A Survey of Mobile Cloud Computing Applications: Perspectives and Challenges

Yating Wang and Ing-Ray Chen
Department of Computer Science
Virginia Tech
7054 Haycock Road
Falls Church, VA 22043, USA
{yatingw,irchen}@vt.edu

Ding-Chau Wang
Department of Information Management
Southern Taiwan University of Science and Technology
Tainan, Taiwan
dcwang@mail.stut.edu.tw

Abstract

As mobile computing has been developed for decades, a new model for mobile computing, namely, mobile cloud computing, emerges resulting from the marriage of powerful yet affordable mobile devices and cloud computing. In this paper we survey existing mobile cloud computing applications, as well as speculate future generation mobile cloud computing applications. We provide insights for the enabling technologies and challenges that lie ahead for us to move forward from mobile computing to mobile cloud computing for building the next generation mobile cloud applications. For each of the challenges, we provide a survey of existing solutions, identify research gaps, and suggest future research areas.

Key Words: Mobile computing, cloud computing, mobile cloud computing, mobile cloud applications.

1 Introduction

With the proliferation of smart mobile devices and cloud computing technologies, mobile cloud computing (MCC) [1, 2, 3] has emerged as a new computing paradigm for building the next generation MCC applications. MCC promises to bring new exciting MCC applications beyond mobile computing (MC) applications by combining cloud computing [4, 5], mobile computing [6, 7], and data analytics [8] at the fingertip of a human operator.

In this paper we survey existing and speculate future generation MCC applications. We limit our survey to infrastructure based MCC applications where the hardware infrastructure remains static and provides services to the mobile users [9]. We provide insights for the enabling technologies and challenges that lie ahead for us to move forward from MC to MCC for building the next generation MCC applications. For each of the challenges, we provide a survey of existing solutions, identify research gaps, and suggest future research areas.

The rest of the survey paper is organized as follows. In Section 2, we provide an overview of MCC extending MC. In Section 3, we survey existing MCC applications and speculate future generation MCC applications. In Section 4 we discuss several major challenges for building the next generation MCC applications, provide a survey of existing solutions, identify research gaps, and suggest future research directions for answering these challenges. Finally, in Section 5 we conclude the paper.

2 From MC to MCC

The NIST defines cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [5]. It has three layers of services, namely, *Software as a Service (SaaS)*, *Platform as a Service (PaaS)*, and *Infrastructure as a Service (IaaS)*. In SaaS, it provides software or applications for users to access the application that will be run in the cloud. So users simply use the applications without concerning system configuration problems. In the PaaS layer, users can choose preferable operating systems and develop personal softwares by using supported resources in the cloud. For instance, users can store their data in the cloud and send any queries to the cloud whenever they need to retrieve information. In the IaaS layer, users can setup personal operating systems, configure computation environment, and develop software. The cloud provides a powerful processing core and a massive storage space with configurable computing resources for users to do computation on it. By this architecture model, from top to down, users can have more control on the available computing resources. Cloud service is characterized as on-demand, elastic, quality of service guaranteed, and pay-per-use [10, 11, 13, 14].

With the advancement of wireless infrastructure, mobile devices can connect to the cloud any place and any time. Cloud computing is especially an enabler for the *bring your own device* (BYOD) technology permitting employees to bring personally owned mobile devices to their workplace, and use those devices to access privileged enterprise content and applications stored on the cloud. Cloud computing goes hand-in-hand with *mobile virtualization* which enables multiple operating systems or virtual machines (VMs) to run simultaneously on a mobile device. That is, cloud computing can provide separate services (including applications, user profiles, contacts and data) to completely isolated VM containers running on the same mobile device or smartphone with mobile virtualization. Furthermore, with cloud computing providing a variety of *task-oriented mobile services* with virtually unlimited computation power and storage on demand, a mobile device can flexibly run task-oriented applications on separate VMs to support its needs. Cloud computing thus advances mobile computing in three major areas: supporting ultra thin mobile devices with mobile virtualization, providing scalable mobile computation, and supporting big data mobile applications. Essentially, cloud computing propels a new class of applications which we call MCC applications extending traditional MC applications with unlimited storage and computation resources as well as task-oriented services. In the literature, there are two definitions for MCC based on the same line of observations discussed above extending MC to MCC. In the first definition [15, 16], MCC is defined as a computing model combining mobile computing and the cloud, where the cloud can handle large storage and processing for mobile devices remotely. In the second definition [17, 18], the cloud does not have to be a remote powerful server, but one that advances mobile devices cooperating for storage and processing.

3 MCC Applications

With scalable computation and large data storage, cloud computing facilitates MCC applications to be run on ultra thin mobile devices. In this section, we first discuss existing MCC applications. Then we discuss emerging and future MCC applications.

3.1 Existing MCC Applications

We have witnessed a number of MCC applications in recent years, including mobile commerce, multimedia sharing, mobile learning, mobile sensing, mobile healthcare, mobile gaming, mobile social networking, location-based mobile service, and augmented reality.

Mobile commerce, such as e-banking, e-advertising and e-shopping, uses scalable processing power and security measures to accommodate a high volume of traffic due to simultaneous user access and data transaction processing. multimedia sharing provides secure viewing and sharing of multimedia information stored on smartphones while providing administrative controls to manage user privileges and access rights necessary to ensure security. Mobile learning allows a thin terminal to access learning materials on the cloud any time and any place. Mobile sensing utilizing sensor-equipped smartphones to collect data will revolutionize many MCC applications including healthcare, social networking, and environment/health monitoring. Mobile healthcare allows an enormous amount of patient data to be stored on the cloud instantaneously. A doctor can conveniently look at the patient records on his/her mobile device for remote diagnosis or monitor a patient's status for preventive actions. Mobile gaming achieves scalability by leveraging scalable computation and instantaneous data update on the cloud side and screen refresh at the mobile device side. Mobile social networking allows a group of mobile users to upload audio/video/multimedia data for real-time sharing, with cloud computing providing not only storage for data, but also security to protect secrecy and integrity of data.

3.2 Emerging and Future MCC Applications

Future MCC applications must leverage unique characteristics of MCC. Due to limitation of power, intensive data processing on mobile devices is always costly. With the technology advancement, however, mobile devices are

Table 1: Emerging and Future MCC Applications.

Application Category	References
Crowdsourcing (crowd computing)	[19] [20] [21]
Collective sensing	[22] [23] [24] [25]
- Traffic/Environment monitoring	[26, 27, 28] [29]
- Mobile cloud social networking	[30] [31] [32]
- Mobile cloud healthcare	[33] [34] [35]
Location-based mobile cloud service	[36] [37]
Augmented reality and mobile gaming	[38] [39]

equipped with more functional units, such as high-resolution camera, barometer, light sensor, etc. Future MCC applications must leverage deep sensing capability of smartphones for data collection. Data can be uploaded to the cloud and the cloud can integrate pieces of observations from mobile devices and utilize data analytics techniques to mine and visualize trends or patterns embedded in massive data collected in parallel at runtime from millions of mobile devices. For instance, given a severe natural disaster, people nearby can send photos taken from the cameras in their smartphones to the cloud, and the cloud server can process these data, analyze possible crucial points, and plot a detailed map, covering not only visible objects but also invisible physical phenomena, such as the presence of poisonous air to help facilitating the rescue mission. With potentially unlimited storage and processing power, MCC brings out potential killer applications.

Table 1 lists emerging and future applications that can power from MCC. Initial efforts toward building these killer MCC applications are also cited in the table.

Crowdsourcing (crowd computing) is one of the emerging MCC applications [19]. It utilizes sensing functions of pervasive mobile devices and high processing capability of the cloud. Two future crowdsourcing applications discussed in [20] can potentially benefit from MCC. One is for finding a lost child; one is for disaster relief. In the first application, smartphones upload pictures taken within an hour to a website in response to an amber alert via texting, and a policeman searches for the lost child by doing data analytics on thousands of photos uploaded using an application in his smartphone.

In the second case, after a disaster the infrastructure around the disaster site is broken and there is no way to assess the damage using the existing infrastructure. By using cameras on smartphones, citizens take pictures of the disaster site and transmit them via wireless communication, helping a detailed map of the disaster site to be reconstructed, such that rescue work can be effectively and efficiently performed. Defining scalable architectures, creating efficient algorithms for crowdsourcing, stimulating crowd participation, and preserving user privacy are major issues. Yang et al. [21] devised two incentive mechanisms for user-centric and platform-centric computing. On the user side, users contribute data through a bidding process to maximize their profits. On the cloud side, a game theoretical approach based on auctioning is used to maximize the system utility.

Another emerging and future mobile cloud application is collective sensing [22]. Cheng et al. designed SenseOrchestra [23] for node location tracking via collective sensing. Lu et al. [24] designed SoundSense to run in Apple iPhones to recognize events by collectively sound sensing. Lastly, Sensorly [25] provides a map of free wireless coverage through collective sensing by its mobile cloud members. Emerging collective sensing applications include composing a realtime traffic map from collective traffic data sensing [26, 27, 28], monitoring environmental pollution [29], mobile cloud social networking [30] [31] [32], and mobile cloud based healthcare [33, 34, 35].

Location-based mobile service is also an emerging MCC application. Tamai et al. [36] designed a platform for location-based services leveraging scalable computation and large storage space to answer a large number of location-based queries efficiently. Location-based mobile service is often context-aware. In addition to taking account of location information, location-based mobile services also consider the environment and application context, such as people, other devices and time between changes. The environment information can be feasibly obtained. Social networking can connect several people around sharing common interests. For example, the application can recommend an online game to play or a chat session connecting people with common interests. La et al. [37] developed a framework for location-based mobile service with user mobile devices monitoring the context information to send to the cloud and with the cloud analyzing and adapting the context information to suggest location-based mobile services to users sharing common interests.

Lastly, augmented reality and mobile gaming is emerging as a MCC killer application. While traditionally augmented reality is made possible only with special equipment with huge processing power, it is now made possible with mobile cloud computing with scalable computation and big data storage. Kangas et al. [38] developed mobile code to be processed by the cloud to realize augmented reality. Luo [39] proposed an augmented reality application to enhance user experiences.

4 Challenges in Building MCC Applications

Ideally, MCC can be an enabler to realize the vision of any time any place computation and data analytics on big data stored on the cloud, returning results requested by a mobile user instantly. Future MCC applications thus will have major social impacts. In this section we discuss major challenges for building the next generation MCC applications, provide a survey of existing solutions in the literature, identify research gaps, and suggest future research directions for answering these challenges.

Table 2 summarizes challenges, existing solutions, and future research directions for building MCC applications.

4.1 Code/Computation Offloading

One design issue for building MCC applications is code/computation offloading [40] to enhance MCC application performance and conserve mobile device energy. The framework of elastic application processing comprises three parts: application partitioning, code offloading, and remote execution. Application partitioning is the process of identifying the units (threads, methods, or classes) that can be processed on the cloud. Static partitioning appeared in [41, 42, 43, 44] in which the entry and exit points of a remote method call can be statically identified. Application partitioning is crucial for code offloading. It requires the application be partitioned at the breakpoints correctly and efficiently. If a breakpoint is not set properly, it will largely degrade performance. It also requires a cost model be developed such that the overall cost related to the computation cost, network cost, and energy cost can be minimized.

Several projects [45, 46, 47, 48] advocated partitioning application threads between the cloud and a mobile device based on QoS considerations, or applying parallelism at the server side to shorten the delay if a complex application

Table 2: MCC Application Challenges, Existing Solutions, and Future Research Areas.

Challenges	Existing Solutions	Future Directions
Code/Computation Offloading	Static partitioning [41, 42, 43, 44] Dynamic profiling [45, 46, 47, 48] Local/Cloud processing decision [49, 50]	Automation of code/computation offloading
Task-Oriented Mobile Services	Mobile-Data-as-a-Service [51] – [52] Mobile-Computing-as-a-Service [53] Mobile-Multimedia-as-a-Service [54] [55] [56] [57] Location-Based Services [58] [59] [60] [61] [62]	Creating human-centric task-oriented mobile services
Elasticity and Scalability	Data intensive computation [63] Resource allocation [64] Scheduling [65] [66] VM migration [67]	Design and validation of resource allocation/scheduling algorithms using valid traffic models for MCC applications
Security	Authentication [68] [69] [70] [71] [72] Authorization [73] [74] [75] [76] [77] [78] [79] [80] Data/Code Integrity [81] [82]	Cloud-to-mobile authentication Authorization without releasing user credential Code integrity verification
Cloud Service Pricing	Auctioning/Bidding [83] [84] [85] [86] [87] Game Theory Based [88]	Empirical validation Pricing optimization

is uploaded to the cloud server for execution.

Chun et al. designed CloneCloud [45] that partitions an application offline and offloads the threads to the cloud which runs a clone of the mobile device. The partition is determined by a static program analyzer followed by dynamic program profiling. The static analyzer pinpoints the legal breakpoints in the application code. For example, a method will not be migrated if it recalls resources such as the accelerator and camera in a mobile device; co-related native stateful methods should be executed in different machines; and methods for excluding nested *suspending* and *resuming* should not be migrated. For each application, a cost model is developed and the cost of each method migration is recorded via dynamic profiling, based on which the optimization solver determines which threads should be executed on the cloud.

MAUI [46] provides an architecture for code offloading in MCC applications, addressing both energy and performance issues. Instead of cloning the mobile device on the cloud, it builds proxies, profilers and solvers on both the client and server sides. The measurement for remote processing is the number of states to be transferred, the CPU cycles, and networking environments. In the beginning, software developers have to manually identify methods or classes capable of being executed remotely. There are certain constraints on code to be offloaded. It should not be API code on the user side, code for interacting with I/O devices or internal resources on mobile devices. During processing, the system has to dynamically determine if offloading is beneficial based on the performance of the last method offloaded. If the user fails to connect with the server, the proxy will reinvoke the method locally or run it on other servers.

Designed for Android, Cuckoo [47] is a programming model for code offloading with consideration given to intermittent network connection. Building on top of MAUI and CloneCloud, ThinkAir [48] provides a framework for code offloading while adding parallel processing capability for running MCC applications. The parallelism is implemented by running the MCC application on the cloud among several Virtual Machines (VMs) based on the *Divide-and-Conquer* design.

Remote Processing Framework (RPF) was introduced in [49] where a client can automatically analyze if a problem should run locally or on a remote server on the cloud. Newhouse et al. [50] designed a remote processing system. The proxy server processes a client's request by first retrieving code from a code repository, and then sending it to a cloud server for executing the code retrieved for load balancing consideration.

The literature thus far does not have work considering automation of the code/computation offloading. This is often a human intervention required to do code analysis for selection of legal breakpoints and/or for parallel code execution. This is then followed by dynamic profiling to confirm if the breakpoints selected are indeed feasible. More effort is required for code/computation offloading, especially in automating application partitioning.

4.2 Task-Oriented Mobile Services

The second design issue is to provide mobile devices flexible task-oriented mobile services for offloading applications to the cloud. Mobile devices have severe limitations in computation, memory and display capability. The wireless network environment often is changeable with unreliable communication service. Mobile users also tend to use smartphones to do certain tasks but not all tasks. Hence, mobile devices require task-oriented mobile services.

While cloud computing offers SaaS, PaaS, and IaaS services, MCC must tailor task-oriented mobile services specifically to the need of mobile users. Below we survey existing work in the literature on Mobile-Data-as-a-Service, Mobile-Computing-as-a-Service, Mobile-Multimedia-as-a-Service and Location-Based Mobile Cloud Services. We note that these task-oriented mobile services do not cover the entire spectrum and more research effort is called for to create more human centric task-oriented mobile services, thus expanding the list of task-oriented mobile services used by mobile users.

4.2.1 Mobile-Data-as-a-Service

More data is sent by mobile devices than it was in 2008 and one of the major reasons is because cloud computing applications bring more mobile data [51]. The data can be heterogeneous and often high-dimensional. Furthermore, mobile data are sent to the cloud as they are collected dynamically with temporal and spatial information. With scalable computation and storage provided by MCC, mobile-data-as-a-service customizes the user need to provide customized data analytics service, leveraging indexing techniques for spatio-temporal data storage and retrieval [89] such as B-Tree [90], R-tree [91, 92], and TPR-tree [93, 94, 95]. GPAC was proposed in [96] for continuously verifying mobile queries for querying on mobile data. Due to the long sequence of mobile data, segmenting part of them is helpful to understand the entire sequence. Guo et al. [52] designed an adaptive method for segmenting multi-dimensional data.

4.2.2 Mobile-Computing-as-a-Service

To improve the QoS when serving a mobile device, one solution is to shorten the transmission delay between the cloud server and the mobile device. A typical implementation is to use a server physically close to the mobile device to instantaneously serve the request and then migrate the VM from the server to other more powerful servers, such as a data center, for further processing. VM migration can improve cloud performance. When a mobile device requests a service, the same VM can be reached from several locations. A VM can be migrated to a cloud server near the user location so as to shorten the access delay and data delivery delay. VM migration can also balance load among several data centers or cloud servers, shifting bursty traffic to other locations.

Migrating and routing VMs on the cloud is an on-going research issue. Hao et al. [53] devised virtual routers by attaching several devices to the virtual ports of one virtual router. Forwarding routers each called a forward element (FE) are distributed in multiple locations. A central controller coordinates FEs and tracks the current locations of VMs. Outbound traffic going out of VMs will be sent to outside networks.

4.2.3 Mobile Multimedia-as-a-Service

With scalable computation and flexible network service delivery, MCC is especially ideal for multimedia data storage and distribution. Zhu et al. [54] envisioned the following challenges for supporting mobile multimedia computing: (a) a multimedia request immediately triggers a multitude of actions on the cloud including data storage, retrieval, and distribution for load balancing which must be handled at runtime; (b) MCC must provide QoS provisioning for multimedia applications and adapt to dynamically changing network and workload conditions with parallel processing and dynamic offloading support. Ferretti et al. [55] proposed an architecture for QoS provisioning of mobile multimedia applications. Implementing the mobile multimedia service by IaaS, a VM acting as a proxy between the cloud and the mobile client is launched and migrated with the mobile client. Nan et al. [56] addressed resource allocation in a multimedia cloud based on queueing network analysis. They considered three queues in the multimedia cloud for scheduling, computation, and data transmission of multimedia services. They considered both single-class and multiple-class services, with the objective to minimize the processing time and resource cost. Energy-as-a-Service (EaaS) was proposed in [57] with the objective to reduce energy consumption of multimedia applications running on smartphones. It compares power consumption data under several circumstances, and reveals that video conversion performed under some conditions will reduce the downloading energy for smartphone devices.

4.2.4 Location-Based Mobile Cloud Services

Location-Based services are often embedded in applications running on mobile devices. Many providers have already established location-based services, such as Intel [58] and AT&T [59]. Most mobile devices are equipped with GPS which can reveal the user location. After a mobile device connects with the nearest cloud, the cloud server can immediately push advertisements or the user can query the cloud server. Besides location-based services, one can also obtain friends/family information from the cloud. Tan et al. [60] explored the potential benefits of cloud- and crowd-assisted GPS to aid GPS calculations and, leveraging location corrections from other mobile device users, to enhance noisy GPS readings. Cho et al. [61] combined location-based services with social networking based on social relationship and human mobility. WhereStore [62] is a store location algorithm based on the location history of a smartphone. Location-based store data do not need to be downloaded each time as long as a local copy exists and the data item has not been updated. WhereStore also can predict a mobile device's future location based on the location history. Linking these two factors together, a smartphone can decide whether or not to cache store data based on location information.

4.3 Elasticity and Scalability

The third design issue is to provide scalable and elastic computation and storage service to mobile clients. *Elasticity* means that the resource capability appears to be unbounded and can be purchased in any quantity at any time [97]. *Scalability* relies on allocating proper computing resources to each

VM and allowing VM migration for load balancing across multiple clouds or data centers [11, 98].

Ganesh et al. designed Amoeba [63] for achieving elasticity for data intensive service computing (DISC). DISC applications are usually processed in parallel. A small latency would result if each task is assigned to the identical compute slots for meeting its service level objective (SLO). However, DISC applications frequently have unpredictable overheads. The authors define "elastic jobs" as relinquishing the slot without rejecting new coming jobs. Amoeba provisions lightweight elasticity by which the original DISC application is correctly partitioned without effecting the computational outcome, thus resources are being elastically shared. Moreover, because of resource multiplexing, more jobs can be accommodated.

Resource allocation is a crucial issue for achieving scalability and elasticity. When a cloud service provider allocates resources to a VM, it may allocate CPU, memory, disk I/O and network bandwidth resources, and then apply fault tolerance mechanisms. Rai et al. [64] proposed a resource allocation scheme called Wrasse by mapping cloud resources to a single multi-dimensional resource vector, with each dimension representing one resource with a fixed capacity. Servers and VMs are represented by bins and balls, respectively. A ball can be assigned to any bin, but no more than one bin. Dynamic resource allocation is achieved by first mapping services to balls and then performing ball-to-bin assignment dynamically reflecting service and resource status changes.

Zeta [65] is a scheduling algorithm with the objective to to improve the response quality to accomplish the expected QoS with fewer resources or to achieve a better QoS with the same amount of resources. Tumanov et al. [66] investigated scheduling issues for MCC applications with various resource needs. Two constraints are considered: hard constraints for which the resource requirement must be achieved and soft constraints for which the resource requirement is satisfied on a best effort basis. The utility theory is being used as a tool for resource scheduling to satisfy hard constraints while maximizing the probability of satisfying soft constraints. Shen et al. designed CloudScale [67] for scalable, elastic resource management. CloudScale monitors VMs in CPU, memory, disk I/O usage, and also tracks SLO. CloudScale allocates CPU and memory cycles utilizing the Xen credit scheduler and the setMemoryTarget API, respectively. The resource usage history is used for predicting short-term resource demands and scheduling conflicts. A migration-based schedule conflict handler determines the VM migration decision.

The difficulty of achieving scalability and elasticity stems from highly heterogeneous workloads, highly dynamic demands for resources, poorly predicted resource utilizations, and a high variability of resource preferences and constraints among heterogeneous service classes in MCC applications. Reiss et al. [99] analyzed Google traces and discovered that real-world workloads tend to be very heterogeneous as opposed to homogeneous as frequently assumed in existing work. Thus, a research challenge in scalability and elasticity is to first model and validate traffic models for MCC applications and then devise and validate resource management and real-time scheduling algorithms using traffic models most suitable for specific MCC applications.

4.4 Security

Before convincing more mobile users to run MCC applications, many security issues must be resolved. On the cloud side, it is a must to make sure a user accessing cloud resources is the one it claims to be. Cloud servers must be secure themselves and protect data from intrusion. On the user side, a mobile user needs to authenticate the cloud; that is, the cloud is the one it claims to be; secondly, it needs assurance of data/code integrity; third, it needs assurance that the user application will run only with permission.

4.4.1 Authentication

The foremost security challenge for MCC applications is authentication. Authentication is bidirectional, i.e., the mobile client must authenticate to the cloud (called mobile-to-cloud authentication) and the cloud must authenticate to the mobile client (called cloud-to-mobile authentication).

In the literature, Harbitter et al. [68] analyzed the performance of Kerberos-based mobile-to-cloud authentication with a proxy created between the mobile client and the cloud. They discovered that the proxy is a performance bottleneck. Another problem of Kerberos-based authentication is its vulnerability to attacks due to password only protection. Al-Muhtadi et al. [69] investigated a mobile-to-cloud authentication framework for ubiquitous computing environments using wearable devices, such as active badges, smart jewelry, smart watches, and biometrics. This provides a better protection for authenticating a mobile user's identity than password only protection. Focusing on both security and usability, Bonneau et al. [70] investigated web authentication. Since the end user tends to carry more than one mobile devices, they proposed an authenticating scheme by which the same password may be used for several devices, as long as one of them passes authentication. The password is encrypted by a master password which is stored in the cloud and is effective for all other devices belonging to the same end user. The merit of this protocol is usability. However, protecting the secrecy of the master password is a vulnerability. OpenID [71] is another mobile-to-cloud authentication protocol by which a server asks for other available servers to serve as trusted identity servers to authenticate mobile users. OpenID possesses the advantage of usability but not security. It is subject to cookie stealthy and phishing attacks. Phoolproof [72] is another authentication protocol without the need of a trusted third party. It is specifically designed for mobile banking applications utilizing PKI technology based on the assumption that mobile devices are operated by the intended mobile users.

All prior work cited above is for mobile-to-cloud authentication. While the literature is abundant in mobile-to-cloud authentication protocol design, there is little research done in cloud-to-mobile authentication. This deserves more research attention as MCC authentication must be bidirectional.

4.4.2 Authorization

Another major security issue for MCC applications is authorization. Authorizing an application to access user data without releasing the user's credential can save energy of mobile devices and reduce data transportation overhead. OAuth 2.0 with single sign-on (SSO) [73] is widely applied in Google, Facebook and Twitter. Prior studies [74, 75, 76] have verified its security. Hammer-Lahav [77] reveals that OAuth 2.0 without signature is risky for users when sending the request to a wrong server. Miculan et al. [78] and Shepard [79] analyzed the implementation of OAuth in Facebook. Paul [80] analyzed the misuse of OAuth 2.0 in Twitter. Sun et al. [74] studied its vulnerability by tracing SSO credentials. They discovered that the access token embedding scope and timeout information for authorization is subject to copying attacks. That is, if another entity has a copy, it will be authorized to accessing user data.

4.4.3 Data/Code Integrity

A classic method for securing data integrity is to store encrypted data on the remote server. It is feasible for a desktop computer to encrypt data before uploading it to the cloud. However, due to the computation cost for running the encryption algorithm, it is not practical to encrypt data on a smartphone. van Dijk et al. [81] developed a protocol allowing a mobile user to encrypt and store data in the cloud. The mobile user needs to store the encryption key on the cloud and upload the plaintext to the cloud for encryption. The user data is divided into equal length blocks, each being encrypted using the encryption key provided, and an image is made out of each encrypted block. A mobile user verifies whether its data are encrypted properly by randomly selecting a block and challenging the cloud server to return the image of the block selected as a response. If the cloud server can return the image of the data block requested within a timeout interval and the returned image matches with that locally computed by the mobile device, then the mobile user considers its data as having been encrypted properly. The timeout interval is the maximum amount of time needed for the cloud server to transmit the image of the requested block to the mobile user, which is much smaller than the time for performing encryption of the data block selected anew and generating an image out of the encrypted data block. The problem with this approach is that it may be difficult to define a proper timeout interval as many factors (e.g., limited wireless bandwidth) may affect the wireless transmission time.

A mobile client also must ensure the application code running on the cloud is indeed the same application code it authorizes the cloud to run (called code integrity). There is little research work reported in the literature regarding code integrity. For computation/code offloading, code integrity is essential to ensure the correctness of application execution. Techniques based on code attestation [82] may be applicable for this research avenue.

4.5 Cloud Pricing

Cloud pricing is specific to cloud computing. Since users can request more resources by paying more, cloud pricing can affect cloud service disciplines and admission policies.

Several cloud pricing schemes have been proposed in the literature, particularly in bidding for VMs (for VM migration) to increase the service provider's profit [83, 84, 85] and/or to minimize the user's cost [86, 87]. Zaman et al. [83] developed a dynamic VM provisioning and allocation scheme based on auctioning/bidding. Wang et al. [84] studied cloud capacity segmentation with a hybrid pricing scheme base on Amazon EC2 [100]. EC2 offers three kinds of pricing schemes: pay-as-you-go (fixed pricing), the spot market (dynamic pricing), and subscription. Utilizing an auctioning model where bidders can periodically bid for resources and use them as long as needed, they discovered that the provider's revenue will be higher under the spot market pricing scheme. Lampe et al. [85] used linear programming to solve the problem of Equilibrium Price Auction Allocation (EPAAP) with two constraints: the provider has to set up specific equilibrium prices for VMs and allocate the VMs to winning bidders. The objective function is the provider's profit calculated by the revenue earned from serving winning bidders minus the cost for VM migration. Zafer et al. [86] designed a VM spot bidding policy by which the bidding price is compared with the realtime spot price for a VM. The provider can terminate the VM whenever the bidding price is less than the realtime spot price. The user can continue to bid for a new VM. Tang et al. [87] utilized a constrained Markov decision process to decide optimal pricing that will result in the lowest cost with acceptable reliability. Li et al. [88] designed a pricing mechanism for a cloud bank under which several providers provision IaaS. They developed a game theoretical approach with three participants: resource providers, the cloud bank, and resource consumers. The resource providers and cloud bank have their own profit functions based on cloud bank bargaining prices to resource providers and consumers. The cloud bank deposits resources as rewards to resource providers and rents resources to consumers, such that the best bargaining price exists to minimize the total cost.

Summarizing above, the literature is abundant in cloud service pricing. The approaches used for determining optimal pricing range from auctioning/bidding, game theory, to linear programming to maximize the service provider's profit and/or to minimize the mobile user's cost in terms of the VM migration cost, computation cost, network cost, and energy cost. However, existing work cited above is theoretical in nature. A further research direction is to validate theoretical results with empirical studies using real traces as input.

Another future research area is to analyze the effect of pricing changes to input traffic to the cloud so that the system can adjust optimal pricing periodically as opposed to adopt dynamic pricing (e.g., the spot market scheme) which is disturbing to end users. The approach proposed by Yilmaz et al. [101, 102] utilizing call admission control for pricing optimization may be applicable in this research avenue.

5 Conclusion

In this paper we discussed perspectives and challenges of migrating from mobile computing to mobile cloud computing. We surveyed existing and speculated future generation mobile cloud computing applications. We identified five major challenges, namely, code/computation offloading, task-oriented cloud services, elasticity and scalability, security, and cloud pricing, for building the next generation mobile cloud computing applications. We provided a survey of existing solutions toward each of these challenges and suggested future research directions. Lastly, we note that trust management to support mobile cloud computing is a totally unexplored area, especially for an ad hoc mobile cloud comprising mobile nodes as resource providers without involving a remote cloud. A future research direction is to apply and possibly combine existing trust management techniques for cloud computing [103], social networks [104], mobile ad hoc networks [105, 106], sensor networks [107, 108], and delay tolerant networks [109] into one that can best enhance security and service quality assurance of mobile cloud computing.

References

[1] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, 2014, pp. 369-392.

[2] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, January 2013.

[3] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, 2013.

[4] R. Buyya, “Introduction to the IEEE Transactions on Cloud Computing,” *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, 2014, pp. 3-9.

[5] P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, National Institute of Standards and Technology Std. 800-145, September 2011. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[6] G. D’Angelo, S. Ferretti, V. Ghini, and F. Panzieri “Mobile Computing in Digital Ecosystems: Design Issues and Challenges,” *7th IEEE Wireless Communications and Mobile Computing Conference*, 2011, pp. 2127-2132.

[7] Y. Li and I.R. Chen, “Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 3, 2011, pp. 349-361.

[8] H. Hu, Y. Wen, T.S. Chua and X. Li, “Toward Scalable Systems for Big Data Analytics: A Technology Tutorial,” *IEEE Access*, vol. 2, 2014, pp. 652-687.

[9] A.U.R. Khan, M. Othman, S.A. Madani, and S.U. Khan, “A Survey of Mobile Cloud Computing Application Models,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, 2014, pp. 393-413.

[10] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A break in the clouds: towards a cloud definition,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, December 2008.

[11] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: state-of-the-art and research challenges,” *Journal of Internet Services and Applications*, vol. 1, pp. 7–18, 2010.

[12] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud computing and grid computing 360-degree compared,” in *Grid Computing Environments Workshop*, nov. 2008, pp. 1 –10.

[13] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *IEEE Security Privacy*, vol. 9, no. 2, pp. 50 –57, March-April 2011.

[14] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *ACM Communications*, vol. 53, no. 4, pp. 50–58, April 2010.

[15] L. Yang, J. Cao, S. Tang, T. Li, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," in *IEEE 5th International Conference on Cloud Computing*, June 2012, pp. 794 –802.

[16] A. Klein, C. Mannweiler, J. Schneider, and H. D. Schotten, "Access schemes for mobile cloud computing," in *2010 Eleventh International Conference on Mobile Data Management*, May 2010, pp. 387 –392.

[17] P. Bahl, R. Y. Han, L. E. Li, and M. Satyanarayanan, "Advancing the state of mobile cloud computing," in *ACM workshop on Mobile cloud computing and services*, New York, NY, USA, 2012, pp. 21–28.

[18] E. Miluzzo, R. Cáceres, and Y.-F. Chen, "Vision: mclouds - computing on clouds of mobile devices," in *ACM workshop on Mobile cloud computing and services*, 2012, pp. 9–14.

[19] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn, "The rise of people-centric sensing," *IEEE Internet Computing*, vol. 12, no. 4, pp. 12–21, 2008.

[20] M. Satyanarayanan, "Mobile computing: the next decade," in *ACM Workshop on Mobile Cloud Computing Services: Social Networks and Beyond*, 2010, pp. 1–6.

[21] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," *ACM MobiCom*, 2012.

[22] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[23] H. Cheng, F. Sun, S. Buthpitiya, and M. Griss, "Sensorchestra: Collaborative sensing for symbolic location recognition," *Mobile Computing, Applications, and Services*, pp. 195–210, 2012.

[24] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *International conference on Mobile systems, applications, and services*, 2009, pp. 165–178.

[25] "Sensorly." [Online]. Available: http://www.sensorly.com/

[26] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *ACM Conference on Embedded Networked Sensor Systems*, 2009, pp. 85–98.

[27] R. Herring, A. Hofleitner, S. Amin, T. Nasr, A. Khalek, P. Abbeel, and A. Bayen, "Using mobile phones to forecast arterial traffic through statistical learning," *Transportation Research Board*, 2009.

[28] T. Hunter, T. Moldovan, M. Zaharia, S. Merzgui, J. Ma, M. Franklin, P. Abbeel, and A. Bayen, "Scaling the mobile millennium system in the cloud," in *ACM Symposium on Cloud Computing*, 2011, p. 28.

[29] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *ACM international conference on Mobile systems, applications, and services*, 2009, pp. 55–68.

[30] O. Khalid, M. Khan, S. Khan, and A. Zomaya, "OmniSuggest: A Ubiquitous Cloud based Context Aware Recommendation System for Mobile Social Networks," *IEEE Transactions on Services Computing*, 2014.

[31] Y. Wang, J. Wu, and W.S. Yang, "Cloud-Based Multicasting with Feedback in Mobile Social Networks," *IEEE Transactions on Wireless Communications,* vol. 12, no. 12, 2013, pp. 6043-6053.

[32] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell, “Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application,” in *ACM conference on Embedded network sensor systems*, 2008, pp. 337–350.

[33] R. Cimler, J. Matyska, and V. Sobeslav, “Cloud based solution for mobile healthcare application,” *ACM 18th International Database Engineering and Applications Symposium,* July 2014.

[34] H. Wu, Q. Wang and K. Wolter, “Mobile Healthcare Systems with Multi-cloud Offloading,” *IEEE 14th International Conference on Mobile Data Management*, 2013, pp. 188-193.

[35] S. Consolvo, D. McDonald, T. Toscos, M. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby *et al.*, “Activity sensing in the wild: a field trial of ubifit garden,” in *ACM SIGCHI conference on Human factors in computing systems*, 2008, pp. 1797–1806.

[36] K. Tamai and A. Shinagawa, “Platform for location-based services,” *Fujitsu Sci. Tech. J*, vol. 47, no. 4, pp. 426 – 433, 2011.

[37] H. La and S. Kim, “A conceptual framework for provisioning context-aware mobile cloud services,” in *IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 466–473.

[38] K. Kangas and J. Röning, “Using code mobility to create ubiquitous and active augmented reality in mobile computing,” in *ACM/IEEE international conference on Mobile computing and networking*, 1999, pp. 48–58.

[39] X. Luo, “From augmented reality to augmented computing: A look at cloud-mobile convergence,” in *International Symposium on Ubiquitous Virtual Reality*, July 2009, pp. 29–32.

[40] M.V. Barbera, S. Kosta, A. Mei, and J. Stefa, “To Offload or Not to Offload? The Bandwidth and Energy Costs of Mobile Cloud Computing,” *IEEE INFOCOM 2013*, Turin, Italy, 2013.

[41] G. Hunt, M. Scott *et al.*, “The coign automatic distributed partitioning system,” *ACM Operating systems review*, vol. 33, pp. 187–200, 1998.

[42] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso, “Calling the cloud: Enabling mobile phones as interfaces to cloud applications,” *Middleware 2009*, pp. 83–102, 2009.

[43] E. Cooper, S. Lindley, P. Wadler, and J. Yallop, “Links: Web programming without tiers,” in *Formal Methods for Components and Objects*. Springer, 2007, pp. 266–296.

[44] F. Yang, J. Shanmugasundaram, M. Riedewald, and J. Gehrke, “Hilda: A high-level language for data-drivenweb applications,” in *IEEE International Conference on Data Engineering*, 2006, pp. 32–32.

[45] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “CloneCloud: elastic execution between mobile device and cloud,” in *ACM Conference on Computer systems*, 2011, pp. 301–314.

[46] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “Maui: Making smartphones last longer with ccode offload,” in *AMC international conference on Mobile systems, applications, and services*, New York, NY, USA, 2010, pp. 49–62.

[47] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, “Cuckoo: A computation offloading framework for smartphones,” in *Mobile Computing, Applications, and Services*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2012, vol. 76, pp. 59–79.

[48] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in *IEEE INFOCOM*, March 2012, pp. 945 –953.

[49] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, “The remote processing framework for portable computer power saving,” in *ACM symposium on Applied computing*, 1999, pp. 365–372.

[50] T. Newhouse and J. Pasquale, “Resource-controlled remote execution to enhance wireless network applications,” in *Workshop on Applications and Services in Wireless Networks*, August 2004, pp. 30 – 38.

[51] "Cloud computing drives mobile data growth," October 2009. [Online]. Available: http://spectrum.ieee.org/telecom/wireless/cloud-computing-drives-mobile-data-growth

[52] T. Guo, Z. Yan, and K. Aberer, "An adaptive approach for online segmentation of multi-dimensional mobile data," in *ACM International Workshop on Data Engineering for Wireless and Mobile Access*, 2012, pp. 7–14.

[53] F. Hao, T. Lakshman, S. Mukherjee, and H. Song, "Enhancing dynamic cloud-based services using network virtualization," in *ACM workshop on Virtualized infrastructure systems and architectures*, 2009, pp. 37–44.

[54] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59–69, 2011.

[55] S. Ferretti, V. Ghini, F. Panzieri, and E. Turrini, "Seamless support of multimedia distributed applications through a cloud," in *IEEE 3rd International Conference on Cloud Computing*, July 2010, pp. 548–549.

[56] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud based on queuing model," in *IEEE 13th International Workshop on Multimedia Signal Processing*, October 2011, pp. 1 –6.

[57] M. Altamimi, R. Palit, K. Naik, and A. Nayak, "Energy-as-a-service (EaaS): On the efficacy of multimedia cloud computing to save smartphone energy," in *IEEE 5th International Conference on Cloud Computing*, June 2012, pp. 764–771.

[58] "Int. cloud services platform beta location-based services," September 2012. [Online]. Available: http://software.intel.com/en-us/articles/cloud-services-location-based-api-overview

[59] "AT&T to launch cloud-based lbs mobility data offering," January 2011. [Online]. Available: http://www.mobilecommercedaily.com/att-to-launch-cloud-based-lbs-mobility-data-offering

[60] Z. Tan, D. Chu, and L. Zhong, "Vision: cloud and crowd assistance for GPS urban canyons," *ACM Mobisys*, 2014, pp. 23-27.

[61] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1082–1090.

[62] P. Stuedi, I. Mohomed, and D. Terry, "Wherestore: location-based data storage for mobile devices interacting with the cloud," in *ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond*, 2010, pp. 1–8.

[63] G. Ananthanarayanan, C. Douglas, R. Ramakrishnan, S. Rao, and I. Stoica, "True elasticity in multi-tenant data-intensive compute clusters," in *ACM Symposium on Cloud Computing*, 2012, pp. 1–7.

[64] A. Rai, R. Bhagwan, and S. Guha, "Generalized resource allocation for the cloud," in *ACM Symposium on Cloud Computing*, pp. 15:1–15:12.

[65] Y. He, S. Elnikety, J. Larus, and C. Yan, "Zeta: scheduling interactive services with partial execution," in *ACM Symposium on Cloud Computing*, 2012, pp. 1–14.

[66] A. Tumanov, J. Cipar, G. R. Ganger, and M. A. Kozuch, "alsched: algebraic scheduling of mixed workloads in heterogeneous clouds," in *ACM Symposium on Cloud Computing*, 2012, pp. 1–7.

[67] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "CloudScale: elastic resource scaling for multi-tenant cloud systems," in *ACM Symposium on Cloud Computing*, 2011, pp. 1–14.

[68] A. Harbitter and D. A. Menascé, "The performance of public key-enabled kerberos authentication in mobile computing applications," in *ACM conference on Computer and Communications Security*, 2001, pp. 78–85.

[69] J. Al-Muhtadi, A. Ranganathan, R. Campbell, and M. Mickunas, "A flexible, privacy-preserving authentication framework for ubiquitous computing environments," in *International Conference on Distributed Computing Systems Workshops*, 2002, pp. 771 – 776.

[70] J. Bonneau, C. Herley, P. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web

authentication schemes," in *IEEE Symposium on Security and Privacy*, 2012, pp. 553–567.

[71] D. Recordon and D. Reed, "Openid 2.0: a platform for user-centric identity management," in *ACM workshop on Digital identity management*, 2006, pp. 11–16.

[72] B. Parno, C. Kuo, and A. Perrig, "Phoolproof phishing prevention," *Financial Cryptography and Data Security*, pp. 1–19, 2006.

[73] "Oauth2.0." [Online]. Available: http://oauth.net/2/

[74] S.-T. Sun and K. Beznosov, "The devil is in the (implementation) details: an empirical analysis of oauth sso systems," in *ACM conference on Computer and communications security*, 2012, pp. 378–390.

[75] S. Chari, C. Jutla, and A. Roy, "Universally composable security analysis of oauth v2. 0," 0. Cryptology ePrint Archive, Report 2011/526, Tech. Rep., 2011.

[76] S. Pai, Y. Sharma, S. Kumar, R. Pai, and S. Singh, "Formal verification of oauth 2.0 using alloy framework," in *IEEE International Conference on Communication Systems and Network Technologies*, 2011, pp. 655–659.

[77] "Oauth 2.0 (without signatures) is bad for the web." [Online]. Available: http://hueniverse.com/2010/09/oauth-2-0-without-signatures-is-bad-for-the-web/

[78] M. Miculan and C. Urban, "Formal analysis of facebook connect single sign-on authentication protocol," in *SOFSEM*, vol. 11, 2011, pp. 22–28.

[79] "Under the covers of oauth 2.0 at facebook." [Online]. Available: http://www.sociallipstick.com/?p=239

[80] "Compromising twitter's oauth security system." [Online]. Available: http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/GENPRESS/A090903P.pdf

[81] M. van Dijk, A. Juels, A. Oprea, R. L. Rivest, E. Stefanov, and N. Triandopoulos, "Hourglass schemes: how to prove that cloud files are encrypted," in *ACM conference on Computer and communications security*, 2012, pp. 265–280.

[82] I. Chen and Y. Wang, "Reliability analysis of wireless sensor networks with distributed code attestation," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1640 –1643, October 2012.

[83] S. Zaman and D. Grosu, "Combinatorial auction-based dynamic vm provisioning and allocation in clouds," in *IEEE Third International Conference on Cloud Computing Technology and Science*, 2011, pp. 107–114.

[84] W. Wang, B. Li, and B. Liang, "Towards optimal capacity segmentation with hybrid cloud pricing," in *IEEE 32nd International Conference on Distributed Computing Systems*, 2012, pp. 425–434.

[85] U. Lampe, M. Siebenhaar, A. Papageorgiou, D. Schuller, and R. Steinmetz, "Maximizing cloud provider profit from equilibrium price auctions [pre-print]," *Bid*, vol. 1, no. B1, pp. 0–20.

[86] M. Zafer, Y. Song, and K. Lee, "Optimal bids for spot vms in a cloud for deadline constrained jobs," in *IEEE 5th International Conference on Cloud Computing*, 2012, pp. 75–82.

[87] S. Tang, J. Yuan, and X. Li, "Towards optimal bidding strategy for amazon ec2 cloud spot instance," in *IEEE 5th International Conference on Cloud Computing*, 2012, pp. 91–98.

[88] H. Li and H. Li, "A research of resource provider-oriented pricing mechanism based on game theory in cloud bank model," in *IEEE International Conference on Cloud and Service Computing*, 2011, pp. 126–130.

[89] M. Mokbel, T. Ghanem, and W. Aref, "Spatio-temporal access methods," *IEEE Data Engineering Bulletin*, vol. 26, no. 2, pp. 40–49, 2003.

[90] C. Jensen, D. Lin, and B. Ooi, "Query and update efficient b+-tree based indexing of moving objects," in *International conference on Very large data bases-Volume 30*, 2004, pp. 768–779.

[91] D. Kwon, S. Lee, and S. Lee, "Indexing the current positions of moving objects using the lazy update r-tree," in *IEEE International Conference on Mobile Data Management*, 2002, pp. 113–120.

[92] M. Lee, W. Hsu, C. Jensen, B. Cui, and K. Teo, "Supporting frequent updates in r-trees: a bottom-up approach," in *International Conference on Very Large Databases*, 2003, pp. 608–619.

[93] B. Lin and J. Su, "On bulk loading TPR-tree," in *IEEE International Conference on Mobile Data Management*, 2004, pp. 114–124.

[94] S. Saltenis and C. Jensen, "Indexing of moving objects for location-based services," in *IEEE International Conference on Data Engineering*, 2002, pp. 463–472.

[95] Y. Tao, D. Papadias, and J. Sun, "The TPR*-tree: an optimized spatio-temporal access method for predictive queries," in *International Conference on Very Large Databases*, vol. 29, 2003, pp. 790–801.

[96] M. Mokbel and W. Aref, "Gpac: generic and progressive processing of mobile queries over mobile data," in *IEEE international conference on Mobile Data Management*, 2005, pp. 155–163.

[97] L. Badger, T. Grance, R. Patt-Corner, and J. Voas, "Draft cloud computing synopsis and recommendations," *NIST Special Publication*, vol. 800, p. 146, 2011.

[98] B. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *IEEE International Joint Conference on INC, IMS and IDC*, 2009, pp. 44–51.

[99] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *ACM Symposium on Cloud Computing*, 2012, pp. 1–13.

[100] "Amazon EC2." [Online]. Available: http://aws.amazon.com/ec2/

[101] O. Yilmaz and I.R. Chen, "Utilizing call admission control for pricing optimization of multiple service classes in wireless cellular networks," *Computer Communications*, vol. 32, no. 2, 2009, pp. 317–323.

[102] I.R. Chen, O. Yilmaz and I.L. Yen, "Admission control algorithms for revenue optimization with QoS guarantees in mobile wireless networks," *Wireless Personal Communications*, vol. 38, no. 3, Aug. 2006, pp. 357-376.

[103] T.H. Noor, Q.Z. Sheng, S. Zeadally, and J. Yu, "Trust Management of Services in Cloud Environments: Obstacles and Solutions," *ACM Computing Survey*, vol. 46, no. 1, article 12, Oct. 2013.

[104] W. Sherchan, S. Nepal, and C. Paris, "A Survey of Trust in Social Networks," *ACM Computing Survey*, vol. 45, no. 4, article 47, August 2013.

[105] J.H. Cho, A. Swami, and I.R. Chen, "Modeling and analysis of trust management for cognitive mission-driven group communication systems in mobile ad hoc networks," in *International Conference on Computational Science and Engineering*, August 2009, pp. 641–650.

[106] J.H. Cho, A. Swami, and I.R. Chen, "Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 35, no. 3, 2012, pp. 1001–1012.

[107] F. Bao, I.R. Chen, M.J. Chang and J.H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169-183.

[108] F. Bao, I.R. Chen, M.J. Chang and J.H. Cho, "Trust-based intrusion detection in wireless sensor networks," *IEEE International Conference on Communications (ICC)*, 2011, pp. 1-6.

[109] I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, 2014, pp. 1200-1210.