Evaluation of mobile cloud architectures

Article *in* Pervasive and Mobile Computing · December 2016

DOI: 10.1016/j.pmcj.2016.12.003

2 authors:

Jacques Bou abdo
Notre Dame University

Jacques Demerjian
Lebanese University

Evaluation of Mobile cloud architectures

Abstract

The progress of mobile cloud computing is currently facing a crossroad on whether to implement user-side tools for job offloading or adopt network-side architecture. Selecting the best mobile cloud architecture is vital for having clear decisions on the future of mobile cloud computing.

This work will survey mobile cloud architectures and compare their performance against existing resource demanding applications. The comparison will be performed using physical implementation. To the best of our knowledge, this paper is the first published work numerically comparing mobile cloud architectures across various mobile cloud applications.

I. Introduction

Mobile technology experienced radical changes in its concepts, aims and needs after introducing smartphones and 4G networks. Protocol traffic overhead appears to be no more a critical problem since mobile networks are achieving high throughputs and new technologies (such as 5G) are promising even higher rates [1]. Mobile devices are not necessarily the slim clients that used to dominate the network ten years ago. Smartphones are powerful devices capable of processing majority of mobile applications, but have very limited power resources (e.g. battery) and it is not likely to change in the foreseen future [2]. Both types of mobile devices, computation/storage-limited (slim client) and power-limited (smartphone), need a technology that helps in decreasing power consumption, delay and user's cost without compromising the privacy, availability and mobility offered by mobile networks.

Mobile Cloud Computing (MCC) is an on-demand, dynamic and self-provisioned outsourcing of IT resources from a centralized service provider to mobile users using an access network. It is a technology that offloads resource-intensive applications (or parts of applications) from resource-limited mobile devices to be processed "somewhere else" [3]. This technology aims to decrease mobile device's power consumption, allows complex applications to be managed from mobile devices and, most importantly, keeps the expenses within the user's cost budget.

Research initiatives resulted in different mobile cloud architectures and a huge number of mobile cloud applications, but failed to create a standardized platform that facilitates wide deployment and compatibility between applications, vendors and ultimately businesses. The aim of this work is to compare the performance of existing mobile cloud architectures in supporting resource demanding applications. Performance is measured by using several metrics, but metric weighting is left for the reader to evaluate due to subjective performance definitions and needs. Performance is studied using physical implementation which is considered the most realistic evaluation method. To the best of our knowledge, this paper is the first published work numerically comparing mobile cloud architectures across various mobile cloud applications.

The rest of this paper is organized as follows: section II surveys existing mobile cloud architectures and applications found in literature. Section III presents the environment surrounding the physical implementation of mobile cloud architectures. It introduces the metrics, used applications and architecture configuration. Section IV presents the implementation results and measurements. Section V concludes this work.

II. Survey on Mobile Cloud Architectures and Applications

Mobile Cloud Computing is relatively a new technology having lots of potential applications but no standardized architecture until nowadays. This motivates researchers to innovate different architectures, each trying to emphasize on a certain requirement (such as mobility, power consumption, etc.) and to select the optimal compromise for the others. In this section, we are going to survey existing mobile cloud architectures (physical layer and application layer architectures) and applications found in literature.

a. Mobile cloud physical layer architectures

As a result of many innovative attempts, various mobile cloud architectures have been proposed, each exploiting a technology (such as: ad hoc Wi-Fi, P2P, mobile networks, etc.) that helps in emphasizing on one of MCC's requirements. The MCC architectures found in literature are:

1. Cloud computing with mobile terminals

It is identical to the normal cloud computing architecture where the service (computation, storage, etc.) is implemented in a remote cloud server (within the Cloud Service Provider's), but the terminals, in this case, are mobile devices such as PDAs, Smart Phones, etc. Mobile devices connect to the Internet, in most of the cases, over an expensive connection such as LTE, UMTS and GPRS [4]. Although mobile connection is expensive, it contains many advantages such as handover, availability, location privacy, native encryption, etc. Connecting through Wi-Fi interface is also possible but it creates many concerns regarding mobility such as: network availability, handover, etc. An example on the above concerns is shown in the following use case: The user is accessing his Cloud Service Provider (CSP) using a mobile device over a Wi-Fi network offered at the coffee shop he is currently in. Leaving the coffee shop makes him drop the connection and lose the cloud service. This architecture is shown in figure 1.

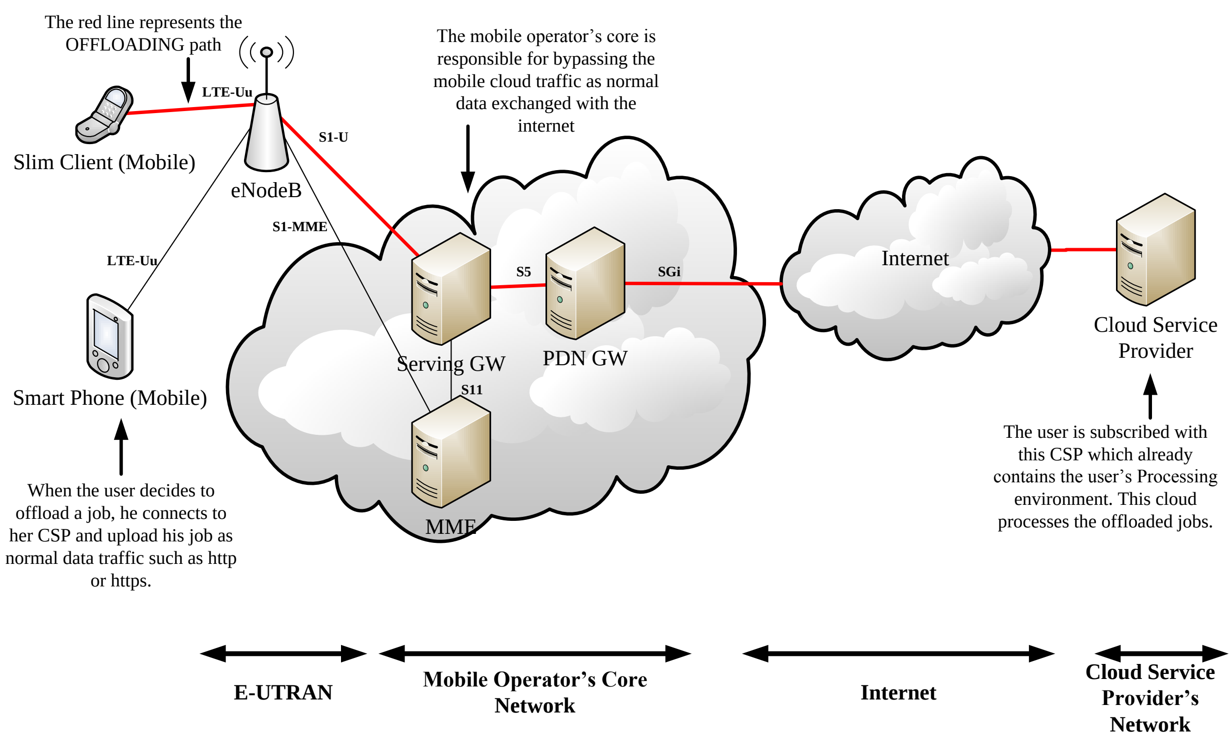

Figure 1 "Cloud computing with mobile terminals" architecture

2. Virtual cloud computing provider

This architecture proposes the creation of a virtual cloud from peer-to-peer connected mobile devices over an ad-hoc Wi-Fi connection to share processing burden [5]. P2P nodes participate only if found in the vicinity (Wi-Fi range) and interested in the processed data. After selecting the mobile devices participating in this virtual cloud, the job requestor divides the job into tasks and offloads, each to a participant. After processing the task, the participant replies back with his processed data which is consolidated by the requestor to get the final result. These results are sent back to the participants. This architecture is shown in figure 2.

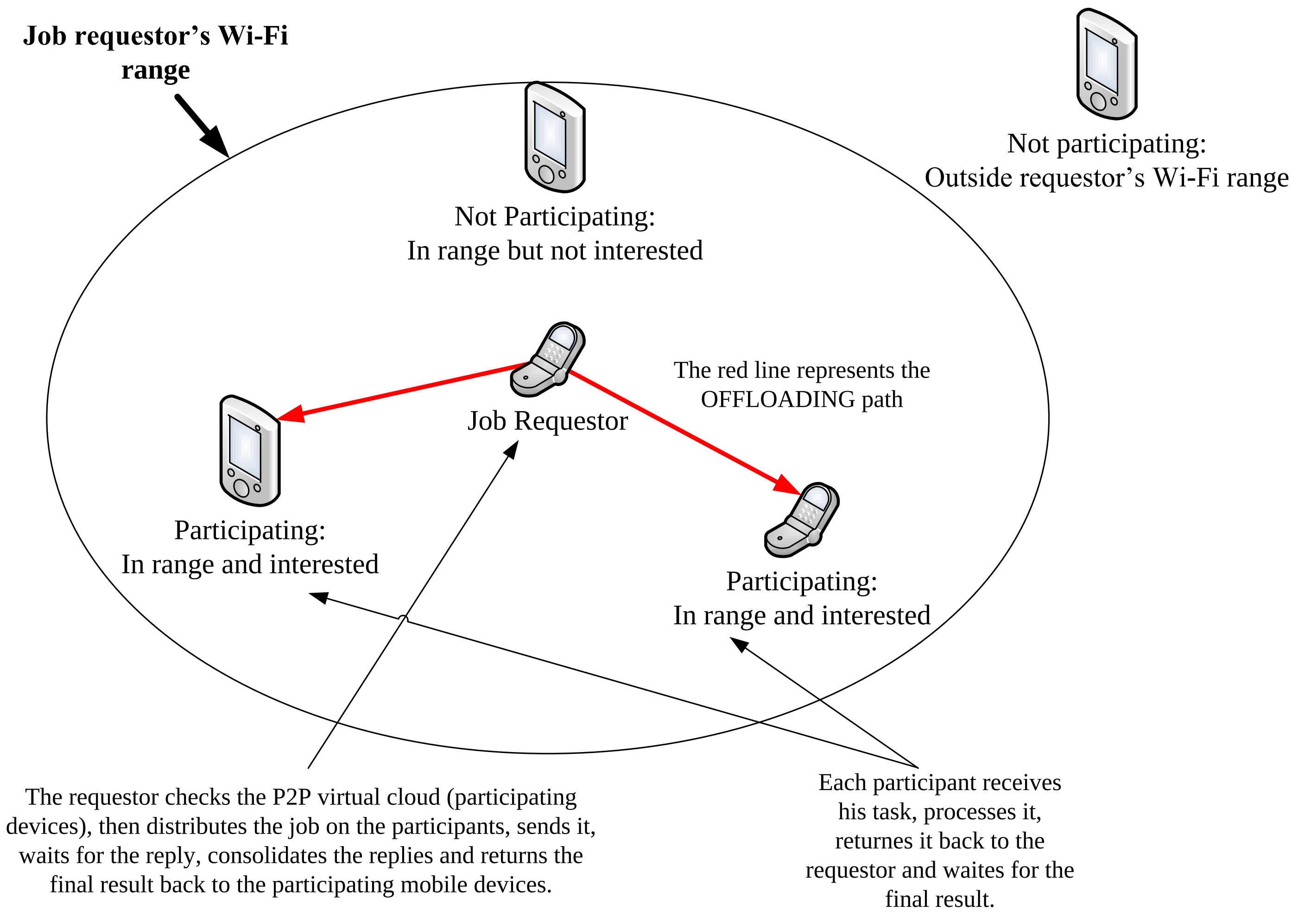

Figure 2 "Virtual cloud computing provider" architecture

3. Cloudlet

This architecture proposes installing cloudlet servers in high density areas (such as: coffee shops, malls, etc.) collocated with Wi-Fi hotspots [6]. The user connects to the cloudlet server and offloads his job to be processed. It contacts the user's CSP to retrieve the user's environment through federation. Starting this point, future jobs will be processed at the cloudlet until getting disconnected (user leaves the hotspot coverage). This architecture is shown in figure 3.

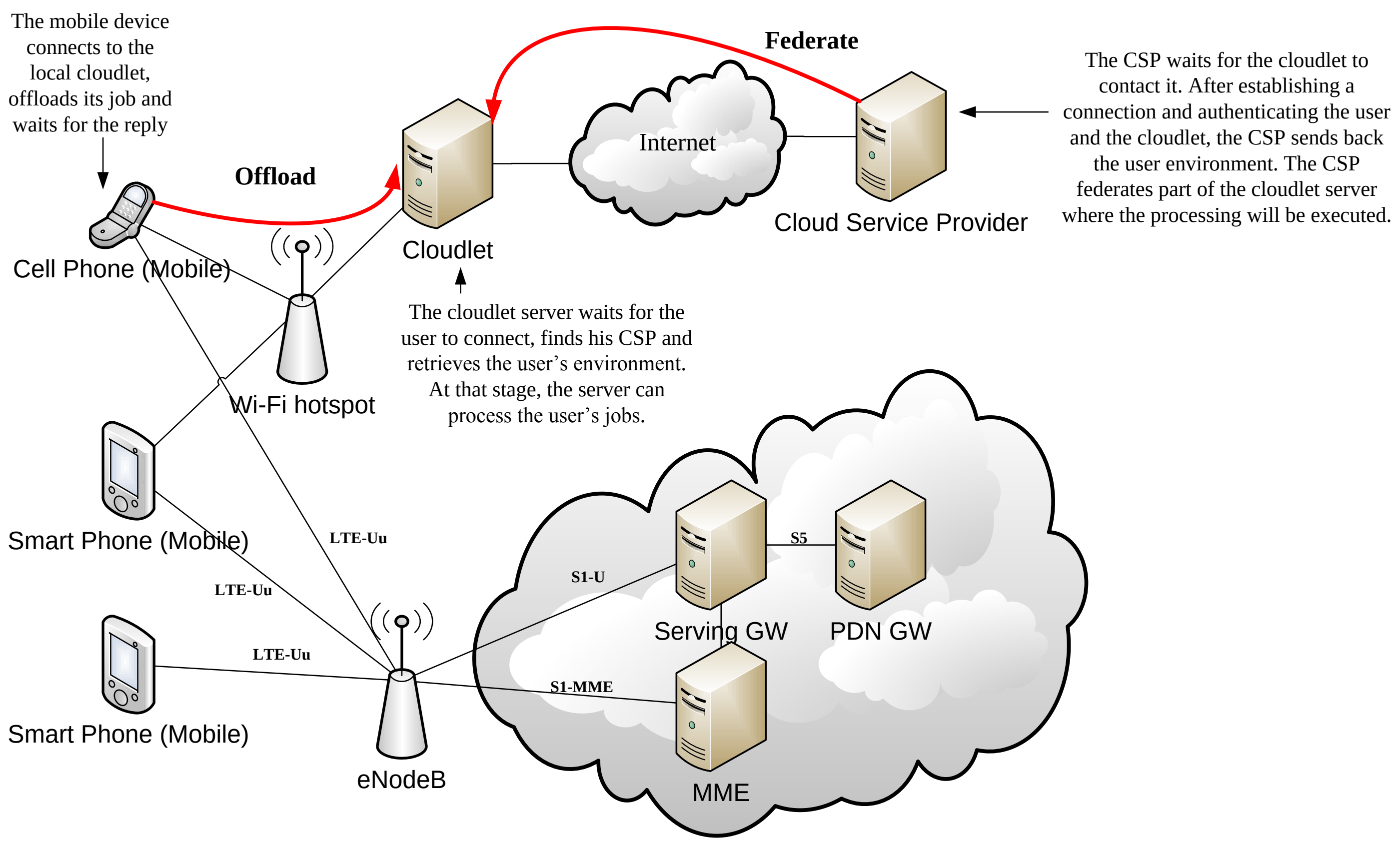

Figure 3 "Cloudlet" architecture

4. Operator Centric Mobile Cloud Architecture (OCMCA)

This architecture proposes installing a cloud server within the mobile operator's network as shown in figure 4 [7]. This makes the cloud closer to the user thus decreases delay without affecting the availability, the location privacy, the handover and all the services offered by mobile communication.

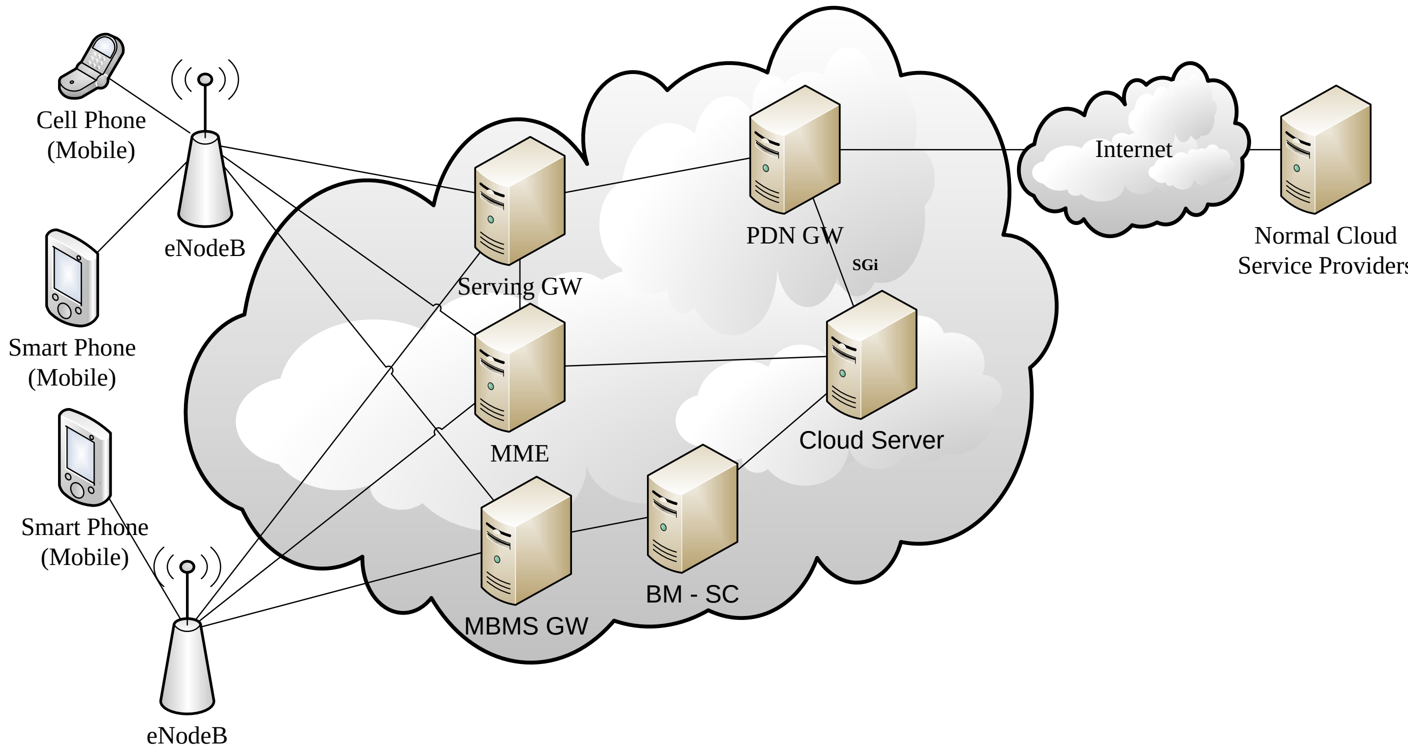

Figure 4 OCMCA

Many mobile applications are multicast-based in nature such as: Google cloud messaging platform [8], multimedia applications and same-content delivery applications (Google play, Apple store, etc.) but its physical implementation and mobile network's characteristics transform its communication into a group of unicast messages leading to unnecessary congestion, to additional delay and to more expensive fees.

This architecture pays special attention to multicast traffic because it considers that top-ranked mobile applications are multicast-based in nature. The traffic paths in OCMCA are shown in figure 5.

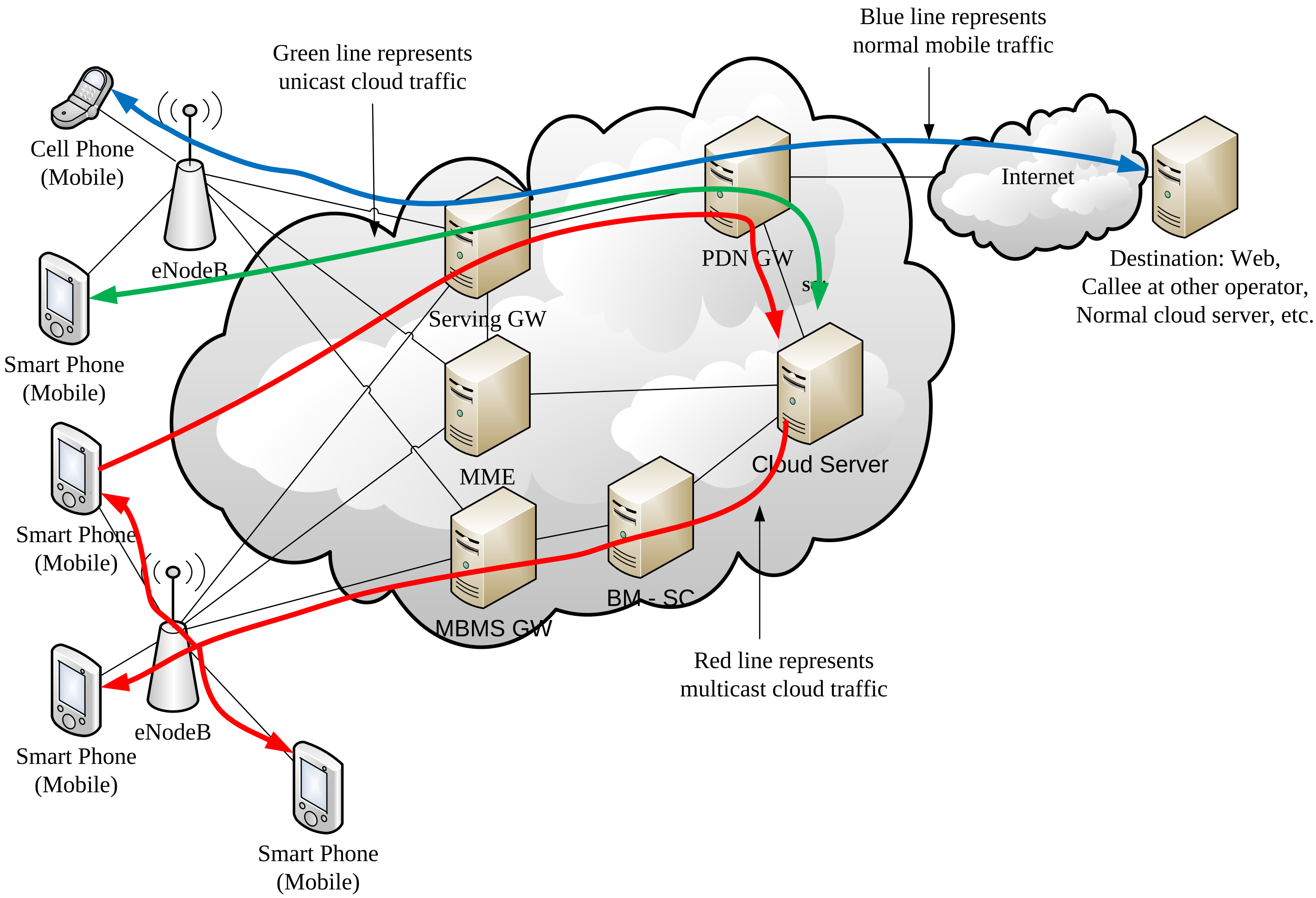

Figure 5 Traffic paths in OCMCA

b. Mobile cloud application layer architectures

Offloading mechanisms are methods to offload resource demanding threads, tasks or portions of an application. They are considered application layer architectures which differ from all the above and try to answer different questions (physical layer architectures answer questions like how to connect to the cloud network?, is user mobility ensured?, how to ensure user confidentiality?; while application layer architectures answer questions like which part of the application would decrease the power consumption if offloaded?) [9]. Application layer architectures are complementary to the physical layer architectures, thus they will not be included in the performance comparison. Available offloading mechanisms are shown below:

1. CloneCloud

It is having a clone of the mobile device running in cloud. This solution uses an application-level virtual machine that can partition an application and run one part on the mobile device and the other at the clone as shown in figure 6. This solution works at the application layer [10] to decide which portion of an application has to be offloaded to the cloud [4] [11] [12].

Cuckoo [13], various Cloudlet versions [14][15], Spectra [16], Chroma [17], Hyrax [18], Mobile Message Passing Interface (MMPI) framework [19], MobiCloud [20] and MAUI [21] are offloading mechanisms and frameworks for dynamic selection of code partitions (pieces of the program) to be executed remotely similar to CloneCloud so they will not be included in the performance comparison.

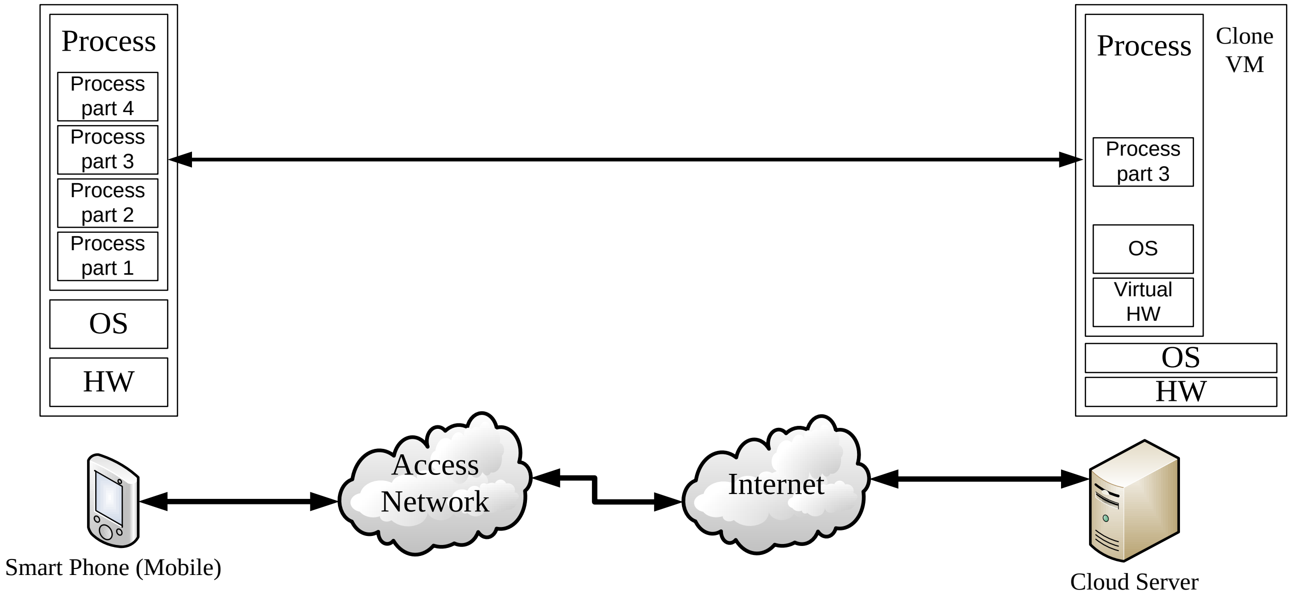

Figure 6 CloneCloud

2. OpenMobster

OpenMobster [22] is an open source project for mobile cloud platform. It provides the mobile applications an opportunity for simple shift into MCC by abstracting the complexities resulting from remote resource accessing. The complexities from the client side are [23]:

- Availability detection of the cloud server.
- Synchronization and queuing of messages to and from the cloud server.
- Network connection management.
- Local database management.
- InterApp communication.

OpenMobster's client-side platform tries to deal with the above complexities using the following services: sync, push, offlineApp, network, database and InterApp bus as shown in figure 7. OpenMobster mediates the communication between the application from one side and the native operating system and remote cloud server from the other.

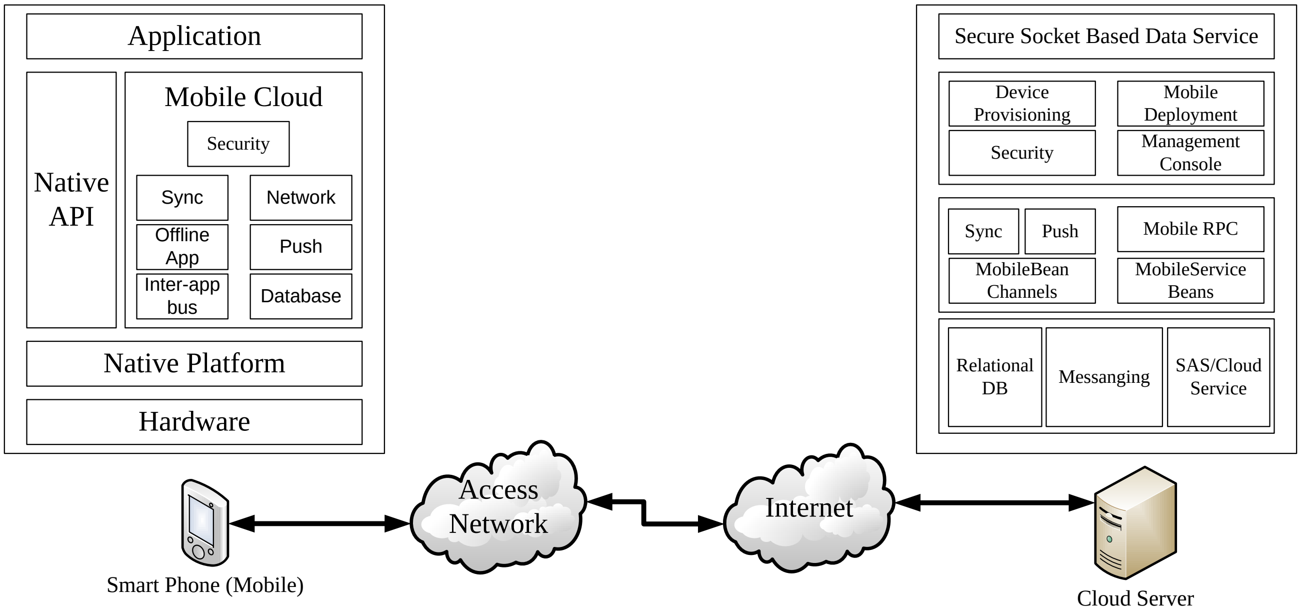

Figure 7 OpenMobster

As seen in figure 7, OpenMobster is another application layer solution which could be used in conjunction with other physical layer mobile cloud architectures. For this reason, it will not be included in the performance comparison.

Application layer architectures have been referenced to using other terms such as application models [24], offloading mechanisms [25], cloud-based mobile application execution frameworks [26] and mobile computing architectures [27]. These architectures have been categorized in [24] into 4 sections as follows:

- Performance Based Application Models
    - CloneCloud [4]
    - Zhang et al. Model [20]
- Energy Based Application Models
    - µ Cloud [28]
- Constraint Based Application Models
    - Satyanarayanan et al. Model [6]
    - Giurgiu et al. Model [10]
    - eXCloud [29]
- Multi-objective Application Models

- MAUI [21]
- ThinkAir [30]
- Cuckoo [13]

c. Mobile cloud Applications

Intensive resource demanding mobile applications are the best candidates for being transformed into mobile cloud applications. It has been shown in [31] that during 2012, games and messaging applications are the most popular from developer side (based on number of developed applications) but Facebook (76% of US Smartphone users), Google Maps (65.9%), Google Play (54.3%), Google Search (53.5%), Gmail (47.6%), YouTube (46.4%), Pandora Radio (42%), Apple iTunes (41%), Cooliris (38%) and Yahoo! Messenger (32%) are the most popular applications from user side (based on the number of downloaded applications). The applications that can benefit from multicast communication (Google Play, YouTube, Pandora Radio, Apple iTunes and Cooliris) constitute a majority of the top ranked mobile applications. In this section, we are going to present different Intensive resource demanding mobile applications presented in literature [2] which is profiled and simulated in later sections. The applications are shown in Table 1. The privacy constraints, found in Table 1, refer to the privacy requirement to be shown in section III.

Table 1 Intensive resource demanding mobile applications

Application group	Application Subgroup	Selected Applications	Logos of the selected applications
OCR (Optical Character Recognition): Is processing an image to extract the found characters/text. Various applications can be implemented on the text after extraction, such as translation.	**Subgroup 1** [5] [2]: A foreign tourist is visiting a museum in South Korea. He is not able to understand an interesting exhibit written in Korean, so he captures the exhibit's image and tries to translate it using an OCR application. The extracted text is then translated to English. Minimum privacy: Low	**Application 1 (OCR Instantly Free) [32]:** Is an application that performs optical character recognition "which is a technology to convert image to text" [32]. It is available for major smartphone vendors running android or iOS. This application is selected since it is the highest installed Arabic language supporting Android OCR application.	

Natural language processing: Is a useful tool for travelers to be able to communicate with locals.	**Subgroup 2** [34]: Text-to-speech is an application allowing a mobile user having a file to be read to locals. Minimum privacy: Low.	**Application 2 (Talk - Text to Voice FREE) [33]:** Is an application that performs text-to-speech. It supports many languages. This application is selected since it is the highest installed Android text to voice application.	
Crowd computing: Is a method allowing different video recordings captured by different mobile devices to construct a single video covering an entire event [18]. Various applications exist such as [36] [37].	**Subgroup 3 (Lost child)** [35]: A five-year old John is attending a parade with his parent in Manhattan. John goes missing and his parents report the incident to the Police, who send out an alert via a message to all mobile phones within two miles radius, requesting them to upload the parade images they have to a server that only the police have access to. John is spotted in some images and his position was located, and he was reunited with his parents. This application requires high privacy as will be shown in section III.b. Privacy is discussed in section III.a. The participation invitation in this application is multicast/broadcast by nature. Minimum privacy: high.	**Application 3 (Crowd bytes) [38]:** It is a crowd sourcing application that allows, among other things, to publish the profiles of found or lost animals (Living) or objects (Non-Living). This application is selected since it is the least installed Android social crowd computing application. Since the tests will be visible to all members, we selected the application with the smallest number of members.	
	Subgroup 4 (Disaster relief) [35]: Electronic maps become useless after a disaster, thus hindering disaster relief teams from performing rescue operations efficiently. Local citizens are asked to use their mobile phones to photograph disaster sites, and upload it to a central server [2]. The collected images are used to create a panoramic view of the sites, thus facilitating the navigation of the relief teams. This application requires high privacy similar to application 3. The participation invitation in this application suits for multicast/broadcast traffic by nature. Minimum privacy: high.	**Application 4 (Brizo) [39]:** Is a crowd sourcing application that is "used to monitor road conditions like flood, wild fire, accident and traffic jam." [39]. This application was selected since it is the least installed Android road monitoring crowd computing application. Since the tests will be visible to all members, we selected the application with the smallest number of members.	

Sharing GPS/Internet data [2]: Instead of reading common data from Internet or GPS by multiple users, one user can download the data and share it with interested nearby devices through local-area or peer-to-peer networks. This application group helps in decreasing the cost and delay [18] resulting from downloading online information.	**Subgroup 5** [2]: Scans using Bluetooth for a co-located device which has a recent GPS reading. Instead of using the expensive GPS connection the needed information can be retrieved from co-located devices. The performance of this application augments in dense events such as: party, football match, etc. Another format of this application is: "Traffic Lights Detector for Blind Navigation" [40]. The requestor's communication with other participants suits for multicast/broadcast traffic by nature. Similar applications can be seen in [41].	**Application 5 (GPS over BT) [42]:** It is an application that reads GPS information and offers it to other applications/devices (It plays the server role). "Bluetooth GPS" [43] is an application that connects to other mobile devices asking for GPS information (It plays the client role). To perform this study both applications are needed. GPS over BT is selected since it is the highest installed Android	

	Minimum privacy: Low	text to voice application.	
	Subgroup 6 [5]: Instead of downloading a P2P file from the Internet over an expensive interface (GPRS, UMTS, LTE, etc.), a mobile user scans using Bluetooth for a nearby device which has downloaded the needed file and retrieve it over a less expensive interface (Bluetooth). The requestor's communication with other participants is multicast/broadcast by nature. Minimum privacy: Low	**Application 6 (ES File Explorer) [44]:** It is a file and multimedia manager. This application is selected since it is the highest installed Bluetooth supporting Android file manager application.	
	Subgroup 7: We propose a new application which fits under this group. Bicycle fans gather yearly to watch "tour de France", a 23-day racing event [45], taking place mainly in France and other nearby countries. Fans and racers are interested in knowing the instantaneous detailed ranking of racers with additional information (the time difference between the first racer and other contestants, velocity, etc.). Each user needs to receive the data on his device, which is transmitted to all application users. The transmitted data is similar to all the users and suits for multicast/broadcast traffic by nature. Minimum privacy: Low	**Application 7 (Tour De France 2015 by Skoda) [46]:** It provides important news on the 2015 Tour de France in real time and features a live stream of every stage on the smartphone or tablet via France Televisions. For this application [46] to be fully profiled, multicast-supporting mobile network should be used but it is outside the capabilities of our lab. Approximating its performance is the only available method. To do so, the unicast traffic needed to request/reply live information is measured and assumed to be equivalent to the multicast traffic. This application is selected since it is the only news feeding app for tour de France.	

Crowdsensing: Are used to share timestamped sensor readings, such as GPS, accelerometer, light sensor, microphone, thermometer, clock, and compass [47-66].	**Subgroup 8** [2]: Queries the users located in 1 mile radius to get the average temperature of nodes within a mile. The requestor's communication with other participants is multicast/broadcast by nature. Minimum privacy: Low	**Application 8 (Weather Signal) [67]:** It is a crowdsourcing/crowdsensing weather application which "uses native phone sensors to measure local atmospheric conditions, to be then displayed to the public" [67]. This application is selected since it is the highest installed weather crowd sensing Android application.	
	Subgroup 9 [18]: Traffic reporting can be implemented by querying the velocity distribution of all nodes within half a mile of the next highway on the current route. The requestor's communication with other participants suits for multicast/broadcast traffic by nature. Minimum privacy: Low	**Application 9:** Weather Signal, studied above, is suitable to represent both subgroups 8 and 9.	

Multimedia search [2]: Mobile devices store many types of multimedia content such as videos, photos and music which can be shared by other users.	**Subgroup 10** [18]: Multimedia files can be searched in the contents of nearby mobile devices. The query in this application is multicast/broadcast by nature. Similar applications are [68]. Minimum privacy: Low	**Application 10:** ES File Explorer [44], studied above, is suitable to represent both applications 6 and 10. For this reason, both applications will be profiled using the above results. This application is selected since it is the highest installed Bluetooth supporting Android file manager application.	

Social networking [2]: Sharing user content with friends on social media facilitates automatic sharing and P2P multimedia access, thus reducing the need for huge servers to manage this amount of data [18].	**Subgroup 11** [18]: Integrating Facebook with mobile cloud to share the active files using mobile interfaces. Minimum privacy: high.	**Application 11 (OurBook (P2P Social Network)) [69]:** It is a "Fully distributed peer-to-peer social network using onion routing" [69]. The application allows the user to view and post various types of posts. This application is selected since it is the highest installed Android P2P social networking application.	

Crowdsourcing: Is outsourcing tasks, used to be executed by machines or employees, into an external set of people [70] [71] [72].	**Subgroup 12:** "Social search engine" is one of the most popular crowdsourcing application theme which focuses on answering context-related questions using human-help (crowd) instead of or in complementary with search engines [73] [74]. Chacha [75] and Aardvark [76] are two popular crowdsourcing applications that have similar objectives, but for detailed discussion, application 12 will represent Chacha. Similar applications are [77] [78] [79]. Minimum privacy: high.	**Application 12 (Chacha) [75]:** It is a Web-based social search engine which allows the user to receive answers from the community either previously recorded or collected especially for the user's question. The user can send a question and receive the answer, respond to asked questions or support an answer. All "Social search engine" mobile applications have very low installations. In place, "Social search engine" web portal has been selected. Chacha is one of the most cited "Social search engine" web portals [73] [80] [81] [82].	
	Subgroup 13: "Crowdsourced location-based service" is a method to fetch the recommendations about certain location based categories posted by people with taste and interest similar to the requester. Foursquared [78] is a popular application offering crowdsourced location based service. Similar applications are [83] [84]. Minimum privacy: high.	**Application 13 (Foursquare) [85]:** It is a crowdsourced location-based service which offers the user the possibility to search for hotels, restaurants and other places to visit either in the user's vicinity or in a specified area. The user can look for places in his vicinity, look for places in a specified area or rate a place he visited. This application	

		is selected since it is the highest installed Android Crowdsourced location-based service application.	

Wearable computing [87] [88]: Accessing information generated (sensor) or received (downloaded) using mini-electronic devices having very limited battery, storage and computation resources.	**Subgroup 14:** A tool to measure the blood pressure, heart rate and other vital signs of the user and upload this information to be presented in real-time to the doctor. Minimum privacy: Low	**Application 14 (S Health) [86]:** It is an application that receives health recordings from mobile device's sensors, from third party applications and from other devices. The user has a consolidated view on all the recordings. This application is selected since it is the highest installed Health monitoring Android wearable computing application.	

There are other mobile cloud applications found in literature but are still under investigation, such as Mobile augmented reality [87 – 91], Cloud gaming [92 - 95], Mobile learning [92] [96 - 99], and Mobile healthcare [92] [100] [101]. These applications cannot be profiled due to the variety of proposed solutions and optimization techniques, thus will not be included in the simulation results. The surveyed architectures will be compared in the coming sections to show which architecture is suitable for each mobile cloud application.

d. Related Work

Niroshinie et al. [2] surveyed all mobile cloud physical layer architectures and categorized mobile cloud applications into 7 categories. They also surveyed all mobile cloud application layer architectures (offloading mechanisms) and their theoretical cost expressions. Although [2] offers a full coverage of mobile computing, it failed to include cost expressions for mobile cloud physical layer architectures. It also failed to include experimental results, since none was available at 2014. The proposed categories miss important applications such as games and healthcare wearable computing.

Dinh et al. [92] surveyed mobile cloud applications and presented an extensive list of issues and challenges in mobile cloud computing. Although [92] offers a comprehensive discussion on the listed challenges, it fails to present any theoretical or experimental results. Wang et al. [102] followed a very similar track.

Xu and Mao [103] studied theoretically the cost of offloading a task vs. local execution and shown the result as Application delay deadline vs. Transmitted/received data size. The authors studied also the power consumption resulting from using Wi-Fi and 3G to be used in offloading. The authors failed to add some decisive factors into their theoretical equations which lead to imprecise results. These factors include power consumption during waiting [104].

Kumar and Lu [104] also studied the theoretical cost of offloading vs. local execution. Although power consumption during waiting time has been included in the expressions, but channel conditions, and retransmission has been ignored. Both, [103] and [104], discussed security, reliability and user experience as open questions and future challenges.

El-Sayed et al. [105] proposed an offloading mechanism that will augment the capabilities of a mobile device including computation power, storage and network bandwidth. The authors demonstrated the performance of the proposed mechanism by implementing it on one application. Zhou el al. [106] followed a very similar track.

Chen [107] argued that even if offloading is beneficial on an individual level, it could become degrading if multiple uses tried to offload their tasks concurrently over the same wireless network. The author proposed a collaboration mechanism based on game theory. The performance of this mechanism is simulated and compared against local computation and offloading.

Shiraz et al. [108] analyzes the resource intensive nature of existing offloading techniques. The paper includes implementation results for 1 task namely “sorting operation”, executed for sorting lists of various length.

As can be seen in the above survey, none has studied all mobile cloud physical layer architectures against a variety of mobile applications. In this work, we categorized various mobile applications in 14 subgroups and selected the most representative application from each subgroup. The selected applications will be implemented and its performance per architecture will be shown. To the best of our knowledge, this paper is the first published work numerically comparing mobile cloud architectures across a wide variety of mobile cloud applications.

III. Evaluation Environment

In this section, we set the evaluation metrics needed to compare the architectures found in literature and show their suitability to the resource demanding mobile applications. To do so, we start by presenting the architecture requirements (metrics) which are used to evaluate the performance of surveyed mobile cloud architectures. We then select mobile applications suitable to represent the subgroups shown in Table 1. These applications will be used, in section IV, for the physical implementation and performance evaluation. Finally, the architecture configuration is presented.

a. Architecture Requirements

MCC was developed to respond to mobile devices' needs for a technology that helps in decreasing power consumption, delay and user's cost without compromising the privacy, availability and mobility offered by mobile networks. When designing mobile cloud architecture several requirements should be taken into consideration which represents the aim behind using MCC. Similarly, these requirements are used as metrics to evaluate the performance and compare the studied architectures. The requirements are divided into quantifiable and non-quantifiable as shown next:

1. Quantifiable requirements

Quantifiable requirements: can be calculated in numerical metrics. The quantifiable requirements are:

- **Cost:** Financial cost due to network usage. Cost is calculated in terms of "financial units" relative to the mobile fees. An architecture better suits the application if it achieves lower cost. This requirement has been evaluated in [109] to be Low for Cloud computing with mobile terminals, Medium for Virtual cloud computing provider and high for cloudlet although no proof has been given to support the claim.
- **Delay:** Transmission, propagation and processing delay. In case of in-house processing, it is the time between starting the execution and finishing the job. In case of offloading, it is the time between sending the first offloaded bit till receiving the last reply bit. Delay is calculated in seconds (s). An architecture better suits the application if it achieves lower delay.
- **Power consumption:** Power consumed by a mobile device during processing, transmission and waiting. It is calculated in Joules (j). An architecture better suits the application if it achieves lower power consumption.

2. Non-quantifiable requirements

Non-quantifiable requirements cannot be calculated in numerical metrics, but will be represented by subjective values. The non-quantifiable requirements are:

2.1. Privacy

This requirement focuses on the privacy of data during transmission and processing. Privacy is considered high, if no private personal data is stored or processed in non-trusted devices. It is considered medium if private personal data (such as the temperature read by a mobile's sensor) is stored or processed in non-trusted devices but cannot be correlated to the user's id, and considered low if private personal data is stored or processed in non-trusted devices and can be correlated to the user's id. An architecture better suits the application if it achieves higher privacy. Architectures get disqualified if it is not able to satisfy the privacy requirements of an application. Each architecture's privacy is studied separately:

- **Cloud computing with mobile terminals:** CSPs with a preset Service Level Agreement (SLA) are considered trusted. "Cloud computing with mobile terminals" architecture offers high privacy [109].
- **Virtual cloud computing provider:** Virtual cloud computing provider offers low privacy.
- **Cloudlet**: A cloudlet server is considered non-trusted since it is not under the supervision of a trusted provider. Cloudlet architecture offers low privacy.
- **OCMCA**: OCMCA servers are under direct supervision of a trusted entity (the operator), thus all private personal data will be performed either at the mobile device itself or within the operator's premises. OCMCA architecture offers high privacy.

2.2. Mobility

This requirement focuses on the expected distance to be covered by a user before disconnecting from a service [110]. Since mobility is one of MCC's requirements, we consider that a studied architecture better suits an application if it achieves higher mobility. Each architecture's mobility is studied separately:

- **Cloud computing with mobile terminals:** The user in "Cloud computing with mobile terminals" has the freedom to move in the entire region covered by his or any another operator (using roaming). This architecture offers high mobility.
- **Virtual cloud computing provider:** In virtual cloud, the user has more mobility freedom since not bounded to the fixed position of the Access Point (AP), but bounded to the group forming the virtual cloud he is connected. This architecture offers medium mobility. This architecture offers medium mobility.
- **Cloudlet**: cloudlet servers can be accessed through local APs only, then this very short range hinders the device's online mobility especially that we don't expect LAN coverage to be continuous. This architecture offers low mobility.

- **OCMCA**: It provides the same mobility as "Cloud computing with mobile terminals" since both architectures utilize the mechanisms of the mobile operator to handle coverage, handover, etc. This architecture offers high mobility.

2.3. Multicast-capable

This requirement is a Boolean value specifying whether the studied architecture is able to handling multicast traffic efficiently (not transforming it into bulk of unicast traffic). Since the majority of the top ranked mobile applications are multicast-based, having multicast-capable mobile cloud architecture is of critical importance.

2.4. Scalability

Mobile applications downloaded in 2013 range between 56 and 82 billion and it is expected to reach 200 billion in 2017 [31]. Mobile cloud architectures are expected to experience high penetration rates which leave the researchers in front of a critical scalability issue. The architecture should be able to serve millions of users, to handle the incremental traffic mobile networks are currently facing and most importantly to offer a satisfying coverage in cities and major villages. Each architecture's scalability is studied separately:

- **Cloud computing with mobile terminals:** This architecture offers excellent coverage and routes all the MCC jobs to be executed at CSP's premises. CSPs are able to process huge amount of jobs and extend his capability by federating some resources from other cloud providers. The only bottleneck is at transmission which might be overwhelmed with the incremental traffic if no newer mobile technology is implemented. This architecture requires no investment in additional physical devices and able to support current traffic rates. This architecture is considered to have high scalability.
- **Virtual cloud computing provider:** This architecture requires no investment in additional physical devices, but will have difficulties (delay and power drainage) in processing very complex applications since the participating mobile devices are sharing the load. A user interested in processing more complex applications

should team up with more users, users with powerful mobile devices and/or upgrade his devices. This architecture is considered to have high scalability.

- **Cloudlet:** This architecture requires the investment of random business owners in deploying physical devices (cloudlet servers) even with the absence of a valid business model. Cloudlets will face low utilization rates, over-investment and high rejection rates due to the traffic commuting. As residential areas are usually apart from business areas, the cloudlet servers in one area will be severely under-utilized throughout the period the users have commuted to other locations. This requires over-investment to offer acceptable services in different areas. If some areas are not well equipped, customer rejection rates will be elevated. This architecture is considered to have low scalability.
- **OCMCA**: It provides the same scalability as "Cloud computing with mobile terminals" since both architectures utilize the mechanisms of the mobile operator. This architecture is highly scalable.

The above discussion is summarized in Table 2.

Table 2 Consolidated results of MCC architectures' performance against the non-quantifiable requirements

	Non-quantifiable Requirements (Higher values are better)			
Architecture	**Privacy**	**Mobility**	**Scalability**	**Multicast Capable?**
In-house Processing (offline device, without using Mobile Cloud Computing)	High	High	Very Low	No
Cloud computing with mobile terminals	High	High	High	No
Virtual cloud computing provider	Medium	Medium	High	No
Cloudlet	Low	Low	Low	No
OCMCA	High	High	High	Yes

b. Architecture environment

The implementation of the surveyed mobile cloud architectures is based on the technologies available in Lebanon. The values are kept to make the results as realistic as possible especially for Middle East countries. These technologies will be presented when needed later in this subsection. The performance of MCC can only be compared with in-house processing (i.e. not using MCC). For this reason, in-house processing and existing MCC architectures are both used to evaluate their suitability to various resource-intensive mobile applications. The implementation of such architectures is as follows:

1. **Basic architecture (In-house Processing)**

In-house processing requires the mobile device to perform all operations in offline mode. For this reason, applications 1, 2 and 6 are the only applications that could be implemented in this architecture. Jobs are executed using two different platforms which are:

- Platform 1: SAMSUNG Note 5 (SAMSUNG's current high-end mobile device which has the following specs: Quad core 2.1 GHz Cortex-A57, 4GB RAM, Android OS v5.1.1)
- Platform 2: SAMSUNG GRAND (SAMSUNG mobile device which has the following specs: Dual-core 1.2 GHz Cortex-A9, 1GB RAM, Android OS v4.1.2)

The processing times of these two platforms are considered to be the upper and lower limits; even though slower devices still exist in the market. The power consumption is recorded using platform 1 which will be used for all other architectures.

2. **Cloud computing with mobile terminals**

"Cloud computing with mobile terminals" requires the jobs to be offloaded to a remote server over a mobile network. All the measurements performed for this architecture use LTE connection at Deir el Qamar, Lebanon. Tests are performed between 10 a.m. and 2 p.m. to ensure result coherence. Internet delay is expected due to high propagation delays since most of the used application servers are found in USA. The used device is identical to platform 1 presented in III.b.1. The used cloud sever is virtual machine with the following specs (4GB RAM, 4 threads allocated from an "Intel Core i7-4770 3.4 GHz" processor, automatics virtualization and Android for x86 v4.4 release 5).

3. Virtual cloud computing provider

Virtual cloud computing provider requires the mobile device to offload part of its operations to interested neighboring devices. In this scenario, three SAMSUNG Note 5 devices (platform 1) are used and connected through an ad-hoc wireless network. This network is constituted of a job requestor and 2 responders. The number of devices could be any integer greater than 2, but 3 has been chosen for practicality.

4. Cloudlet

In this architecture, each application will upload its request to a local server over Wi-Fi AP. The used sever is (4GB RAM, 2 threads reserved from a "Intel Core i5-3230M 2.6 GHz" processor, enabled acceleration and Android for x86 v4.4 release 5). The test considers that the mobile-server session is the only active session over the used Wi-Fi AP.

5. OCMCA

In this architecture, each application will upload its request to a cloud server at the CSP’s premises. Since this architecture is not yet standardized and implemented at our local CSPs, a server located at the CSP’s ISP (called ISP’s server) is used as shown in figure 8.

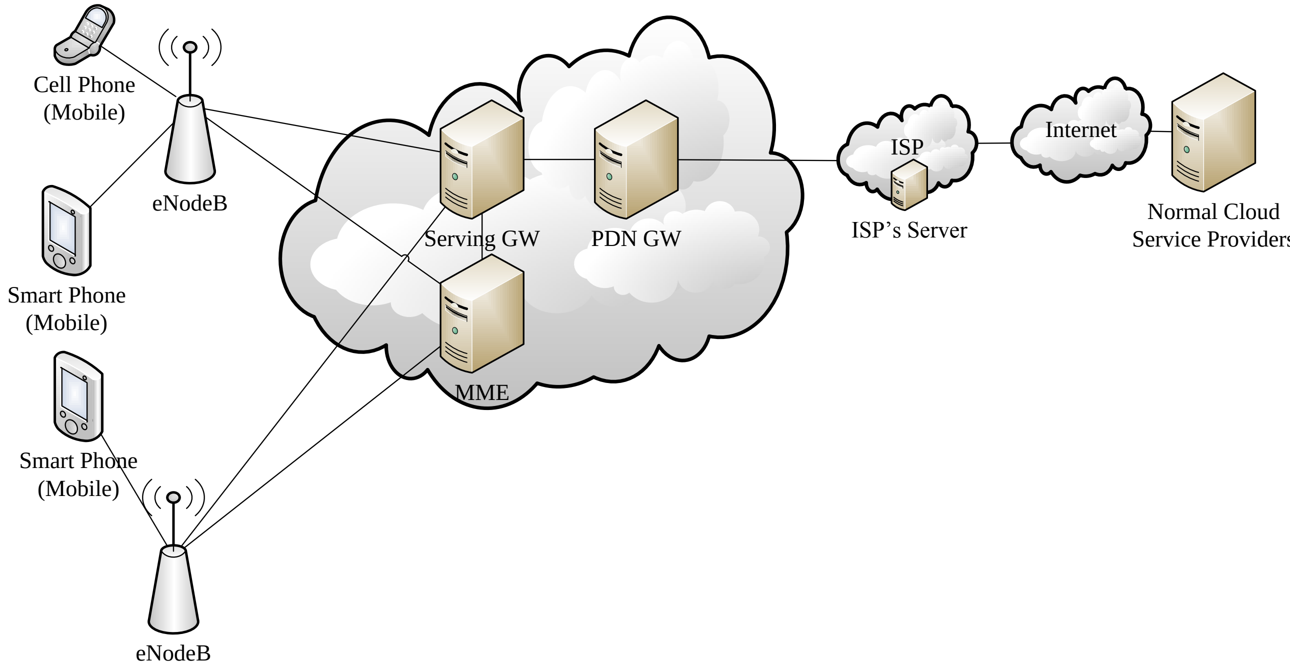

Figure 8 OCMCA Implementation

The used sever has the same specs as those in "Cloud computing with mobile terminals" (4GB RAM, 4 threads allocated from a "Intel Core i7-4770 3.4 GHz" processor, automatics virtualization and Android for x86 v4.4 release 5). All the measurements performed for this architecture use LTE connection at Deir el Qamar, Lebanon.

IV. Physical Implementation

In this section, we present the physical implementation of the applications shown in Table 1 across different mobile cloud architectures, where cost, delay and power consumption are measured. The implementation results are not generalized but considered as an evaluation of sample instances. Power consumption is measured using an android application called Powertutor [111]. Implementations are categorized per application as shown next:

a. Application 1 (OCR Instantly Free)

"OCR Instantly Free" is an application that performs optical character recognition "which is a technology to convert image to text" [32]. It is available for major smartphone vendors running android or iOS. OCR’s results across different mobile cloud architectures are shown in Table 3.

Table 3 OCR’s results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network (10^3Bytes)	Downloaded Data over Mobile network (10^3Bytes)	Total Delay (sec)	Power Consumption (J)
In-house processing	0	0	6.29	12.93
Cloud computing with mobile terminals	5762.2	0.7	121.6	62.99
Virtual cloud computing provider	1920.73	0.23	6.19	5.40
Cloudlet	0	0	8.24	0.99
OCMCA	5762.2	0.7	54.24	29.78

It can be seen in Table 3 that "in-house processing", "virtual cloud computing with mobile terminals" and "cloudlet" have the best cost since no traffic is sent over the expensive mobile network. It can be also seen that "virtual cloud computing with mobile terminals" has the best delay while "in-house processing" fall shortly behind and "cloudlet" relatively close. As for power consumption, "cloudlet" has the best result followed by "virtual cloud computing with mobile terminals" and "in-house processing". Irrelevant of the metric priority, "cloudlet" and "virtual cloud computing with mobile terminals" are considered to have very good results in addition to "in-house processing". We would like to note that "platform 1" is the current high-end device (highest specs found on the market) and if replaced by other devices (with lower specs) "in-house processing" and "virtual cloud computing with mobile terminals" will lag behind "cloudlet". "OCMCA" and "Cloud computing with mobile terminals" fail to compete with other architectures on all metrics. The main reason for this poor performance is the slow transmission rate achieved on upload and download using LTE. This is due to the fact that the tests are performed at non-peak hours, but relatively high traffic hours. The results would be different if the same tests were performed at low traffic hours, but this will not evaluate the real scenario.

b. Application 2 (Talk - Text to Voice FREE)

Talk is an application that performs text-to-speech operations. It has been studied for 6 phrases each with different criteria (Note: all phases are in English). Talk's results across different mobile cloud architectures are shown in Table 4.

Table 4 Talk's results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network	Downloaded Data over Mobile network	Total Delay (sec)	Power Consumption (J)
In-house processing	0	0	1.12	1.47
Cloud computing with mobile terminals	0.46	647	0.9	0.22
Virtual cloud computing provider	0	0	1.32	1.32
Cloudlet	0	0	1.24	0.12
OCMCA	0.46	647	0.83	0.08

As can be seen in Table 4, "OCMCA" has the best delay and power consumption, but the worst cost. "OCMCA" 's cost over local network is negligible but might be considered expensive over roaming networks especially with expensive operators such as those in Middle East.

c. Application 3 (Crowd bytes)

"Crowd bytes" is a crowd sourcing application that allows, among other things, to publish the profiles of found or lost animals (Living) or objects (Non- Living). This application has been studied for 4 profiles. Crowd Byte's results are shown in Table 5.

As can be seen in Table 5, "OCMCA" has achieved the best power consumption, and the second best delay following "cloudlet". "Cloudlet" has the worst power consumption. Architecture selection in this case requires extra attention on metric weighting.

Table 5 "Crowd Byte" 's results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network	Downloaded Data over Mobile network	Total Delay (sec)	Power Consumption (J)
In-house processing	X	X	X	X
Cloud computing with mobile terminals	202.6	0.1	5.25	3.03
Virtual cloud computing provider	67.53	0.03	1.80	2.84
Cloudlet	0	0	0.1	4.18
OCMCA	202.6	0.1	1.78	0.8725

d. Application 4 (Brizo)

"Brizo" is a crowd sourcing application that is "used to monitor road conditions like flood, wild fire, accident and traffic jam." [39]. This application has been studied for 5 operations. Brizo's results across different mobile cloud architectures are shown in Table 6.

Table 6 Brizo's results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network	Downloaded Data over Mobile network	Total Delay (sec)	Power Consumption (J)
In-house processing	X	X	X	X
Cloud computing with mobile terminals	6.29	51.45	10.87	0.89
Virtual cloud computing provider	6.29	51.45	12.16	1.32
Cloudlet	0	0	12.81	1.54
OCMCA	18.86	154.35	3.70	0.30

As can be seen in Table 6, "OCMCA" has achieved the best power consumption and delay followed by "Cloud computing with mobile terminals". As for the cost, "Cloudlet" achieved the best score since it is the only architecture using Wi-Fi traffic solely.

e. Application 5 (GPS over BT)

"GPS over BT" [42] is an application that reads GPS information and offers it to other applications/devices (It plays the server role). "Bluetooth GPS" [43] is an application that connects to other mobile devices asking for GPS information (It plays the client role). To perform this study both applications are needed. This application has been studied for 2 measurement types. GPS over BT's results across different mobile cloud architectures are shown in Table 7.

Table 7 "GPS over BT" ’s results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network (10^3Bytes)	Downloaded Data over Mobile network (10^3Bytes)	Total Delay (sec)	Power Consumption (J)
In-house processing	X	X	X	X
Cloud computing with mobile terminals	0.02	0.02	22.43	0.55
Virtual cloud computing provider	0.02	0.02	10.06	0.36
Cloudlet	0	0	22.62	0.36
OCMCA	0.02	0.02	22.43	0.55

As can be seen in Table 7, "Virtual cloud computing provider" has achieved the best delay and sharing the best power consumption score with "Cloudlet". As for the cost, "Cloudlet" achieved the best score.

f. Application 6 (ES File Explorer)

ES File Explorer "is a full-featured file manager for both local and networked use" [44]. When used in network mode, device 1 accesses the filesystem of device 2 where it can upload, download or delete files. We have studied the application for 5 different files with "in-house processing". In this scenario, two "platform 1" devices are used where device 1 (job requestor) connects to device 2 and retrieves a file over Bluetooth. Our measurements focus on the job requestor. "ES File Explorer" ’s results across different mobile cloud architectures are shown in Table 8.

Table 8 "ES File Explorer" ’s results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network (10^3Bytes)	Downloaded Data over Mobile network (10^3Bytes)	Total Delay (sec)	Power Consumption (J)
In-house processing	0	0	113.4	17.48
Cloud computing with mobile terminals	21,490.62	1,684.63	418	14.10
Virtual cloud computing provider	0	0	4.56	0.34
Cloudlet	0	0	4.98	0.47
OCMCA	21,490.62	1,684.63	202.42	8.07

As can be seen in Table 8, "Virtual cloud computing provider" best suits this application across all measured metrics and followed directly by "Cloudlet".

g. Application 7 (Tour De France 2015 by Skoda)

For this application [46] to be fully profiled, multicast-supporting mobile network should be used but it is outside the capabilities of our lab. Approximating its performance is the only available method. To do so, the unicast traffic needed to request/reply live information is measured and assumed to be equivalent to the multicast traffic. "Tour De France 2015 by Skoda" ’s results across different mobile cloud architectures are shown in Table 9.

Table 9 "Tour De France 2015 by Skoda" ’s results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network (10^3Bytes)	Downloaded Data over Mobile network (10^3Bytes)	Total Delay (sec)	Power Consumption (J)
In-house processing	X	X	X	X
Cloud computing with mobile terminals	4.33	61.72	2.84	0.34
Virtual cloud computing provider	1.44	20.57	2.61	0.45
Cloudlet	0	0	2.49	0.5
OCMCA	4.33	61.72	1.68	0.26

As can be seen in Table 9, "OCMCA" has achieved the best power consumption and delay. As for the cost, "Cloudlet" achieved the best score.

h. Application 8 (Weather Signal)

Weather Signal [67] is a crowdsourcing/crowdsensing weather application which “uses native phone sensors to measure local atmospheric conditions, to be then displayed to the public” [67]. "Weather Signal" ’s results across different mobile cloud architectures are shown in Table 10.

Table 10 "Weather Signal" ’s results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network (10^3Bytes)	Downloaded Data over Mobile network (10^3Bytes)	Total Delay (sec)	Power Consumption (J)
In-house processing	X	X	X	X
Cloud computing with mobile terminals	10.83	247.25	5.29	2.33
Virtual cloud computing provider	3.61	82.42	2.68	2.04
Cloudlet	0	0	1.37	1.89
OCMCA	10.83	247.25	2.67	2.22

As can be seen in Table 10, "cloudlet" has achieved the best power consumption, delay and cost. "OCMCA" and "Virtual cloud computing provider" come in second and third place while "Cloud computing with mobile terminals" comes last.

i. Application 11 (OurBook (P2P Social Network))

OurBook (P2P Social Network) [69] is a "Fully distributed peer-to-peer social network using onion routing" [69]. The application allows the user to view and post various types of posts. OurBook (P2P Social Network)'s results across different mobile cloud architectures are shown in Table 11.

Table 11 "OurBook (P2P Social Network)" 's results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network (10^3Bytes)	Downloaded Data over Mobile network (10^3Bytes)	Total Delay (sec)	Power Consumption (J)
In-house processing	X	X	X	X
Cloud computing with mobile terminals	41.40	44.7	1.02	0.96
Virtual cloud computing provider	13.80	14.90	1.15	1.49
Cloudlet	0	0	1.22	1.75
OCMCA	41.40	44.7	0.51	0.69

As can be seen in Table 11, "OCMCA" has achieved the best power consumption and delay followed by "Cloud computing with mobile terminals". As for the cost, "Cloudlet" achieved the best score.

j. Application 12 (Chacha)

Chacha [69] is a Web-based social search engine which allows the user to receive answers from the community either previously recorded or collected especially for the user's question. The user can send a question and receive the answer, respond to asked questions or support an answer. "Chacha" 's results across different mobile cloud architectures are shown in Table 12.

Table 12 "Chacha" 's results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network	Downloaded Data over Mobile network	Total Delay	Power Consumption
In-house processing	X	X	X	X
Cloud computing with mobile terminals	0.65	5,419	11.47	0.56
Virtual cloud computing provider	0.14	1,204.22	20.37	6.71
Cloudlet	0	0	24.83	2.17
OCMCA	0.65	5,419	6.79	0.43

As can be seen in Table 12, "OCMCA" has achieved the best power consumption and delay followed by "Cloud computing with mobile terminals". As for the cost, "Cloudlet" achieved the best score.

k. Application 13 (Foursquare)

Foursquare [85] is a crowdsourced location-based service which offers the user the possibility to search for hotels, restaurants and other places to visit, either in the user's vicinity or in a specified area. This application has been studied for 7 operations. Foursquare's results across different mobile cloud architectures are shown in Table 13.

Table 13 "Foursquare" 's results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network (10^3Bytes)	Downloaded Data over Mobile network (10^3Bytes)	Total Delay (sec)	Power Consumption (J)
In-house processing	X	X	X	X
Cloud computing with mobile terminals	1,113.83	6,885.84	23.26	12.16
Virtual cloud computing provider	371.28	2295.28	21.02	6.55
Cloudlet	0	0	31.54	3.75
OCMCA	1,113.83	6,885.84	9.85	6.65

As can be seen in Table 13, "Cloudlet" has achieved the best power consumption and cost; while "OCMCA" has achieved the best delay.

l. Application 14 (S Health)

"S Health" [86] is an application that receives health recordings from mobile device's sensors, from third party applications and from other devices. The user has a consolidated view on all the recordings. The traffic generated from receiving recordings from external devices and synchronizing. "S Health" 's results across different mobile cloud architectures are shown in Table 14.

Table 14 "S Health" 's results across different mobile cloud architectures

Architecture	Uploaded Data over Mobile network (10^3Bytes)	Downloaded Data over Mobile network (10^3Bytes)	Total Delay	Power Consumption
In-house processing	X	X	X	X
Cloud computing with mobile terminals	1,720	420	8.3	1.62
Virtual cloud computing provider	573.33	140	11.16	8.95
Cloudlet	0	0	12.5	12.62
OCMCA	1,720	420	5.07	1.01

As can be seen in Table 14, "OCMCA" has achieved the best power consumption, and delay followed by "Cloud computing with mobile terminals". As for the cost, "Cloudlet" achieved the best score.

V. Conclusion

"Cloudlet" is the best suitable architecture from cost perspective; since, for all applications, no traffic is transmitted over the expensive mobile network. Things become more complex when evaluating delay and power consumption. We will start by profiling and categorizing applications into bandwidth demanding and non-demanding. We profile the applications based on the following criteria:

- (Sum of the upload and download traffic) > 10MB: Very Highly Demanding
- 10MB > (Sum of the upload and download traffic) > 1MB: Highly Demanding
- 1MB > (Sum of the upload and download traffic) > 200KB: Moderately Demanding
- 200KB > (Sum of the upload and download traffic) > 10KB: Non Demanding
- 10KB > (Sum of the upload and download traffic): Highly Non Demanding

We will also categorize the applications into either requiring extensive processing or not. The categorized applications are shown in Table 15 sorted by bandwidth demand.

Table 15 Bandwidth demanding and non-demanding applications

Application	Upload Data (10^3Bytes)	Download Data (10^3Bytes)	Total Data Size (10^3Bytes)	Bandwidth Demanding?	Extensive Processing?	Delay suitable architecture	Power consumption suitable architecture
ES File Explorer	21,490.62	1,684.63	23,175.25	Very Highly Demanding	No	3	3
OCR Instantly Free	5,762.20	0.70	5,762.90	Highly Demanding	Yes	3	4
Chacha	0.65	5,419.00	5,419.65		No	5	5
Foursquare	1,113.83	6,885.84	7,999.67		No	5	4
S Health	1,720.00	420.02	2,140.02		No	5	5
Talk – Text to Voice	0.46	647.00	647.46	Moderately Demanding	Yes	5	5
Crowd bytes	202.60	0.10	202.70		No	4	5
Weather Signal	10.83	247.25	258.08		No	4	4
Brizo	18.86	154.35	173.21	Non-Demanding	No	5	5
Tour De France 2015 by Skoda	4.33	61.72	66.05		No	5	5
OurBook	41.40	44.70	86.10		No	5	5
GPS over BT	0.38	0.38	0.76	Highly Non-Demanding	No	3	3 & 4

Architectures are represented by their numbers as presented in section III.c. The architectures are as follows:

1. Architecture 1 means "In-house processing".
2. Architecture 2 means "Cloud computing with mobile terminals".
3. Architecture 3 means “Virtual cloud computing provider”.
4. Architecture 4 means "Cloudlet".
5. Architecture 5 means "OCMCA".

Based on the results shown in Table 15, we can select the best architecture suiting a category from delay perspective. We will start first by the applications without extensive processing and the results are as follows:

- "Virtual cloud computing provider" is the best suitable architecture for "Very highly demanding" applications.
- "OCMCA" is the best suitable architecture for "Highly demanding" applications.
- "Cloudlet" is the best suitable architecture for "Moderately demanding" applications.
- "OCMCA" is the best suitable architecture for “Non-demanding” applications.
- "Virtual cloud computing provider" is the best suitable architecture for "Very non-demanding" applications.

The results achieved for "Highly Demanding", "Moderately Demanding" and "Non-Demanding" applications can be generalized for other applications belonging to these categories since all the measurements support these findings. The results are also presented in Figure 9, but one application was eliminated (ES File Explorer because it has huge delay compared to other applications) thus affecting the readability of the figure.

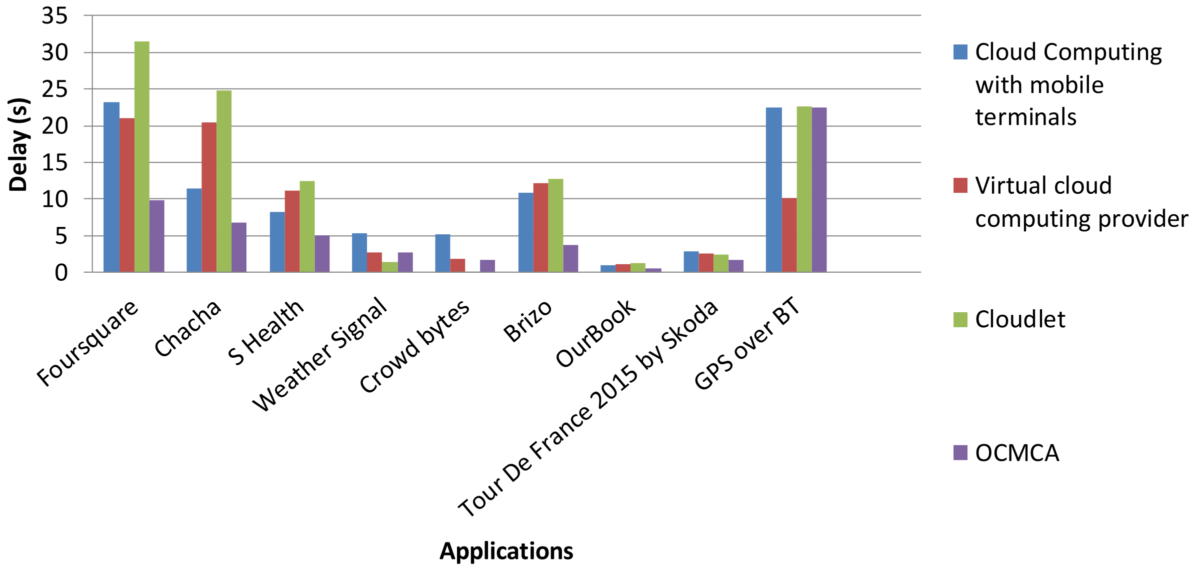

Figure 9 Delay results for non-processing extensive applications

For applications requiring extensive processing, they have the following suitable architectures from delay perspective:

- "Virtual cloud computing provider" is the best suitable architecture for "Highly demanding" applications.
- "OCMCA" is the best suitable architecture for "Moderately demanding" applications.

From power consumption perspective, the best suiting architectures are as follows:

- "Virtual cloud computing provider" is the best suitable architecture for "Very highly demanding" applications.
- "OCMCA" or "Cloudlet" is the best suitable architectures for "Highly demanding" and "Moderately demanding" applications.
- "OCMCA" is the best suitable architecture for "Non-demanding" applications.
- "Virtual cloud computing provider" is the best suitable architecture for "Very non-demanding" applications.

In this paper, mobile cloud architectures have been surveyed to cover all the physical-layer architectures found in literature. The architectures were compared based on their performance in servicing a wide range of mobile cloud applications. The comparison has been performed using both simulation and physical implementation. It has been shown that no mobile cloud architecture suits all applications, but network designers can decide their best suiting architecture based on the expected applications to be used.

References

[1] Chen, Shanzhi, and Jian Zhao. "The requirements, challenges, and technologies for 5G of terrestrial mobile telecommunication." Communications Magazine, IEEE 52, no. 5 (2014): 36-43.

[2] F. Niroshinie, S. W. Loke & W. Rahayu, "Mobile cloud computing: A survey", Future Generation Computer Systems, vol. 29 no. 1, pp. 84-106, 2013.

[3] J. Bou Abdo, J. Demerjian, H. Chaouchi, K. Barbar & G. Pujolle, "Single-Sign-on in Operator Centric Mobile Cloud Architecture", 17th IEEE Mediterranean Electrotechnical Conference (IEEE Melecon 2014).

[4] B. G. Chun, S. Ihm, P. Maniatis, M. Naik & A. Patti, "Clonecloud: elastic execution between mobile device and cloud", 6th ACM conference on Computer systems, pp. 301-313, 2011.

[5] G. Huerta-Canepa & D. Lee, "A virtual cloud computing provider for mobile devices", 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, 2010.

[6] M. Satyanarayanan, P. Bahl, R. Caceres & N. Davies, "The case for VM-based cloudlets in mobile computing", IEEE Pervasive Computing, vol. 8 no. 4, 2009.

[7] J. Bou Abdo, J. Demerjian, H. Chaouchi, K. Barbar, & G. Pujolle, "Operator Centric Mobile Cloud Architecture", IEEE Wireless Communications and Networking Conference, pp. 3024-3029, 2014.

[8] Making the Most of Google Cloud Messaging, Link: https://stuff.mit.edu/afs/sipb/project/android/docs/training/cloudsync/gcm.html (Access Date: 30- Dec - 2015).

[9] J. Liu, E. Ahmed, M. Shiraz, A. Gani, R. Buyya & A. Qureshi, "Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions." Journal of Network and Computer Applications, vol. 48, pp. 99-117, 2015.

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

[11] B. G. Chun, S. Ihm, P. Maniatis & M. Naik, "Clonecloud: boosting mobile device applications through cloud clone execution", arXiv preprint arXiv: 1009.3088. 2011.

[12] E. Hyytiä, T. Spyropoulos & J. Ott, "Optimizing offloading strategies in mobile cloud computing", pp. 1-9, 2013.

[13] R. Kemp, N. Palmer, T. Kielmann & H. Bal, "Cuckoo: a computation offloading framework for smartphones", Mobile Computing, Applications, and Services, Springer Berlin Heidelberg, pp. 59-79, 2012.

[14] M. N. El-Derini, H. H. Aly, A. E. H. G. El-Barbary & L. A. El-Sayed, "DroidCloudlet: Towards cloudlet-based computing using mobile devices", 5th IEEE International Conference on Information and Communication Systems, pp. 1-6, 2014.

[15] T. Verbelen, P. Simoens, F. De Turck & B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user", 3rd ACM workshop on Mobile cloud computing and services, pp.29-36, 2012.

[16] J. Flinn, S. Park & M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing", 22nd IEEE International Conference on Distributed Computing Systems, pp. 217–226, 2002.

[17] R. Balan, M. Satyanarayanan, S. Park & T. Okoshi, "Tactics-based remote execution for mobile computing", 1st ACM International Conference on Mobile Systems, Applications and Services, pp. 273–286, 2003.

[18] E.E. Marinelli, "Hyrax: cloud computing on mobile devices using MapReduce", Master Thesis, Carnegie Mellon University, 2009.

[19] D.C. Doolan, S. Tabirca & L.T. Yang, "Mmpi a message passing interface for the mobile environment", 6th ACM International Conference on Advances in Mobile Computing and Multimedia, pp. 317–321, 2008.

[20] D. Huang, X. Zhang, M. Kang & J. Luo, "Mobicloud: building secure cloud framework for mobile computing and communication", 5th IEEE International Symposium on Service Oriented System Engineering, pp. 27–34, 2010.

[21] E. Cuervo, A. Balasubramanian, D. K. Cho, A. Wolman, S. Saroiu, R. Chandra & P. Bahl, "MAUI: making smartphones last longer with code offload", 8th ACM International conference on Mobile systems, applications, and services, pp.49-62, 2010.

[22] Openmobster, Website link: https://code.google.com/p/openmobster/, (Access Date: 29/12/2015)

[23] S. Chetan, G. Kumar, K. Dinesh, K. Mathew, & M. A. Abhimanyu, "Cloud computing for mobile world", A literature survey on the current status of Mobile Cloud Computing (MCC) and analysed it's shortcomings, Available at: http://chetan.ueuo.com/projects/CCMW.pdf, 2010.

[24] A. Khan, M. Othman, S. Ahmad Madani & S. U. Khan. "A survey of mobile cloud computing application models", IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 393-413, 2014.

[25] S. Zohreh, S. Abolfazli, A. Gani & R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges", Communications Surveys & Tutorials, IEEE 16, no. 1 pp. 369-392, 2014.

[26] A. Ejaz, A. Gani, M.Sookhak, S. Hafizah Ab Hamid & Feng Xia, "Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges", Journal of Network and Computer Applications, vol. 52, pp. 52-68, 2015.

[27] M. Reza Rahimi, J. Ren, C. Harold Liu, A. V. Vasilakos, & N. Venkatasubramanian. "Mobile cloud computing: A survey, state of art and future directions", Mobile Networks and Applications 19, no. 2, pp.133-143, 2014.

[28] V. March, Y. Gu, E. Leonardi, G. Goh, M. Kirchberg, and B. S. Lee, “µcloud: towards a new paradigm of rich mobile applications,” Procedia Computer Science, vol. 5, pp. 618–624, 2011.

[29] R. K. Ma, K. T. Lam, and C.-L. Wang, “excloud: Transparent runtime support for scaling mobile applications in cloud,” in Cloud and Service Computing (CSC), 2011 International Conference on. IEEE, 2011, pp. 103–110.

[30] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “Unleashing the power of mobile cloud computing using thinkair,” arXiv preprint arXiv:1105.3232, 2011.

[31] Global mobile statistics 2014, Link: http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats (Access Date: 01- July - 2014).

[32] OCR Instantly Free, Website link: https://play.google.com/store/apps/details?id=com.thesimplest.ocr&hl=en, (Access Date: 29/12/2015)

[33] Talk - Text to Voice FREE, Website link: https://play.google.com/store/apps/details?id=com.ktix007.talk&hl=en, (Access Date: 29/12/2015)

[34] J. Cheng, R.K. Balan & M. Satyanarayanan, "Exploiting rich mobile environments", Technical Report, 2005.

[35] M. Satyanarayanan, "Mobile computing: the next decade", 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, 2010.

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

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

[38] Crowd Bytes, Website link: https://play.google.com/store/apps/details?id=com.crowdbytes.social, (Access Date: 29/12/2015)

[39] Brizo, Website link: https://play.google.com/store/apps/details?id=cyberknights.info.Brizo, (Access Date: 29/12/2015)

[40] P. Angin, B. Bhargava & S. Helal, "A mobile-cloud collaborative traffic lights detector for blind navigation", 9th IEEE International Conference on Mobile Data Management, pp. 396-401, 2010.

[41] Z. Niu, S. Li, & N. Pousaeid, "Road extraction using smart phones GPS", 2nd ACM International Conference on Computing for Geospatial Research & Applications, 2011.

[42] GPS over BT, Website link: https://play.google.com/store/apps/details?id=com.Saenko.GpsOverBt, (Access Date: 29/12/2015)

[43] Bluetooth GPS, Website link: https://play.google.com/store/apps/details?id=googoo.android.btgps, (Access Date: 29/12/2015)

[44] ES File Explorer, Website link: https://play.google.com/store/apps/details?id=com.estrongs.android.pop, (Access Date: 29/12/2015)

[45] Tour de France, Website link: http://www.letour.fr (Access Date: 01 - July - 2013).

[46] Tour De France 2015 by Skoda, Website link: https://play.google.com/store/apps/details?id=com.aso.tdf2012int, (Access Date: 29/12/2015)

[47] P. Zhou, Y. Zheng & M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing", 10th ACM International conference on Mobile systems, applications, and services, 2012.

[48] E. Koukoumidis, L.-S. Peh & M. R. Martonosi, "SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory", 9th International conference on Mobile systems, applications, and services, 2011.

[49] R. K. Balan, K. X. Nguyen & L. Jiang, "Real-time trip information service for a large taxi fleet", 9th International conference on Mobile systems, applications, and services, pp. 99-112, 2011.

[50] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser & W. Trappe, "Parknet: drive-by sensing of road-side parking statistics", 8th International conference on Mobile systems, applications, and services, pp. 123-136, 2010.

[51] X. Bao & R. Choudhury, "Movi: mobile phone based video highlights via collaborative sensing", 8th International conference on Mobile systems, applications, and services, pp. 357-370, 2010.

[52] M. Wirz, C. Strohrmann, R. Patscheider, F. Hilti, B. Gahr, F. Hess, D. Roggen & G. Troster, "Real-time detection and recommendation of thermal spots by sensing collective behaviors in paragliding", 1st ACM international symposium on From digital footprints to social and community intelligence, pp. 7-12, 2011.

[53] H. Lu, W. Pan, N. D. Lane, T. Choudhury & A. T. Campbell, "SoundSense: scalable sound sensing for people-centric applications on mobile phones", 7th ACM International conference on Mobile systems, applications, and services, pp. 165-178, 2009.

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

[55] H. Cheng, F. Sun, S. Buthpitiya & M. Griss, "Sensorchestra: Collaborative sensing for symbolic location recognition", Book Title: Mobile Computing, Applications, and Services, Print ISBN: 978-3-642-29335-1, pp. 195–210, Publisher: Springer Berlin Heidelberg, 2012.

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

[57] Sensorly, Website link: http://www.sensorly.com/ (Access Date: 30 - Dec - 2015).

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

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

[60] T. Hunter, T. Moldovan, M. Zaharia, S. Merzgui, J. Ma, M. Franklin, P. Abbeel & A. Bayen, "Scaling the mobile millennium system in the cloud", 2nd ACM Symposium on Cloud Computing, p. 28, 2011.

[61] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West & P. Boda, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research", 7th ACM International conference on Mobile systems, applications, and services, pp. 55–68, 2009.

[62] O. Khalid, M. Khan, S. Khan & A. Zomaya, "OmniSuggest: A Ubiquitous Cloud based Context Aware Recommendation System for Mobile Social Networks", IEEE Transactions on Services Computing, pp. 401 – 414, 2014.

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

[64] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng & A. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceMe application", 6th ACM conference on Embedded network sensor systems, pp. 337–350, 2008.

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

[66] H. Wu, Q.Wang & K.Wolter, "Mobile Healthcare Systems with Multi cloud Offloading", 14th IEEE International Conference on Mobile Data Management, pp. 188-193, 2013.

[67] Weather Signal, Website link: https://play.google.com/store/apps/details?id=com.opensignal.weathersignal, (Access Date: 29/12/2015)

[68] J. Joy, N. Akula Suresh Babu, C. Kuo, H. Kapadia & M. Gerla, "Linearly scalable crowdsourced media broadcasting in the mobile cloud", ACM workshop on Student workshop, pp. 21-22, 2013.

[69] OurBook (P2P Social Network) , Website link: https://play.google.com/store/apps/details?id=onion.network, (Access Date: 29/12/2015)

[70] B. L. Bayus, "Crowdsourcing new product ideas over time: an analysis of the Dell IdeaStorm community" Management Science Journal, vol. 59 no. 1, pp. 226-244. 2013.

[71] J. Howe, "Crowdsourcing: How the power of the crowd is driving the future of business", Book, Random House, 2008.

[72] J. Zimmerman, A. Tomasic, C. Garrod, D. Yoo, C. Hiruncharoenvate, R. Aziz, N. R. Thiruvengadam, Y. Huang & A. Steinfeld, "Field trial of tiramisu: crowd-sourcing bus arrival times to spur co-design", SIGCHI Conference on Human Factors in Computing Systems, pp. 1677-1686, 2011.

[73] M. F. Bulut, Y. S. Yilmaz & M. Demirbas, "Crowdsourcing location-based queries", IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 513-518, 2011.

[74] D. Horowitz & S. D. Kamvar, "The anatomy of a large-scale social search engine", 19th ACM International conference on World Wide Web, pp. 431-440, 2010.

[75] Chacha, Website link: http://www.chacha.com/ (Access Date: 01 - July - 2013).

[76] Aardvark, Website link: http://www.Aardvark.com/ (Access Date: 01 - July - 2013).

[77] A. Bozzon, M. Brambilla & S. Ceri, "Answering Search Queries with CrowdSearcher", 21st International conference on World Wide Web, pp. 1009-1018, 2012.

[78] Foursquare Website, Link: https://foursquare.com/ (Access Date: 01 - July - 2013).

[79] W. S. Lasecki, R. Wesley, J. Nichols, A. Kulkarni, J. F. Allen & J. P. Bigham, "Chorus: A Crowd-Powered Conversational Assistant", 26th annual ACM symposium on User interface software and technology, pp. 151-162, 2013.

[80] W. Gross, T. McGovern & S. Colwell, "Personalized search engine", U.S. Patent Application 11/130,629, filed May 16, 2005.

[81] E. Chi, "Information seeking can be social." Information Seeking Support Systems (2008).

[82] M. R. Morris, "Collaborative search revisited." 2013 conference on Computer supported cooperative work, pp. 1181-1192. ACM. 2013.

[83] K. Tamai & A. Shinagawa, "Platform for location-based services", Fujitsu Sci. Tech. J, vol. 47, no. 4, pp. 426 – 433, 2011.

[84] H. La & S. Kim, "A conceptual framework for provisioning context aware mobile cloud services", 3rd IEEE International Conference on Cloud Computing, pp. 466–473, 2010.

[85] Fourquare, Website link: https://play.google.com/store/apps/details?id=com.joelapenna.foursquared (Access Date: 29/12/2015)

[86] S Health , Website link: https://play.google.com/store/apps/details?id=com.sec.android.app.shealth, (Access Date: 29/12/2015)

[87] T. Höllerer & S. Feiner, "Mobile augmented reality", Telegeoinformatics: Location Based Computing and Services. Taylor and Francis Books Ltd., London, UK, 2004.

[88] B. S. P. Lin, W. H. Tsai, C. C. Wu, P. H. Hsu, J. Y. Huang & T. H. Liu, "The Design of Cloud Based 4G/LTE for Mobile Augmented Reality with Smart Mobile Devices", 7th IEEE International Symposium on Service Oriented System Engineering, pp. 561-566, 2013.

[89] Google glass, Website link: http://www.google.com/glass/start/ (Access date: 23/1/2014).

[90] K. Kangas & J. Roning, "Using code mobility to create ubiquitous and active augmented reality in mobile computing", ACM/IEEE International conference on Mobile computing and networking, pp. 48–58, 1999.

[91] X. Luo, "From augmented reality to augmented computing: A look at cloud-mobile convergence", International Symposium on Ubiquitous Virtual Reality, pp. 29–32, 2009.

[92] H. T. Dinh, C. Lee, D. Niyato & P. Wang, "A survey of mobile cloud computing: architecture, applications and approaches", Wireless communications and mobile computing, vol. 13 no.18, pp. 1587-1611, 2013.

[93] S. Shi, C. H. Hsu, K. Nahrstedt & R. Campbell, "Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming", 19th ACM International conference on Multimedia, 2011.

[94] S. Wang & S. Dey, "Modeling and characterizing user experience in a cloud server based mobile gaming approach", IEEE Global Telecommunications Conference, pp. 1-7, 2009.

[95] S. Wang & S. Dey, "Rendering adaptation to address communication and computation constraints in cloud mobile gaming", IEEE Global Telecommunications Conference, pp. 1-6, 2010.

[96] A. Kukulska-Hulme & J. Traxler, "Mobile learning: A handbook for educators and trainers", Psychology Press, 2005.

[97] M. Sharples, J. Taylor & G. Vavoula, "Towards a theory of mobile learning", mLearn 2005, vol. 1 no. 1, pp. 1-9, 2005.

[98] L. F. Motiwalla, "Mobile learning: A framework and evaluation", Computers & Education, vol. 49 no. 3, pp. 581-596, 2007.

[99] J. Traxler, "Defining, Discussing and Evaluating Mobile Learning: The moving finger writes and having writ", The International Review of Research in Open and Distance Learning, vol. 8 no. 2, 2007.

[100] D. B. Hoang & L. Chen, "Mobile cloud for assistive healthcare (MoCAsH)", IEEE Asia-Pacific Services Computing Conference, pp. 325-332, 2010.

[101] C. Doukas, T. Pliakas & I. Maglogiannis, "Mobile healthcare information management utilizing Cloud Computing and Android OS", 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1037-1040, 2010.

[102] Y. Wang, R. Chen & D. Wang, "A survey of mobile cloud computing applications: perspectives and challenges." Wireless Personal Communications, vol. 80, no. 4, pp. 1607-1623. 2015.

[103] Y. Xu & S. Mao, "A survey of mobile cloud computing for rich media applications." IEEE Wireless Communications, vol. 20, no. 3, pp. 46-53, 2013.

[104] K. Kumar & Y. H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" Computer vol. 43 no. 4, pp. 51-56. 2010.

[105] S. El-Sayed, M. Hatem, M. Abdul Kader, M. Hadhoud, & D. AbdElminaam, "Mobile cloud computing framework for elastic partitioned/modularized applications mobility." International Journal of Electronics and Information Engineering, vol. 1, no. 2, pp. 53-63. 2014.

[106] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, & R. Buyya, "A context sensitive offloading scheme for mobile cloud computing service", 2015 IEEE 8th International Conference on Cloud Computing, pp. 869-876. IEEE, 2015.

[107] X. Chen, "Decentralized Computation Offloading Game For Mobile Cloud Computing", IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 4, pp. 974-983. 2015.

[108] M. Shiraz, M. Sookhak, A. Gani, & S. A. A. Shah, "A study on the critical analysis of computational offloading frameworks for mobile cloud computing", Journal of Network and Computer Applications, vol. 47, pp. 47-60. 2015.

[109] S. Abolfazli, Z. Sanaei, M. Sanaei, M. Shojafar, and A. Gani. "Mobile cloud computing: the-state-of-the-art, challenges, and future research." Encyclopedia of Cloud Computing, Wiley, USA (2015).

[110] H. Qi & A. Gani, "Research on mobile cloud computing: Review, trend and perspectives", In Digital Information and Communication Technology and it's Applications (DICTAP), 2nd IEEE International Conference on, pp. 195-202, 2012.

[111] Power Tutor, Website link: https://play.google.com/store/apps/details?id=edu.umich.PowerTutor&hl=en, (Access Date: 27/12/2015)