annotations. This system considers

Table III. Common mobile computing environmental changes.

Changes	Priority level	Description
Client side power level	1	Power can be divided into sufficient and insufficient power levels, which will depend on the particular situation.
Connection status	2	The connection status can be faded, disconnected from the mobile network, or reconnected to the mobile network.
Bandwidth	3	The bandwidth varies from time to time and depends on several factors, such as the network traffic condition, and so on.

both partitioning application and security issue. However, it lacks accuracy because the partition is based on a prediction model through an off-line analysis.

Cuervo *et al.* [21] introduces an architecture to dynamically partition an application at a runtime in three steps. First, MAUI uses *code portability* to create two versions of a mobile application, one for the local execution on devices and the other for the remote execution in cloud. Besides, because today's smartphones typically use an instruction set architecture (Advanced RISC Machine, ARM) different from desktop and servers (x86), so MAUI is designed to execute the same program on different CPU architectures, preferably without access to the program source code. Second, MAUI uses *programing reflection* to identify which methods of the application are marked '*remoteable*' or not and *type safety* to extract only the program state needed by the '*remoteable*' methods. Then, MAUI sends the necessary program state to the cloud. There are some certain types of code that should not be marked '*remoteable*' attribute including: a code that implements the application's user interface, a code that interacts with I/O devices where such interaction is only possible on the mobile device, and a code that interacts with any external component that would be affected by a re-execution. Third, the MAUI profiles each method of an application and uses *serialization* to determine communication costs (e.g., due to the size of its state). Then, the MAUI combines three main factors including the communication cost, mobile device's energy consumption cost, and network status (e.g., bandwidth and latency) at a runtime to construct a linear programming formulation. MAUI can make optimal decisions for partitioning based on the linear programming formulation. The *serialization* enables MAUI's program partitioning to be highly dynamic. The authors find that MAUI can maximize the potential for energy savings through the fine-grained code offloading while minimizing the changes required to applications.

(2) *Security.* Protecting user privacy and data/application secrecy from adversary is a key to establish and maintain consumers' trust in the mobile platform, especially in MCC. In the following, the security-related issues in MCC are introduced in two categories: the security for mobile users and the security for data. Also, some solutions to address these issues are reviewed.

(a) *Security for mobile users.* Mobile devices such as cellular phone, personal digital assistant (PDA), and smartphone are exposed to numerous security threats like malicious codes (e.g., virus, worm, and Trojan horses) and their vulnerability. In addition, with mobile phones integrated global positioning system (GPS) device, they can cause privacy issues for subscribers. Two main issues are as follows:

- *Security for mobile applications.* Installing and running security softwares such as Kaspersky, McAfee, and AVG antivirus programs on mobile devices are the simplest ways to detect security threats (e.g., virus, worms, and malicious codes) on the devices. However, mobile devices are constrained in their processing and power; protecting them from the threats is more difficult than that for resourceful device (e.g., PC). For example, it is impossible to keep running the virus detection software on mobile devices. Oberheide *et al.*[31] presents an approach to move the threat detection capabilities to clouds. This paradigm is an extension of the existing cloud AV platform that provides an in-cloud service for malware detection. The platform consists of host agent and network service components [82,83]. Host agent is a lightweight process that runs on mobile devices, and its function is to inspect the file activity on a system (i.e., it is similar to the function of antivirus software). If an identified file is not available in a cache of previous analyzed files, this file will be sent to the in-cloud network service for verification. The second major component of CloudAV is a network service that is responsible for file verification. The network service will determine whether a file is malicious or not. The most advantage of this solution is that moving the detection capabilities to a network service enables the use of multiple antivirus engines in parallel by hosting them in virtualized containers. However, to apply CloudAV platform for the mobile environment, a mobile agent should be improved and customized to fit in the mobile devices. Oberheide *et al.* [31] builds a mobile agent to interact with the CloudAV network service for the Linux-based Maemo platform implemented on a Nokia N800 mobile device. The mobile agent is deployed in Python and uses the Dazuko [84] framework to interpose on the system events. In particular, the mobile agent requires only 170 lines of code. As a result, it is suitable with resource-limited mobile devices. To demonstrate the efficiency of using CC for detecting malicious

Wirel. Commun. Mob. Comput. 2013; **13**:1587–1611 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

softwares on mobile devices, [85] presents a paradigm in which attack detection for a smartphone is performed on a remote server in the cloud. Similarly, instead of running an antivirus program locally, the smartphone records only a minimal execution trace and transmits it to the security server in the cloud. This paradigm not only enhances the efficiency of detecting malware but also improves battery lifetime up to 30%.

- *Privacy.* With the advantages of GPS positioning devices, the number of mobile users using the location based services (LBS) increases. However, the LBS faces a privacy issue when mobile users provide private information such as their current location. This problem becomes even worse if an adversary knows the user's important information. Location trusted server (LTS) [86] is presented to address this issue. As shown in Figure 5, after receiving the mobile users' requests, LTS gathers their location information in a certain area and cloaks the information called 'cloaked region' based on a 'k-anonymity' concept [87] to conceal the user's information. The 'cloaked region' is sent to LBS, so LBS knows only general information about the users but cannot identify them. Wang and Wang [88] point out the problem that if LTS reveals the users' information, or if LTS colludes with LBS, the users' information will be in danger. The authors propose to generate the 'cloaked region' on mobile devices based on Casper cloaking algorithm [89]. Meanwhile, gathering the information of other users around the sender will be done on the cloud to reduce cost and improve speed and scalability. When launching the program on the sender's mobile device, the program will require the cloud to provide information about surrounding users. After that, the mobile client will generate the 'cloaked region' by itself and send 'cloaked region' to the LBS. In this way, both LTS and LBS cannot know the sender's information.

(b) *Securing data on clouds.* Although both mobile users and application developers benefit from storing a large amount of data/applications on a cloud, they should be careful of dealing with the data/applications in terms of their integrity, authentication, and digital rights. The data-related issues in MCC are as follows:

- *Integrity.* Mobile users are often concerned about their data integrity on the cloud. Several solutions are proposed to address this issue (e.g., [90,91]). However, such solutions do not take the energy consumption of mobile users into account. Itani *et al.*[92] considers the energy consumption issue. This scheme consists of three main components: a mobile client, a cloud storage service, and a trusted third party. The scheme performs three phases: the initialization, update, and verification. In the first phase, files (F_x) that need to be sent to the cloud will be assigned with a message authentication code (MAC_{F_x}). These MAC_{Fx} will be stored locally, while the files will be sent and stored on the cloud. In the update phase, a case when a user wants to insert the data into file (F_x) is considered. The cloud then sends file (F_x) to this user. At the same time, the cloud also sends a requirement to the trusted crypto coprocessor (TCC) to generate MAC'_{F_x}. TCC then sends MAC'_{F_x} to the client to verify F_x by comparing it with MAC_{F_x}. If everything is properly authenticated, the user can insert/delete data. Finally, the mobile client can request

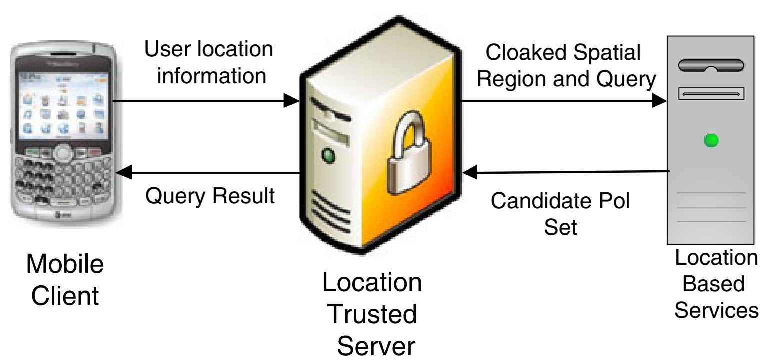

Figure 5. Overall Architecture of Spatial Cloaking.

the integrity verification of a file, collection of files, or the whole file system stored in the cloud. This phase starts when the user sends a requirement to verify integrity of files to TCC. TCC then retrieves files that need to be checked from the cloud and generates MAC'_{F_x} to send to the client. The client only compares the received MAC'_{F_x} and MAC_{F_x} that are stored on its device to verify the integrity of such files. This approach not only verifies the integrity of data but also saves energy for the device and bandwidth for the communication network. The reason is that checking and verification are processed on TCC and the client just runs a simple code for comparison. The result shows that this solution can save 90% processing requirements, thus saving significant energy for mobile device.

- *Authentication.* Chow [93] presents an authentication method using CC to secure the data access suitable for mobile environments. This scheme combines TrustCube [94] and implicit authentication [95,96] to authenticate the mobile clients. TrustCube is a policy-based cloud authentication platform using the open standards, and it supports the integration of various authentication methods. The authors build an implicit authentication system using mobile data (e.g., calling logs, SMS messages, website accesses, and location) for existing mobile environment. The system requires input constraints that make it difficult for mobile users to use complex passwords. As a result, this often leads to the use of simple and short passwords or personal identification numbers (PINs). Figure 6 shows the system architecture and how the system secures mobile users' access. When a web server receives a request from a mobile client, the web server redirects the request to the integrated authenticated (IA) service along with the details of the request. The IA service retrieves the policy for the access request, extracts the information that needs to be collected, and sends an inquiry to the IA server through the trusted network connect protocol. The IA server receives the inquiry, generates a report, and sends it back to the IA service. After that, the IA service applies the authentication rule in the policy and determines the authentication result (whether or not the mobile client is authenticated successfully for the access request) and sends the authentication result back to the web server. Based on the authentication result, the web server either provides the service or denies the request.
- *Digital rights management.* The unstructured digital contents (e.g., video, image, audio, and e-book) have often been pirated and illegally distributed. Protecting these contents from illegal access is of crucial importance to the content providers in MCC like traditional CC and peer-to-peer networks. Zou *et al.*[30] proposes Phosphor, a cloud-based mobile digital rights management (DRM) scheme with a subscriber identity module (SIM) card in mobile phone to improve the flexibility

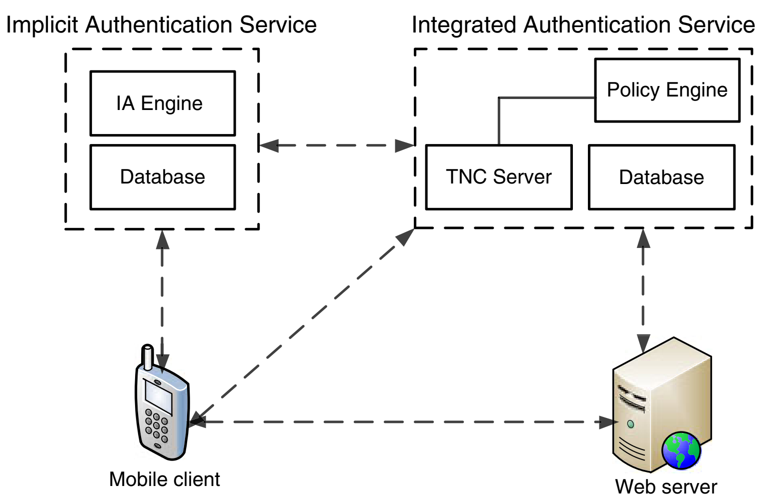

Figure 6. TrustCube architecture.

and reduce the vulnerability of its security at a low cost. The authors design a license state word (LSW) located in a SIM card and the LSW protocol based on the application protocol data unit (APDU) command. In addition, the cloud-based DRM with an efficient unstructured data management service can meet the performance requirements with high elasticity. Thus, when a mobile user receives the encrypted data (e.g., video stream) from the content server via real-time transport protocol, he/she uses the decryption key from a SIM card via APDU command to decode. If the decoding is successful, the mobile user can watch this video on his/her phone. The drawback of this solution is that it is still based on the SIM card of the mobile phone; so, it cannot be applied for other kinds of access; that is, a laptop using WiFi to access these contents.

(3) *Enhancing the efficiency of data access.* With an increasing number of cloud services, the demand of accessing data resources (e.g., image, files, and documents) on the cloud increases. As a result, a method to deal with (i.e., store, manage, and access) data resources on clouds becomes a significant challenge. However, handling the data resources on clouds is not an easy problem because of the low bandwidth, mobility, and the limitation of resource capacity of mobile devices.

For commercial cloud storage providers (e.g., Amazon S3), every I/O operations (e.g., put, copy, cut, and list) are taken by the cloud provider. The I/O operations are executed at a file-level in general, so this increases the cost of network communication and service for mobile users. Nam *et al.* [97] proposes an algorithm in which I/O operations are executed at a block-level. The algorithm uses log-structured I/O transaction [98] to minimize the number of the block-level I/O operations. The main idea here is to allow the cloud storage log-structure perform write operation with the optimal number of data blocks that adaptively changes with I/O and cloud storage pricing policy. The authors demonstrate that, through experimentation, the proposed solution reduces the total I/O costs considerably up to 54% compared with the data management at a file level in Amazon Simple Storage Service. However, this solution does not consider about access methods to adapt for this new data management.

Shen *et al.*[99] presents a cloud-based framework, named E-Recall to address the data access issue. This approach builds a novel infrastructure in managing, searching, sharing, and archiving the rich media resources based on the coordination of mobile search, CC, and multimodality integration. As shown in Figure 7, there are three main functional blocks as follow: query formulation, cloud-based indexing structure, and user-centric media sharing and publishing. Query formulation block is designed based on the principle of query dependent fusion [100] to optimize the representation for describing user information and search detail. Meanwhile, the aim of a cloud-based indexing structure block is to provide a database access method and that of user-centric media is to help mobile clients share and publish media resources in a

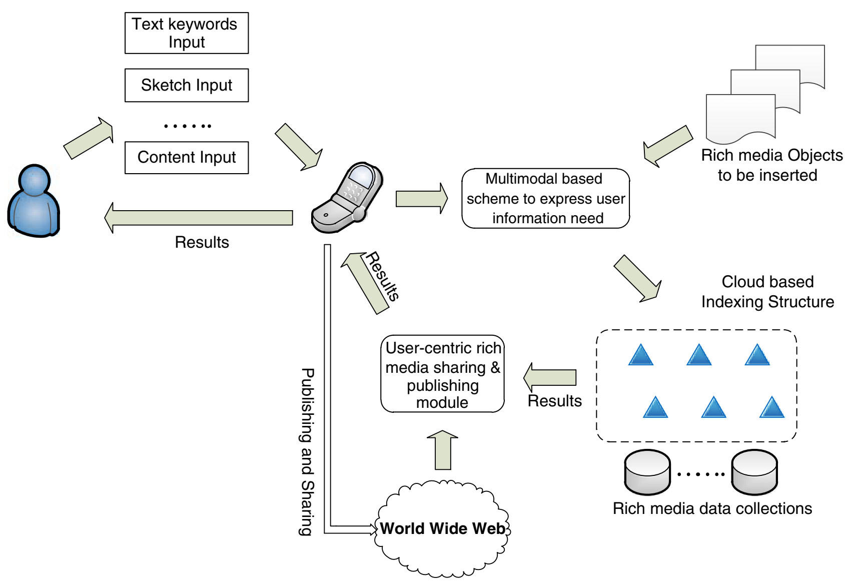

Figure 7. Architecture of E-Recall system.

flexible and fast way. This approach addresses both managing and accessing media resource issues in the cloud.

Another solution to increase the efficiency of accessing data on the cloud is using a local storage cache. Koukoumidis *et al.* [101] presents a solution which utilizes a memory capacity of mobile devices to increase the speed of data access, reduce latency, and improve energy efficiency for the mobile devices. The idea of this solution is to build a *Pocket Cloudlet* based on nonvolatile memory to store the specific parts or even full cloud services in the mobile devices. Using the *Pocket Cloudlet* clearly brings many benefits not only for users but also for service providers because this solution can increase access speed and reduce bottleneck of wireless link in a cellular network. However, not all data can be stored on the mobile cache. The authors develop an architecture for the *Pocket Cloudlet* including data selection and data management to determine the amount of data to be stored on the device for each cloud service. Also, this architecture introduces a mechanism to manage and synchronize data between mobile devices and the cloud to deal with changes of data on the cloud (e.g., web contents to be updated over time). This is an effective architecture for the mobile users accessing data on the cloud. Nonetheless, there are two issues in this approach. First, this architecture requires an expensive nonvolatile memory to store data from clouds. Second, for each cloud service, the mobile devices need to determine which parts of services to be cached locally. Thus, this is not flexible when applying for diverse services on the cloud.

Dong *et al.* [102] addresses three main issues as follows: maintaining seamless connection between users and clouds, controlling cache consistency, and supporting data privacy. As depicted in Figure 8, there are two main functional blocks; that is, random file system (RFS) client developed on the mobile device and RFS server located on the cloud. On the client side, RFS that is built above a local file system (FS) layer (e.g., Ext3 or FAT), consists of four components: encrypt, sync, metadata, and local cache heaps. Sync component will be connected to the comm component on the RFS server (as shown in Figure 8) via HTTP protocol to synchronize the data between a client and the cloud. In this approach, the authors propose using RESTful web service [6] as a service provider and HTTP as a communication protocol, because they are widely supported. Besides, to protect data privacy, encrypt component is used to control the data encryption and decryption. Sync and encrypt components are used by the users depending on their demands. In the kernel, metadata and local cache heaps aim to manage and track all files cached on the client. They also provide cache access for the synchronization. On the cloud side, user image service provider is responsible for managing the user accounts and the RFS image (i.e, the mobile file systems) for each RFS user. The cloud cache component caches data access for all RFS users to improve the performance of file access to the cloud. When the cloud receives a request from a client, it records the file block access patterns. Hence, the cloud can predict a new user access pattern and apply a server prepush optimization to increase the speed of file delivery. Finally, the cloud adapter is used to enable all RFS services to be performed on the diverse cloud storage systems, because cloud storage systems have their own application programing interface. This approach can be considered as a suitable solution for accessing data on the cloud from mobile users because it addresses several issues; that is, device-aware cache management, data privacy, and wireless connectivity. In the future, RFS can be improved by posing policies to manage users (i.e., what and when to encrypt). Alternatively, some other approaches (e.g., Moxie [103]) can be applied to hash the contents of files, thereby optimizing the use of bandwidth.

(4) *Context-aware mobile cloud services.* It is important for the service provider to fulfill mobile users' satisfaction by monitoring their preferences and providing appropriate services to each of the users. A lot of

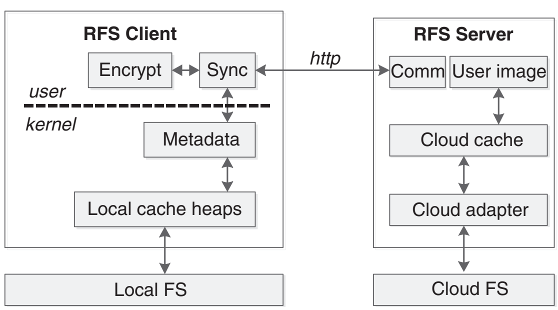

Figure 8. Random file system architecture.

research work try to utilize the local contexts (e.g., data types, network status, device environments, and user preferences) to improve the quality of service (QoS).

Samimi *et al.*[104] builds a model, called Mobile Service Clouds (MSCs), which is extended from service clouds paradigm [105]. In this model, when a customer uses a service on the cloud, the user's request firstly goes to a service gateway. The gateway will choose an appropriate primary proxy to meet the requirements (e.g., the shortest path and minimum round-trip time) and then sends the result to the user. In the case of disconnection, MSCs will establish transient proxies [106] for mobile devices to monitor the service path, and support dynamic reconfiguration (with minimum interruption). The advantages of this model are that the model addresses the disconnection issue and can maintain the QoS at an acceptable level.

La and Kim [107] propose a framework for providing context-aware mobile services based on the algorithm to choose a context-aware adapter. The authors consider several contexts such as device environments, user preferences, and situational contexts. The algorithm, firstly, determines a gap occurring in the given contexts. A gap is defined as a result of context changes. Then, the algorithm determines a cause of predefined gaps before saving the current states of the service invocation for recovering in the case of disconnection. After that, for each case of the identified gap, this algorithm will choose an appropriate adapter for the mobile user. Because the relationship between a cause and an adapter is predefined, the proper action can be chosen and performed. In the case of user preference context, the relationship can be checked when the context of mobile user changes. However, in the other contexts, the relationship cannot be known when mobile users change to another context. Moreover, the causes, adapters, and gaps in this model are predefined, so this may lack flexibility in practical usage.

Unlike [107], [108] builds a middleware module, called VOLARE, embedded on mobile device, which monitors the resources and contexts of the mobile device, thereby dynamically adjusting requirements of users at a runtime. As shown in Figure 9, when a mobile user launches an application on his/her mobile device that requires services on the cloud, this request is generated at mobile OS before it is sent to the service request module. At the same time, mobile OS simultaneously sends context data to context monitoring module, and QoS monitoring data to QoS monitoring module. An adaptation module will receive a service request request from a service request module and process this request along with the alerts received from a context monitoring module if there are significantly differences of the contexts and notifications about QoS from QoS monitoring module at

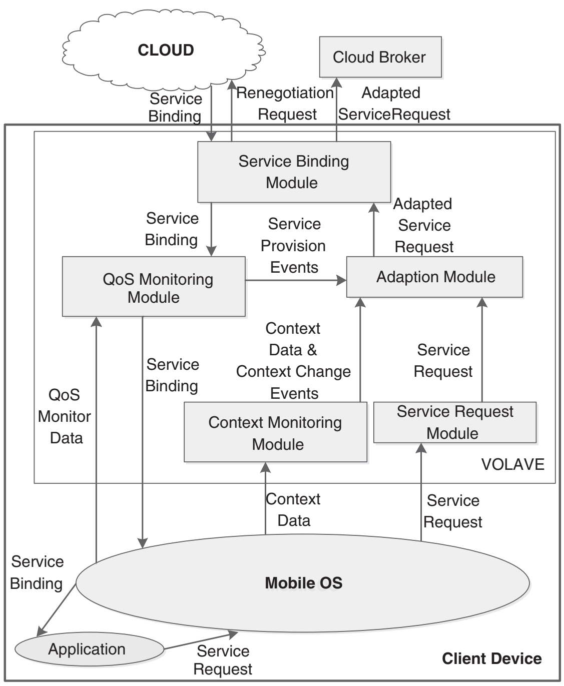

Figure 9. The VOLARE middleware modules.

runtime. Therefore, adaptation module can offer the appropriate service requests based on the context and resource data. Because the QoS monitoring module periodically performs checking, if the QoS levels of a service provided by service providers are lower than an accepted level of the request, a service request module will be notified to launch a new request for discovery of a new service satisfying the new requirements. The advantage of this model is that the model can automatically recognize changes in the contexts on a mobile device through the dependent functionality modules, thereby providing an effective service request for mobile users at runtime.

5. OPEN ISSUES AND FUTURE RESEARCH DIRECTIONS

Several research works contribute to the development of MCC by tackling issues as presented in the previous section. However, there are still some issues which need to be addressed. This section discusses several open issues and possible research directions in the development of MCC.

5.1. Low bandwidth

Although many researchers propose the optimal and efficient way of bandwidth allocation, the bandwidth limitation is still a big concern because the number of mobile and cloud users is dramatically increasing. We consider that fourth generation (4G) network and Femtocell are emerging as promising technologies that overcome the limitation and bring a revolution in improving bandwidth.

(1) *4G network.* Fourth generation network is a technology that significantly increases bandwidth capacity for subscribers. 4G network is capable of providing up to 100 Mbit/s (for 'LTE Advanced' standard) and 128 Mbit/s (for 'WirelessMAN-Advanced' standard) for mobile users, whereas the current 3G network supports a maximum of 14.4 Mbit/s. Furthermore, 4G network also promises other advantages such as widering mobile coverage area, smoothering quicker handoff, varied services, and so on [109]. Nevertheless, 4G wireless networks still have several issues related to network architecture, access protocol, or QoS that are taken into account in [110].

(2) *Femtocell.* Femtocell [111] is a small cellular base station, designed for use in a small area. Hay Systems Ltd (HSL) [112] develops a service to combine femtocells and CC to deliver a highly economical, scalable, and secure network for mobile operators. This allows the resources employed in delivering mobile services over the femtocell network to expand or contract as user demands for services increase or decrease, respectively. The result is a highly economical femtocell network with only sufficient resources being used at any given point, without impacting the ability to immediately scale to meet demands. In this paradigm, femtocells located in homes and offices of users connect via the Internet to the cloud to gain access to their operator's network. Mobile operators connect with the cloud enabling their subscribers to gain access to their network when using a femtocell connected to the cloud. However, [112] just shows that femtocell is practically useful when used with clouds. We need to investigate a uniform standard and performance impact of using femtocells in MCC.

5.2. Network access management

An efficient network access management not only improves link performance for mobile users but also optimizes bandwidth usage. Cognitive radio can be expected as a solution to achieve the wireless access management in mobile communication environment [113]. Cognitive radio increases the efficiency of the spectrum utilization significantly, by allowing unlicensed users to access the spectrum allocated to the licensed users. When this technique is integrated into MCC, the spectrum can be utilized more efficiently. The spectrum scarcity can be solved and thus millions of dollars for network providers can be saved [114]. However, cognitive radio is defined as wireless communication technology in which each node communicates via an optimal wireless system based on recognition of radio resource availability in heterogeneous wireless communication environment. Therefore, mobile users in MCC must be able to detect this radio resource availability (through spectrum sensing) while ensuring that the traditional services will not be interfered.

5.3. Quality of service

In MCC, mobile users need to access to servers located in a cloud when requesting services and resources in the cloud. However, the mobile users may face some problems such as congestion due to the limitation of wireless bandwidths, network disconnection, and the signal attenuation caused by mobile users' mobility. They cause delays when the users want to communicate with the cloud, so QoS is reduced significantly. Two new research directions are CloneCloud and Cloudlets that are expected to reduce the network delay.

(1) *CloneCloud.* CloneCloud brings the power of CC to your smartphones [115]. CloneCloud uses nearby computers or data centers to increase the speed of running smartphone applications. The idea is to clone the entire set of data and applications from the smartphone onto the cloud and to selectively execute some operations on the clones, re-integrating

Wirel. Commun. Mob. Comput. 2013; **13**:1587–1611 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

the results back into the smartphone. One can have multiple clones for the same smartphone, and clones pretend to be more powerful smartphones, and so on. CloneCloud is limited in some respects by its inability to migrate native state and to export unique native resources remotely. A related limitation is that CloneCloud does not virtualize access to native resources that are not virtualized already and are not available on the clone.

(2) *Cloudlets.* A cloudlet is a trusted, resource-rich computer or cluster of computers which is well-connected to the Internet and available for use by nearby mobile devices. Thus, when mobile devices do not want to offload to the cloud (maybe due to delay and cost), they can find and use a nearby cloudlet. In this way, mobile users may meet the demand for real-time interactive response by low-latency, one-hop, high-bandwidth wireless access to the cloudlet. If no cloudlet is available nearby, the mobile device may refer to the default mode that will send requirements to a distant cloud, or in the worst case, solely its own resources. Satyanarayanan *et al.*[116] builds an architecture through exploiting a virtual machine technology to rapidly instantiate customized service software on a nearby cloudlet and then uses that service over a wireless local area network. This technology can help mobile users overcome the limitations of CC due to wide area network latency and low bandwidth. However, there are some considerations that need to be addressed before this idea can be applied widely in practical system. For example, how to distribute processing, storage, and networking capacity for each cloudlet? How to manage policies for cloudlet providers to maximize user experience while minimizing cost? Also, trust and security for cloudlet are other issues in implementing this idea because adversaries can create a fake cloudlet to steal the user's information.

5.4. Pricing

Using services in MCC involves both mobile service provider (MSP) and cloud service provider (CSP). However, MSPs and CSPs have different services management, customers management, methods of payment, and prices. Therefore, this will lead to many issues; that is, how to set price, how the price will be divided among different entities, and how the customers pay. For example, when a mobile user runs mobile gaming application on the cloud, this involves the game service provider (providing a game license), mobile service provider (accessing the data through base station), and CSP (running game engine on a data center). The price paid by the game player has to be divided among these three entities such that all of them are satisfied with the division. It is clear that the business model including pricing and revenue sharing has to be carefully developed for MCC.

5.5. Standard interface

Interoperability becomes an important issue when mobile users need to interact and communicate with the cloud. The current interface between mobile users and cloud are mostly based on the web interfaces. However, using web interfaces may not be the best option. First, web interface is not specifically designed for mobile devices. Therefore, web interface may have more overhead. Also, compatibility among devices for web interface could be an issue. In this case, the standard protocol, signaling, and interface for interacting between mobile users and cloud would be required to ensure seamless services. In the future, HTML5 is expected as a promising technique to address this issue. HTML5 WebSockets offer an interface. However, an extensive performance evaluation and feasibility study have to be performed to ensure that it will work in MCC efficiently.

5.6. Service convergence

The development and competition of CSPs can lead to the fact that in the near future, these services will be differentiated according to the types, cost, availability and quality. Moreover, in some cases, a single cloud is not enough to meet the mobile user's demands. Therefore, the new scheme is needed in which the mobile users can utilize multiple clouds in a unified fashion. In this case, the scheme should be able to automatically discover and compose services for user. One of the potential solutions of this issue is the *sky computing*, which will be the next step of *cloud computing*. Sky computing is a computing model where resources from multiple cloud providers are leveraged to create a large scale distributed infrastructure [117]. Similarly, the *mobile sky computing* will enable the providers to support a cross-cloud communication and enable users to implement mobile services and applications. However, to offer a service to mobile user in a unified way, the service integration (i.e., convergence) would need to be explored.

6. CONCLUSION

Mobile cloud computing is one of the mobile technology trends in the future because it combines the advantages of both MC and CC, thereby providing optimal services for mobile users. That traction will push the revenue of MCC to $5.2 billion. With this importance, this article has provided an overview of MCC in which its definitions, architecture, and advantages have been presented. The applications supported by MCC including m-commerce, m-learning, and mobile healthcare have been discussed which clearly show the applicability of the MCC to a wide range of mobile services. Then, the issues and related approaches for MCC (i.e., from communication and computing sides) have been discussed. Finally, the future research directions have been outlined.

REFERENCES

1. Satyanarayanan M. Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MCS), 2010.
2. Satyanarayanan M. Fundamental challenges in mobile computing, In *Proceedings of the 5th annual ACM symposium on Principles of distributed computing*, 1996; 1–7.
3. Ali M. Green cloud on the horizon, In *Proceedings of the 1st International Conference on Cloud Computing (CloudCom)*, Manila, 2009; 451–459.
4. http://www.mobilecloudcomputingforum.com/.
5. White Paper. *Mobile Cloud Computing Solution Brief*. AEPONA, 2010.
6. Christensen JH. Using RESTful web-services and cloud computing to create next generation mobile applications, In *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications (OOPSLA)*, 2009; 627–634.
7. Liu L, Moulic R, Shea D. Cloud service portal for mobile device management, In *Proceedings of IEEE 7th International Conference on e-Business Engineering (ICEBE)*, 2011; 474.
8. Foster I, Zhao Y, Raicu I, Lu S. Cloud computing and grid computing 360-degree compared, In *Proceedings of Workshop on Grid Computing Environments (GCE)*, 2009; 1.
9. Calheiros RN, Vecchiola C, Karunamoorthy D, Buyya R. The Aneka platform and QoS-driven resource provisioning for elastic applications on hybrid Clouds. *Future Generation Computer Systems,* to appear.
10. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Journal on Future Generation Computer Systems* 2009; **25**(6): 599–616.
11. Huang Y, Su H, Sun W, *et al.* Framework for building a low-cost, scalable, and secured platform for web-delivered business services. *IBM Journal of Research and Development* 2010; **54**(6): 535–548.
12. Tsai W, Sun X, Balasooriya J. Service-oriented cloud computing architecture, In *Proceedings of the 7th International Conference on Information Technology: New Generations (ITNG)*, 2010; 684–689.
13. Forman GH, Zahorjan J. The Challenges of mobile computing. *IEEE Computer Society Magazine* 1994; **27**(4): 38–47.
14. Kakerow R. Low power design methodologies for mobile communication, In *Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors*, 2003; 8.
15. Paulson LD. Low-power chips for high-powered handhelds. *IEEE Computer Society Magazine* 2003; **36**(1): 21.
16. Davis JW. Power benchmark strategy for systems employing power management, In *Proceedings of the IEEE International Symposium on Electronics and the Environment*, 2002; 117.
17. Mayo RN, Ranganathan P. Energy consumption in mobile devices: why future systems need requirements aware energy scale-down, In *Proceedings of the Workshop on Power-Aware Computing Systems*, 2003.
18. Rudenko A, Reiher P, Popek GJ, Kuenning GH. Saving portable computer battery power through remote process execution. *Journal of ACM SIGMOBILE on Mobile Computing and Communications Review* 1998; **2**(1).
19. Smailagic A, Ettus M. System design and power optimization for mobile computers, In *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, 2002; 10.
20. Kremer U, Hicks J, Rehg J. A compilation framework for power and energy management on mobile computers, In *Proceedings of the 14th International Conference on Languages and Compliers for Parallel Computing*, 2001; 115–131.
21. Cuervo E, Balasubramanian A, Dae-ki C, *et al.* MAUI: making smartphones last longer with code offload, In *Proceedings of the 8th International Conference on Mobile systems, applications, and services*, 2010; 49–62.
22. http://aws.amazon.com/s3/.
23. Vartiainen E, Mattila KV-V. User experience of mobile photo sharing in the cloud, In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, 2010.
24. http://www.flickr.com/.
25. http://www.shozu.com/portal/index.do.
26. http://www.facebook.com/.
27. Garcia A, Kalva H. Cloud transcoding for mobile video content delivery, In *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, 2011; 379.
28. Kumar K, Lu Y. Cloud computing for mobile users: can offloading computation save energy. *IEEE Computer Society* 2010; **43**(4).

29. Li L, Li X, Youxia S, Wen L. Research on mobile multimedia broadcasting service integration based on cloud computing, In *Proceedings of the IEEE International Conference on Multimedia Technology (ICMT)*, 2010; 1.
30. Zou P, Wang C, Liu Z, Bao D. Phosphor: a cloud based drm scheme with SIM card, In *Proceedings of the 12th International Asia-Pacific on Web Conference (APWEB)*, 2010; 459.
31. Oberheide J, Veeraraghavan K, Cooke E, Flinn J, Jahanian F. Virtualized in-cloud security services for mobile devices, In *Proceedings of the 1st Workshop on Virtualization in Mobile Computing (MobiVirt)*, 2008; 31–35.
32. Yang X, Pan T, Shen J. On 3G mobile e-commerce platform based on cloud computing, 2010; 198–201.
33. Dai J, Zhou Q. A PKI-based mechanism for secure and efficient access to outsourced data, In *Proceedings of the 2nd International Conference on Networking and Digital Society (ICNDS)*, 2010; 640.
34. Leina Z, Tiejun P, Guoqing Y. Research of mobile security solution for fourth party logistics, In *Proceedings of the 6th International Conference on Semantics Knowledge and Grid (SKG)*, 2011; 383–386.
35. Chen X, Liu* J, Han J, Xu H. Primary exploration of mobile learning mode under a cloud computing environment, In *Proceedings of the International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT), vol.2*, 2010; 484–487.
36. Gao H, Zhai Y. System design of cloud computing based on mobile learning, In *Proceedings of the 3rd International Symposium on Knowledge Acquisition and Modeling (KAM)*, 2010; 293–242.
37. Li J. Study on the development of mobile learning promoted by cloud computing, In *Proceedings of the 2nd International Conference on Information Engineering and Computer Science (ICIECS),* 2010; 1.
38. Zhao W, Sun Y, Dai L. Improving computer basis teaching through mobile communication and cloud computing technology, In *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, 2010; 452–454.
39. Yin C, David B, Chalon R. Use your mobile computing devices to learn - Contextual mobile learning system design and case studies, In *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT),* 2009; 440.
40. Rieger R, Gay G. Using mobile computing to enhance field study, In *Proceedings of the 2nd international conference on Computer support for collaborative learning (CSCL)*, 1997; 218–226.
41. Ferzli R, Khalife I. Mobile cloud computing educational tool for image/video processing algorithms, In *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE)*, 2011; 529.
42. Kohn LT, Corrigan JM, Donaldson S. *To Err Is Human: Building a Safer Health System*. National Academy Press: Washington, 1999.
43. Kopec D, Kabir MH, Reinharth D, Rothschild O, Castiglione JA. Human errors in medical practice: systematic classification and reduction with automated information systems. *Journal of Medical Systems* 2003; **27**(4): 297–313.
44. Varshney U. Pervasive healthcare and wireless health monitoring. *Journal on Mobile Networks and Applications* 2007; **12**(2-3): 113–127.
45. Doukas C, Pliakas T, Maglogiannis I. Mobile healthcare information management unitizing cloud computing and Android OS, In *Annual International Conference of the IEEE on Engineering in Medicine and Biology Society (EMBC)*, 2010; 1037–1040.
46. Tang W-T, Hu C-M, Hsu C-Y. A mobile phone based homecare management system on the cloud, In *Proceedings of the 3rd International Conference on Biomedical and Informatics (BMEI)*, 2010; 2442.
47. Hoang DB, Chen L. Mobile cloud for assistive healthcare (MoCAsH), In *Proceedings of the 2010 IEEE Asia-Pacific Services Computing Conference (APSCC)*, 2011; 325.
48. Nkosi MT, Mekuria F. Cloud computing for enhanced mobile health applications, In *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science*, 2011; 629.
49. Li Z, Wang C, Xu R. Computation offloading to save energy on handheld devices: a partition scheme, In *Proceedings of the 2001 international conference on Compilers, architecture, and synthesis for embedded systems (CASES)*, 2001; 238–246.
50. Wang S, Dey S. Rendering adaptation to address communication and computation constraints in cloud mobile gaming, In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2011; 1–6.
51. Li H, Hua X-S. Melog: mobile experience sharing through automatic multimedia blogging, In *Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing (MCMC)*, 2010; 19–24.
52. Ye Z, Chen X, Li Z. Video based mobile location search with large set of SIFT points in cloud, In *Proceedings of the 2010 ACM multimedia workshop*

on Mobile cloud media computing (MCMC), 2010; 25–30.
53. http://www.onehourtranslation.com/.
54. Pendyala VS, Holliday J. Performing intelligent mobile searches in the cloud using semantic technologies, In *IEEE International Conference on Granular Computing*, 2010; 381.
55. Lagerspetz E, Tarkoma S. Cloud-assisted mobile desktop search, In *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM)*, 2010; 826.
56. Lagerspetz E, Lindholm T, Tarkoma S. Dessy: towards flexible mobile desktop search, In *Proceedings of the DIALM-POMC International Workshop on Foundations of Mobile Computing*, 2007.
57. Fabbrizio GD, Okken T, Wilpon JG. A speech mashup framework for multimodal mobile services, In *Proceedings of the 2009 international conference on Multimodal interfaces (ICMI-MLMI)*, 2009; 71–78.
58. Cai-Dong G, Kan L, Jian-Ping W, *et al.* The investigation of cloud-computing-based image mining mechanism in mobile communication WEB on Android, In *Proceedings of the 9th International Conference on Grid and Cooperative Computing (GCC)*, 2011; 408.
59. Angin P, Bhargava B, Helal S. A mobile-cloud collaborative traffic lights detector for blind navigation, In *Proceedings of the 2010 Eleventh International Conference on Mobile Data Management (MDM)*, 2010; 396–401.
60. Li Y-C, Liao I-J, Cheng H-P, Lee W-T, A cloud computing framework of free view point real-time monitor system working on mobile devices, In *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS),* 2011; 1.
61. Zhenyu W, Chunhong Z, Yang J, Hao W. Towards cloud and terminal collaborative mobile social network service, In *Proceedings of the 2nd IEEE International Conference on Social Computing (SocialCom)*, 2010; 623.
62. Kelenyi I, Nurminen JK. CloudTorrent - energy-efficient bittorrent content sharing for mobile devices via cloud services, In *Proceedings of the 7th IEEE on Consumer Communications and Networking Conference (CCNC)*, 2010; 1.
63. Jin X, Kwok YK. Cloud assisted P2P media streaming for bandwidth constrained mobile subscribers, In *Proceedings of the 16th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, 2011; 800.
64. Jung E, Wang Y, Prilepov I, Maker F, Liu X, Akella V. User-profile-driven collaborative bandwidth sharing on mobile phones, In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MCS)*, 2010.
65. Huerta-Canepa G, Lee D. A virtual cloud computing provider for mobile devices, In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MCS),* 2010.
66. Zhang L, Ding X, Wan Z, Gu M, Li XY. WiFace: a secure geosocial networking system using WiFi-based multi-hop MANET, In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MSC)*, 2010.
67. Klein A, Mannweiler C, Schneider J, Hans D. Access schemes for mobile cloud computing, In *Proceedings of the 11th International Conference on Mobile Data Management (MDM)*, 2010; 387.
68. Klein A, Mannweiler C, Schotten HD. A framework for intelligent radio network access based on context models, In *Proceedings of the 22nd WWRF meeting 2009*, 2009.
69. Jigang W, Thambipillai S. A branch-and-bound algorithm for hardware/software partitioning, In *Proceedings of the 4th IEEE International Symposium on Signal Processing and Information Technology*, 2005; 526.
70. Chen G, Kang BT, Kandermir M, Vijaykrishnan N, Irwin MJ, Chandranouli R. Studying energy trade offs in offloading computation/compilation in Java-enabled mobile devices. *IEEE Transactions on Parallel and Distributed Systems* 2004; **15**(9): 795.
71. Wang C, Li Z. A computation offloading scheme on handheld devices. *Journal of Parallel and Distributed Computing Special issue on middleware* 2004; **64**(6): 740–746.
72. Andersien L. Program analysis and specialization for the C programming language, *Ph.D thesis*, DIKU, University of Copenhagen, 1994.
73. Wilson RP, Lam MS. Efficient context-sensitive pointer analysis for C programs, In *Proceedings of the ACM SIGPLAN '95 Conference on Programming Language Design and Implementation (PLDI)*, 1995; 1–12.
74. Kimmo E, Raatikainen E. Cluster analysis and workload classification, In *ACM SIGMETRICS Performance Evaluation Review*, 1993.
75. Dai YS, Xie M, Wang X. A heuristic algorithm for reliability modeling and analysis of grid systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 2007; **37**(2): 189.

76. Hunt GC, Scott ML. The Coign automatic distributed partitioning system, In *Proceedings of the 3rd symposium on Operating systems design and implementation (OSDI)*, 1999; 187–200.
77. Stoer M, Wagner F. A simple min-cut algorithm. *Journal of the ACM (JACM)* 1997; **4**: 44.
78. Xian C, Lu YH, Li Z. Adaptive computation offloading for energy conservation on battery-powered systems, In *International Conference on Parallel and Distributed Systems*, 2009; 1.
79. Ou S, Yang K, Liotta A, Hu L. Performance analysis of offloading systems in mobile wireless environments, In *Proceedings of the IEEE International Conference on Communications (ICC)*, 2007; 1821.
80. Tang MH, Cao J. A dynamic mechanism for handling mobile computing environmental changes, In *Proceedings of the 1st international conference on Scalable information systems (InfoScale)*, 2006; 1–9.
81. Chun B-G, Maniatis P. Dynamically partitioning applications between weak devices and clouds, In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MCS)*, 2010.
82. Oberheide J, Cooke E, Jahanian F. Rethinking antivirus: executable analysis in the network cloud, In *Proceedings of the 2nd USENIX workshop on Hot topics in security (HOTSEC)*, 2007.
83. Oberheide J, Cooke E, Jahanian F. Cloudav: N-version antivirus in the network cloud, In *Processing of the 17th USENIX Security Symposium*, 2008.
84. Ogness J. Dazuko: an open solution to facilitate on-access scanning, In *Proceedings of the 13th Virus Bulletin International Conference*, 2003.
85. Portokalidis G, Homburg P, Anagnostakis K, Bos H. Paranoid Android: versatile protection for smartphones, In *Proceedings of the 26th Annual Computer Security Application Conference (ACSAC)*, 2010; 347–356.
86. Zhangwei H, Mingjun X. A distributed spatial cloaking protocol for location privacy, In *Proceedings of the 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, 2010; 468.
87. Sweeney L. K-anonymity: a model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems* 2002; **10**(5): 557–570.
88. Wang S, Wang XS. In-device spatial cloaking for mobile user privacy assisted by the cloud, In *Proceedings of the 11th International Conference on Mobile Data Management (MDM)*, 2010; 381.
89. Chow C-Y, Mokbel MF, Aref WG. Casper*: query processing for location services without compromising privacy. *ACM Transactions on Database Systems (TODS)* 2009; **34**(4): 1–48.
90. Tanenbaum A, Van Steen M. *Distributed Systems: Principles and Paradigms*. Pearson Prentice Hall, 2007.
91. Wang W, Li Z, Owens R, Bhargava B. Secure and efficient access to outsourced data, In *ACM Cloud Computing Security Workshop (CCSW)*, 2009; 55–66.
92. Itani W, Kayssi A, Chehab A. Energy-efficient incremental integrity for securing storage in mobile cloud computing, In *International Conference on Energy Aware Computing (ICEAC)*, January 2011; 1.
93. Chow R, Jakobsson M, Masuoka R, *et al.* Authentication in the clouds: a framework and its application to mobile users, In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop (CCSW)*, 2010; 1–6.
94. Song Z, Molina J, Lee S, Kotani S, Masuoka R. TrustCube: an infrastructure that builds trust in client, In *Proceedings of the 1st International Conference on Future of Trust in Computing*, 2009.
95. Jakobsson M, Shi E, Golle P, Chow R. Implicit authentication for mobile devices, In *Processing of the 4th USENIX Workshop on Hot Topics in Security (HotSec)*, 2009.
96. Shi E, Niu Y, Jakobsson M, Chow R. Implicit authentication through learning user behavior, In *Proceedings of the Implicit authentication Security Conference (ISC)*, 2010.
97. Nam YJ, Park YK, Lee JT, Ishengoma F. Cost-aware virtual USB drive: providing cost-effective block i/o management commercial cloud storage for mobile devices, In *Proceedings of the 13th IEEE International Conference on Computational Science and Engineering (CSE)*, 2011; 427.
98. Rosenblum M, Ousterhout JK. The design and implementation of a log-structured file system. *ACM Transactions on Computer Systems (TOCS)* 1992; **10**(1): 26–52.
99. Shen J, Yan S, Hua X-S. The e-recall environment for cloud based mobile rich media data management, In *Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing*, 2010; 31–34.
100. Kennedy L, Chang SF, Natsev A. Query-adaptive fusion for multimodal search. *Proceedings of the IEEE Magazine* 2008; **96**(4): 567–588.
101. Koukoumidis E, Lymberopoulos D, Strauss K, Liu J, Burger D. Pocket cloudlets, In *Proceedings of the 16th international conference on Architectural support for programming languages and operating systems (ASPLOS)*, 2011; 171–184.

102. Dong Y, Peng J, Wang D, *et al.* RFS: a network file system for mobile devices and the cloud. *ACM SIGOPS Operating Systems Review* 2011; **45**(1): 101–111.
103. Eaton RP. *Improving Access to Remote Storage for Weakly Connected Users*. University of California at Berkeley: Berkeley, 2007.
104. Samimi FA, Mckinley PK, Sadjadi SM. Mobile service clouds: a self-managing infrastructure for autonomic mobile computing services, In *Proceedings of the 2nd International Workshop on Self-Managed Networks,Systems & Services (SelfMan),vol. 3996*, 2006; 130–141.
105. McKinley PK, Samimi FA, Shapiro JK, Tang C. Service Clouds: A Distributed Infrastructure For Constructing Autonomic Communication Services, In *Proceedings of the 2nd IEEE International Symposium on Dependable,Autonomic and Secure Computing*, 2006; 341.
106. Samimi FA, McKinley PK, Sadjadi SM, Ge P. Kernel-middleware interaction to support adaptation in pervasive computing environments, In *Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing (MPAC)*, 2004; 140–145.
107. La HH, Kim SD. A Conceptual framework for provisioning context-aware mobile cloud services, In *Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD)*, 2010; 466.
108. Papakos P, Capra L, Rosenblum DS. VOLARE: context-aware adaptive cloud service discovery for mobile systems, In *Proceedings of the 9th International Workshop on Adaptive and Reflective Middleware (ARM)*, 2010; 32–38.
109. Rahman M, Mir FAM. Fourth Generation (4G) mobile networks - features,technologies & issues, In *Proceedings of the 6th IEEE International Conference on 3G and Beyond*, 2007; 1.
110. Varshney U, Jain R. Issues in emerging 4G wireless networks. *Computer* 2002; **34**(6): 94–96.
111. Boccuzzi J, Ruggiero M. *Femtocells: Design & Application*. McGraw-Hill, 2011.
112. http://www.haysystems.com/.
113. Yucek T, Arslan H. A survey of spectrum sensing algorithms for cognitive radio applications, In *IEEE Communications Surveys & Tutorials*, 2009; 116–130.
114. Ge F, Lin H, Khajeh A. Cognitive radio rides on the cloud, In *Military Communications Conference (MILCOM)*, 2011; 1448.
115. Chun BG, Ihm S, Maniatis P, Naik M, Patti A. CloneCloud: elastic execution between mobile device and cloud, In *Proceedings of the 6th conference on Computer systems (EuroSys)*, 2011; 301–314.
116. Satyanarayanan M, Bahl P, Caceres R, Davies N. The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing* 2009; **8**(4): 14–23.
117. Keahey K, Tsugawa M, Matsunaga A, Fortes J. Sky Computing. *IEEE Internet Computing Magazine* 2009; **13**(5): 43.

AUTHORS' BIOGRAPHIES

Dinh Thai Hoang received his Electronics and Telecommunications diploma in 2009 from HUT (Hanoi University of Technology), Vietnam. He joined the Network lab in HUT from February 2009 to June 2010 in project 'Development of the service platform for next generation networks (NGN/UMTS)'. He has been a project officer since November 2010 in NTU (Nanyang Technological University) – Singapore. His current research interests include design, analysis, and optimization of wireless communication and mobile cloud computing.

Chonho Lee currently works as a research fellow at the School of Computer Engineering, Nanyang Technological University, Singapore. He received his BS and MS in Computer Science from the University of California, Irvine and PhD in Computer Science from the University of Massachusetts, Boston, 2010. His current research interests include optimization and self-adaptation using game theory and bio-inspired approaches in large-scale network systems such as data centers and clouds.

Dusit Niyato is currently an assistant professor in the Division of Computer Communications, School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include design, analysis, and optimization of wireless communication and mobile cloud computing, smart grid systems, and green radio communications. Dusit serves as an editor for the IEEE Transactions on Wireless Communications, Wireless Communications and Mobile Computing (WCMC) Journal, and Journal of Communications and Networks (JCN).

Ping Wang received the BE and ME degrees from Huazhong University of Science and Technology, Wuhan, China, in 1994 and 1997, respectively, and the PhD degree from University of Waterloo, Ontario, Canada, in 2008, all in electrical engineering. Since June 2008, she has been an assistant professor in the School of Computer Engineering, Nanyang Technological University, Singapore. Her current research interests include QoS provisioning and resource allocation in wireless communications and cloud computing. She was a corecipient of the Best Paper Award from the IEEE International Conference on Communications (ICC 07). She is a member of the IEEE. She is an editor of IEEE Transactions on Wireless Communications, EURASIP Journal on Wireless Communications and Networking, International Journal of Communication Systems, and International Journal of Ultra Wideband Communications and Systems.