EFFICIENT MOBILE CLOUD COMPUTING THROUGH COMPUTATION OFFLOADING

ABDUL KHADEER, PAMBALA NAGESWARA RAO, J RANJITH
4.J.SHARATH, 5.M.CHITTI BABU
Assistant Professor, Department of Computer Engineering, Ellenki college of Engineering and Technonlogy, patelguda (vi), near BHEL ameenpur (m), Sangareddy Dist. Telangana 502319.

ABSTRACT: In recent years, there has been a remarkable development of mobile devices such as smartphones, PDAs. With the great improvement of mobile application, users are not confined to use mobile phones with only calling and sending SMS. The multi-dimensional and useful different application markets (Apples iTunes, Google Play, Nokia's Ovi suite) paved the way to the smartphones users to sense the environment, make social contacts, use convenient applications (e.g., health care, e-commerce, mobile learning). Moreover, the cumulative effect of mobile devices enabled users to interact with different devices, environments, social community beyond time and space restriction. The revenue of mobile applications is expected to be more than 25 billion dollars by 2015 [1]. According to the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, it is estimated that 982 million smartphones will be conveyed worldwide by 2015 [2]. However, it is still challenging to develop extremely sophisticated applications on mobile devices because of the resource constraints like finite battery energy, low CPU speed, insufficient storage space, low network bandwidth and inadequate sensing capacities [3]. Besides, to minimize device fragmentation [4], virtually all smartphones adopt cross-platform runtime environments, such as Java ME, .NET CF, and Android to develop and run applications. So the challenges in mobility management, quality of service (QoS) insurance, energy management, and security issues come to the front.

Those problems influenced the researchers to search for the architecture which can provide the needed resource for mobile devices [5]. Cloud computing (CC) has been addressed to resolve the conflicts mentioned above. CC has been widely accepted as the future computing technology. It allows users to use infrastructure (e.g. servers, networks, and storages), platforms (e.g. middleware services and operating systems), and software (e.g. application programs) provided by cloud providers (e.g. Google, Amazon, and Salesforce) at on-demand computing, utility computing or pay as you go computing concept. The CC concept is based on offloading computation to remote resources providers over the internet. As CC enables users to elastically utilize resources in an on-demand fashion, mobile applications can be rapidly delivered with minimum effort and more computing power. For a large number of mobile application users, CC can provide a variety of services. This introduces Mobile Cloud Computing (MCC) as the integration of CC into a mobile computing environment.

In MCC the key idea is to offload a task from the mobile environment to the cloud process the task there and transmit the result to the mobile device again. This execution offloading technique decides which code region is to execute where during their runtime. By relieving loads, smartphones are benefited in terms of energy and execution time. The research of searching optimal partitioning of code into the distributed systems has developed several models. Some researchers [6] suggest static partitioning scheme which refers to a fixed job partition among the machines during its compile time. However, the mobility of the user can not ensure the terms of constant configuration always. So, some other work [7-9] has directed the dynamic and semi-dynamic execution offloading. In dynamic partitioning scheme, the part of the code region decides which part is to be transmitted to the server depending on the current status of the mobile device.

When the executable part is offloaded in the run time, the current state and control command are saved for further considerations. So, in MCC three types of tasks can be identified (1) those are processed locally, (2) those are processed remotely, (3) tasks are partitioned between mobile devices and cloud. Here, (2) and (3) need to consider the multi-dimensional network characteristics (WLAN, LTE, 3G, 4G) and their transmission costs, rates, availability. Obviously, there are trade-offs between the proposed execution offloading techniques. The questions may arise here are: (1) how can computation be offloaded?

(2) How the tasks are distributed to the cloud efficiently? (3) In which ways this differs from traditional distributed computing (4) which initiatives can be taken to persuade surrogate devices to share the workload?

In this paper, we discuss in detail the current research and findings of the energy efficient execution offloading techniques and thereby presented a mobile cloud dependent application model. First, we define MCC, execution offloading and why this is beneficial. Then we demonstrated different offloading schemes and their advantages. We explored and combined the schemes in appropriate conditions to get the best outcomes of MCC. The successful offloading and proper use of CC in the mobile environment leads to develop more sophisticated and richer applications and services such as mobile locating [10], voice, key-word, picture searching [11-13], mobile sensing [14] and mobile games [15], healthcare, e-commerce and many more. So MCC plays a vital role in developing mobile applications. Finally, we design these rich application models and examples for MCC.

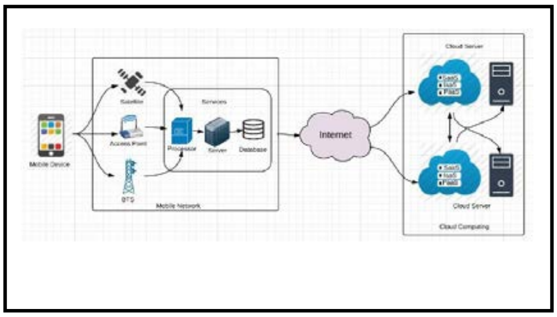
Overview of mobile cloud computing (MCC)

With the emerging development of cloud computing the term “mobile cloud computing” comes to the front. It attracted the entrepreneurs because of developing powerful mobile applications in lower cost and effort. MCC has drawn the attention of the researchers and has been a rapidly growing sector. This section introduces MCC, its architecture and advantages.

A. What is MCC?: MCC is a new paradigm that takes away the data processing and storage from the mobile device to the remote cloud and reduces a load of the mobile device in terms of processing power and energy. After the processing has been done in the cloud server, the resulting data is sent back to the device. With the advancement of data transmitting rate in recent times, the centralized cloud server access has been easy. The cloud servers are large and available as they provide all facilities of cloud computing. Therefore, MCC is the combination of mobile computing and cloud computing. Mobile computing constraints have been resolved through this technology combination. With the advancement of MCC, now users are able to utilize powerful applications to reach various services with their mobile devices.

B. The architecture of MCC: In mobile cloud networking architecture the mobile devices can connect to the cloud server in two ways (e.g., through mobile network or through access point network).

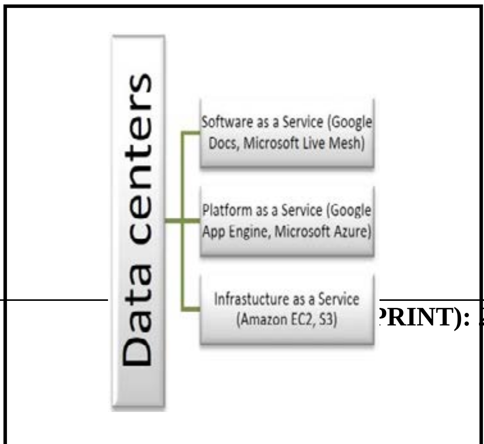
In mobile network case, the mobile devices are connected to the mobile network providers via base stations. The base station (e.g., transceiver station, access point, satellite) controls the connection speed, link capacity and functional interfaces between the network and the mobile users. The user’s requests are sent to the central processor which is connected to the server. This server maintains the service of a mobile network including authorization, authentication, and account validation on the basis of stored information in the database. Then the requests are sent to the clouds where the processing is done and the resulting data are sent back through the mobile network. Utility computing, virtualization, and service-oriented architecture is developed in the cloud servers (Figure 1).



In the access point case, the mobile devices connect to the access point through nearer Wi-Fi that is further connected to the Internet Service Provider (ISP) to provide internet connectivity. The users are able to access the cloud services without using the telecom network which may charge for data traffic. Moreover, the telecom network (e.g., 3G/HSDPA or 4G) connectivity requires more energy compared to the access point connection [16]. Therefore the mobile devices also prefer to use the Wi-Fi connectivity to access the cloud services when available.

The cloud computing architecture can be different in different contexts. For example, [17] proposed architecture for creating market-oriented clouds, [18] created a web-delivered business-oriented architecture. A four layer architecture is presented in [19] and cloud computing has been compared with grid computing. In this paper, we focused on the layered architecture of CC. This architecture generally describes CC in terms of its usability and efficiency.

Generally, CC is large scale architecture where a numerous data centers and servers provide continuous service in “pay as you go”, “on demand” and “utility” based concept. The cloud providers (e.g., Google, Amazon) provides infrastructure (e.g., servers, networks, and storages), platforms (e.g., middleware services and operating systems), and software (e.g., application programs) as their cloud service. In the layer concept (Figure 2) of CC commonly three layers are stacked. They are (1) Infrastructure as a Service, (2) Platform as a Service, (3) Software as a service.



Terminology of Figure 2

Data centers: This layer provides the hardware facility to all other layers. All the storage and core computing is done in this layer. This layer needs a continuous energy supply and heavy cooling system. Data centers are connected with very high-speed connectivity.

IaaS: Infrastructure as a service provides the vendors to use the storage and high computing power of servers. No control of service providers is imposed on this service. The service gainers use the infrastructures as they need. They can extend the limit anytime. They have to pay only for the part they used for a particular time. So this is very cost effective for the application developers. The examples are Amazon Elastic Cloud Computing (EC2) and Simple Storage Service (S3).

PaaS: Platform as a Service is an integrated environment for the developers to build custom applications. A particular developing environment binds the developers to work with particular tools and services. This is an easy and fast development platform which enables the user to access efficient applications. Google App Engine, Microsoft Azure, and Amazon Map Reduce/Simple Storage Service are the example of PaaS.

SaaS: Software as a Service is one kind of built application for the users in the cloud platform. The users can access those applications through the internet. The application modules and services are secured by the service providers. The users have to pay only for their use according to the playing rules of the providers. Sales Force is one of the pioneers of this cloud service model. Microsoft Azure is a SaaS model to share files and folders among different devices. Google Docs is another example.

Although the architecture model says about the layers it’s possible to build one layer without the support of the preceding layer. The layers can be implemented autonomously. The users select the service and layer according to their need for application flexibility.

The advantage of mobile cloud computing

Mobile computing problems have been resolved by the integration of cloud computing. Portability, energy limit and communication gap have been overcome through mobile cloud computing. In the following, we describe the advantages of MCC.

Energy saving: Mobile devices have limited battery lifetime. So developing applications having large and complex computation becomes infeasible because of enormous battery consumption. Several types of research have proposed the idea to save energy in computing or manage disk and screen [20,21]. But the proposed system requires new hardware configuration or change to the existing mobile devices architecture which is not practical or feasible. So the computation offloading technique is proposed which will take away large computing code region to the remote cloud. Rudenko et al. [22] evaluates the computation offloading and shows that up to 45% of energy saving is possible. In addition, offloading computation can save 41% energy in an image processing application. Memory arithmetic unit and interface

[16] proposes 27% energy consumption in a computer game and 45% in a chess game. Increasing processing power: The computation intensive applications need high processing power and speed. As mobile devices are poor in terms of processing power, the high scale computation cannot be done locally. Though some of them are possible to process locally, it takes a long time and energy. Mobile cloud computing helps to run those computing intensive codes in the cloudand retransmit the resulting data to the mobile devices. For example, clouds can be used for playing chess or broadcasting multimedia services to mobile devices. In all the cases the extreme computation like computer chess moves or voice searching is done in the resourceful clouds.

Storage capacity: Mobile devices have a storage limit. The cloud is the solution for uploading a large amount of data and accessing it from different devices necessarily. Amazon Simple Storage Service (S3) is an example for file storage service. Dropbox, a popular cloud storage service uses Amazons S3 cloud as its back-end storage. The AES-256 encryption system is used to secure the user's data in Dropbox. Another example is the image sharing cloud which occupies a large space to upload your images taken by mobile devices. The images are automatically when anyone captures a photo through the internet.

Security and reliability: The clouds can efficiently check the mobile user’s uploaded data and code whether it is malicious or virus containing. Moreover, the cloud service can protect users digital contents (e.g., video, audio, picture, article) from being unauthorized and abused. In addition, since the data are copied and protected in the cloud there is no risk of data loss. The cloud can obtain another user’s review and opinions to validate public information to improve reliability.

Computation offloading

The general idea is computation offloading to the cloud. The processing intensive tasks are difficult to run on the mobile devices because of its low processing speed and configuration. Moreover, battery limit is another big concern for running time-consuming applications. The idea is to offload the code regions to the cloud. The cloud can run the tasks efficiently and quickly comparably with the mobile device because of its powerful and large processors. It has been possible to build computing intensive applications by offloading techniques in recent times. However, issues such as heterogeneity of the system, physically distributed mobile devices and cloud, the inconsistency of data transmission rate have made the offloading decision difficult in a different environment.

A. Offloading decision making: Offloading decision for a particular computation can depend on several issues. Sometimes offloading can consume more energy and time in case of small amount of code. Then it is wise to compute that region locally. Again the user may not want to offload because of data charge. So the decision making and developing a convenient offloading technique is essential for successful mobile cloud computing.

A mobile cloud application goes through the steps showed in Figure

3. First, it is decided by the application that either the execution will be statically or dynamically. In a static process, the portion of the code which will be offloaded is predefined. So, after separating the code regions some tasks are processed locally and others are offloaded. In static offloading, there is any change of the offloading decision with the change of connection speed or another process effect. On the other hand, on the basis of current environmental conditions the offloading decision is taken in dynamic offloading. If offloading is enabled and the resource is available in the cloud then the favorite part is offloaded to the cloud. Otherwise, the code is executed in the mobile devices.

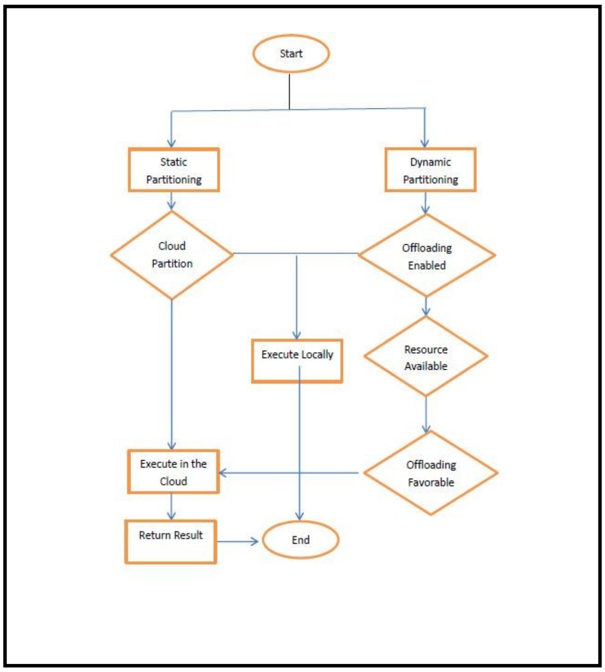


Figure 3: Computation offloading workflow

The decision making in offloading execution is a complex task. The offloading decision may be affected by network, mobile device, application model, cloud traffic and user. In a different environment, the offloading decision can be changed. Our aim is to reduce power utilization and execution time. The entities which can affect the offloading decision making are given below:

• Network: A mobile device can use a different network (e.g., 3G, 4G, LTE, and Wi-Fi) at different times because of its mobility. The different network uses different transmission technology having different connection speed and energy requirement. This network heterogeneity affects the offloading scheme. For example, mobile devices prefer to connect available Wi-Fi in spite of mobile network (3G/HSDPA, 4G). In Wi-Fi connection, smartphones require less battery consumption [16]

• Mobile device: Different mobile devices have different processing power and configuration. Day by day the smartphones and tablet PCs are becoming more powerful. One of the latest smartphone HTC One X has a 1.5 Ghz Quad-core processor, 1GB RAM, 32GB data storage support and 1800mAh battery. So, the dominant smartphones may require less offloading

• Application model: The design and objectives of the application define its model. The application model may also differ in terms of application security, cloud dependency, network availability and application partitioning. So the computation offloading decision depends on the nature of the application. Suppose, the application is a voice search application that searches the desired item from the user's voice input through his/her smartphone. For voice pattern recognition millions of training data are needed which cannot be possible to store locally. Therefore, the application model affects the offloading decision for its various purposes

• Cloud traffic: The cloud is supposed to provide continuous service to an unlimited number of requests. Nevertheless, the requests sent from a lot of mobile devices at a time may create a long queue. A new request will have to wait for a certain amount of time before the other requests are processed. Hence, the offloading decision isaffected by the traffic that hits the cloud. Moreover, the nature of the cloud, its resource, and service quality is crucial for offloading decision making

• User: A user may not wish to use cloud service for data privacy, communication cost, cloud service cost etc. So the user can enable and disable the cloud service. Besides, a user’s objective such as energy saving, quality service, application effectiveness can affect the offloading decision

The criteria stated above play a vital role in offloading scheme. However, it is the responsibility of the application model to make the decision based on the above criteria and offload task to enhance capability. We will discuss the application model in the next section.

B. Offloading method: Several types of research have been attempted to find an efficient offloading technique. Different offloading methods represent different network architecture, device configuration, and protocol. This paper describes the offloading methods which have developed in recent times. We discuss the offloading methods in three main directions: Client-Server Communication, Virtualization, Mobile agent.

• Client-Server communication: In client-Server Communication method, offloading is done from the mobile devices using Remote Procedure Call (RPC), Remote Method Invocation (RMI) and sockets. Both RPC and RMI have well supported APIs to communicate between surrogate devices but offloading through the protocol needs pre-installment on communicating devices. This disadvantage of Client-Server Method constrains the usability of mobile devices

• Spectra: Spectra is an example which uses pre-installed service on the mobile devices and servers to offload execution via RPC. The applications are designed to use RPC to invoke functionality in Spectra servers. When an application needs offloading, it contacts with a database that keeps the information about the server load, process queue etc. The application services are installed in the Spectra servers previously and thereby the application is designed. The developers manually partition the code region for offloading

• Hyrax: Marinelli [23] has presented Hyrax for Android applications which supports distributing of data and computation based on Hadoop. It uses a cluster of mobile devices as a resource provider. They proposed “HyraxTube” as a sample application for multimedia search and sharing. The objective of “HyraxTube” is to search files according to user’s requests from the local devices depending on their location, time and quality. Apache Hadoop is an open source implementation of MapReduce which provides a virtualized implementation on different devices. A central server controls the data and job offloading in „Hyrax‟. The mobile devices are connected through an isolated 802.11g network. The central server is not responsible for any processing task but does the job of device coordination. Like other Hadoop implementation, “Hyrax” has Name Node and JobTracker running on the central server with access to the mobile devices. Every android device runs DataNode and TaskTracker. As well as, the devices run threads to store data on Hadoop Distributed File System (HDFS). The DataNode and TaskTracker make heartbeats that are sent to the central NameNode and JobTracker. The heartbeats indicate that the device is alive and can process or transmit data. Then the central server decides the offloading and processing decision on particular device depending on their responses

• Cuckoo: The Cuckoo framework offloads jobs to the cloud based on the Java stub/proxy method. In this framework, all offloaded jobs are executed on the cloud rather than the mobile device cluster. Any cloud resource such as commercial Amazon EC2 or private cluster PCs that runs Java virtual machine can be used for processing the offloaded tasks. Cuckoos implementation of Android operating system targets the performance and battery consumption capability. The Ibis High-Performance Programming Interface is used for Cuckoos communication component. The application is re-written for running on the cloud. A programming model is to be developed for the Cuckoo implementation. For an Android device, the existing activity-service model is used which supports the task separation. The services (resource intensive tasks those are a candidate for offloading) and the activities (interactive methods of the application) are separated for making the offloading decision. When some tasks are offloaded to the cloud, Cuckoo generates the same version of the code for cloud processing because of the change of processors. The offloading decision also may depend on the network connection criteria. If the connection is not established the whole tasks are processed locally

• Mobile Message Passing Interface (MMPI): The MMPI framework is a mobile version of the standard MPI network where Bluetooth is used to communicate over devices. Mobile devices are the resource provider in MMPI. Mesh network interconnects the devices in spite of the typical star network. Tasks are

ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-3, ISSUE-10, 2016
DOI:10.21276/ijcesr.2016.3.10.4

handled by libraries so that Bluetooth specific codes need not be written explicitly. The framework is implemented by Java and the third party library BlueCove is used for Bluetooth operations. The master device passes the tasks to the slave devices for execution. The setting up MMPI network takes three steps: device discovery, service discovery and network discovery. The time of service discovery increases with the number of devices. After the completion of the first two devices, third, one takes less time

- Grid model: Deboosere et al. [24] propose a grid model which is actually a client-server protocol where a mobile device connects to the server as a thin client. A thin client protocol such as Virtual Network Computing (VNC) or streaming protocol is used to establish a connection. The user's request is sent to the server through a wireless network. Then the request is processed on the server and the response is sent back to the mobile device. The research focuses on the efficient algorithm to find a suitable server depending on the location of the device. The performance depends on selecting the nearer server to lower the battery consumption and bandwidth utilization

Virtual machine (VM) migration

VM migration refers to transferring memory image of a VM from one server to another server without stopping the execution. This is a live migration process where memory pages are pre-copied to avoid the interruption of OS or any application. VM migration is a seamless offloading technique where no code change is needed. It is secure in the sense that, the devices are insulated in operation. However, it may be proved time-consuming and heavy workload for running on the mobile devices.

- Cloudlet: During offloading tasks from mobile devices from the cloud the communication delay is an important factor. The rich application needs high-speed communication with the central cloud. Cloudlet is an emerging framework to reduce

communication delay. A Cloudlet is a resource-rich server which is well connected with the mobile devices and has a high-speed Local Area Network (LAN) connection. The mobile devices use nearby Cloudlet to offload tasks. Thus mobile users can reach the cloud by one hop, high bandwidth, and low-latency wireless network. If there is no Cloudlet nearby then using the default mode tasks will be offloaded to the central cloud or in the worst case processing is done locally. Satyanarayanan et al. [25] build an architecture which exploits virtual machine migration to accelerate offloading using wireless local area network. Their solution is to use transient customization of cloudlet infrastructure using hardware VM technology. The technology can overcome the drawbacks of CC. However, issues to be considered before implementing Cloudlet practically. Network capacity, processing distribution, storage are the challenges of Cloudlet. Security, management policy and cost- effectiveness should be taken into account in the future researches

- MAUI: Cuervo et al. [16] propose an architecture which uses both virtual migration and code partitioning in a dynamic environment. The application code is offloaded to the cloud in three steps. First, two versions of the application are built. One is for local execution and another is for remote execution. Today's smartphones use instruction set architecture (Advanced RISC Machine, ARM) which is different from PCs and servers. MAUI is designed to run a different version of the application in a different environment, preferably without any access to the program source code. Secondly, MAUI uses code reflection to mark the part of the code is 'remotely' or not. The 'remotely' methods are selected to be offloaded to the cloud. There are some types of code which should not be marked as 'removable' such as user interface, user input or the code which may be needed for reuse. Third, MAUI profiles every method of the application for serialization by determining the communication cost. Then it combines three factors: communication cost, the mobile devices battery consumption and network status (bandwidth, latency) to construct a linear programming formulation. The serialization enables to offload the suitable part of the application in a highly dynamic environment. The authors proposed MAUI as a fine-grained offloading to minimize the energy consumption without changing the application code

• Clone cloud: Clone Cloud also uses virtual migration to offload part of the application to the cloud using Wi-Fi or 3G network. Since the clone of the physical device is used for migration, the application need not be modified. Besides, developers do not annotate methods such as MAUI. Clone Cloud uses a 'cost model' to analyze the cost to offload and execute remotely comparable with local execution. The model was tested on Android phones, clones running on Dell desktop running on Ubuntu. They reported speedups up to 21.2 times with Wi-Fi over 3G. However, they establish trust in clone virtual migration environment for energy efficiency and emphasize future work on it

• MobiCloud: MobiCloud model proposes to use cloud computing technology for MANETs (Mobile Ad hoc NETworks) in a secure way. MobiCloud can transfer MANETs into a service-oriented structure. In MobiCloud every mobile device is supposed to be a service provider or service broker. These service nodes are considered as a virtualized component and mirrored in the cloud. The Extended Semi-Shadow Images (ESSIs) are not exact as the virtual machine migration because the service nodes can be a clone, a partial clone, or just an image of the real device which can extend some functionality

Mobile agent

Scavenger: Scavenger is another agent based framework which employs a cyber-foraging using Wi-Fi connectivity. It uses a mobile code approach to distribute jobs to surrogate mobile devices. It introduces a scheduler for cost assessment. The cost assessment is based on the network speed and a benchmarking method is used to do that. The distributed jobs are executed in parallel on surrogate devices which enhances the performance. However, it does not provide any method for fault tolerance. Since Scavenger only offloads jobs on the surrogate and does not share execution, it is not really dynamic.

Discussion

The most recent researches have focused on the virtual machine migration or mobile code as offloading technique. Without Hyrax, Virtual Cloud, Cuckoo other frameworks do not use client-server architecture. Though virtual machine migration and mobile code use older framework Hadoop and Ibis which was designed for distributed and grid computing, the methods help VM migration over conventional client-server architecture. The reason is Hadoop and Ibis have more advantages than RPC. Although client-server communication has well supported APIs, it is not so robust in case of the distributed mobile network. It requires pre-installed services on devices and disconnected operations are not supported. Considering the heterogeneity, on-going continuous connection and ad hoc nature of mobile computing researchers emphasize the VM migration and mobile code frameworks.

Cloudlet, CloneCloud, MAUI, MobiCloud use VM migration and it reduces a lot of burden from the developer to rewrite application code. However, full virtualization is not suitable for fine-grained code partitioning, although rewriting the full or part of the application is not expected. MAUI uses a combination of both virtualization and code partitioning. Again for the mobility of devices sometimes the VM migration may be proved heavyweight thereby Scavenger uses the mobile agent to offload to surrogate devices in a dynamic environment.

Applications of MCC

With the evolvement of cloud computing, there has been opened many opportunities to build rich applications for mobile devices. Resource and energy problems have been reduced by cloud technology. Now the vendors are able to build new applications for smartphones which were almost impossible previously. Therefore, the global mobile application market is greatly influenced by mobile cloud computing. In this section, the different MCC applications are introduced.

A. Augmented reality: A new class of mobile applications is the augmented reality which has drawn user's attention in recent time. The idea is to reduce the distance between the cyber world and the real world. Wearable mobile devices paved the way to use applications in real life unlike virtually. Today's smartphones have many sensors to perceive the environment and reality. The gestural interface

Sixth Sense and Google Glass are the devices which can implement the applications of augmented reality. Sixth Sense can project live news on real newspaper; Google glass allows the user to use the calendar, map direction, video conversation and so on. Augmented reality can be also implemented in games where there will be real-world objects. The applications are built in a way that it appears to the user as a real-world event. However, for building real-world applications mobile devices need high processing power and storage. CC is the solution for the problems. Using the power of cloud new augmented reality applications can be built.

B. Mobile healthcare: The purpose of the mobile healthcare is to enhance facility to the patients, doctors, and hospitals. It will make the medical treatment less costly, easy and error free. Besides, it provides the opportunity to use standalone local server applications to the hospitals [26]. Five mobile healthcare applications in the universal environment are presented:

- Health monitoring services provide the opportunity to communicate with the user continues to monitor and record the condition of the patient
- Emergency services can respond to the medical emergency (e.g., car accident) and send a medical team to the exact location immediately
- Health-aware mobile devices can detect pulse rate, blood pressure, level of sugar or alcohol and send the record to the monitoring service
- The universal database can allow users and doctors to access the present or past health information anytime from anywhere
- Universal lifestyle management can help to pay healthcare expenses and other related charges automatically

For a practical system, a telemedicine system is implemented in Taiwan where the users (especially patients of hypertension and diabetes) can get medical services. Here 300 users can get support and 4736 records regarding sugar and blood pressure is stored in the cloud.

C. Mobile commerce: Mobile commerce is a business model for commercial activities using a mobile phone. Now a day, maximum financial transactions are done through the internet. Mobile commerce can enhance the finance, advertising, shopping because of its mobility and availability. Different secure applications (e.g., mobile payment, transaction, ticketing) can be designed for mobile devices using the power of cloud computing. Several issues like network latency, security may arise for mobile commerce. Yang et al. [27] propose an infrastructure for mobile commerce which combines the advantage of the 3G network and CC. They use Public Key Infrastructure (PKI) for data security. PKI uses encryption-based access control and over- encryption to secure the transaction.

D. Mobile learning: Mobile learning (m-learning) is developed based on electronic learning (e-learning). The mobile devices can be a useful way for learning and keep in touch with large stores of information. The traditional mobile learning needed costly devices, communication delay, and low capacity of storage. CC can broaden the idea and made possible to build rich mobile learning applications with high processing power, large information storage, and low cost. Zhao et al. [28] present a combination of mobile learning and CC to enhance the communication of students and teachers. The students communicate with the teachers over a web application built in Google App Engine. The teachers can monitor students' knowledge and answer the questions in a timely manner. In addition, a contextual m-learning system developed in augmented reality platform helps students to get learning resources more easily.

E. Mobile gaming: Mobile gaming is a rapid growth sector from which a lot of revenue is earned. Vendors are now trying to build newer and more attractive games to draw user's attention. With the support of cloud, high resource consuming games can be developed. All the processing (e.g., graphics rendering) intensive tasks are offloaded to the cloud servers and only user interface control is done locally. Li et al. [29] demonstrate that offloading processing task can save energy and increases the time of playing. Several experiments have been done to find suitable offloading decision over 3G and Wi-Fi network to save mobile device's energy. The experiments show that instead of offloading the whole part to the cloud, it is wise to partition application code in the run time depending on the communication cost and processing cost in terms of bandwidth and energy. This technique can save energy up to 45% for video games and 27% for chess. The refresh rate also increases from 6 to 13 frames per second.

F. Web applications: With the development of web browser different user attractive web application is being built. HTML 5 and JavaScript can now work with high performance in mobile devices. Besides, being cross-platform in nature web applications and web games are now more powerful using Canvas, 3D transform, complex logic, and visual effects. However, JavaScript interpretation and rendering tasks are energy consuming for mobile devices. So, those tasks are offloaded to the cloud. Comet experimented accelerated the task up to 8.5 times. Amazon's Silk browser for its Kindle Fire tablet uses a partitioning task. Silk uses browser subsystem which can run some part of the computation locally and some remotely. When a request comes the Silk browser divides tasks for local and remote computation based on the page complexity, resource size etc. With the help of cloud computing, future web applications can be far improved.

Conclusion

Since mobile devices are being more powerful through MCC, more research can be undertaken to find a convenient strategy. The mobile cloud can change the technology trend and bring a huge improvement in our daily lives by solving many real-life problems. In this paper, we unveiled the mobile cloud computing. We demonstrated the offloading decision making and offloading techniques to make the best use of MCC. Later, we have described the application models which can be the future mobile apps and can change the view of today's application.

References

1. https://www.marketsandmarkets.com/Market-Reports/mobile- applications-228.html
2. https://www.businesswire.com/news/home/20110609005403/en/ Worldwide-Smartphone-Market-Expected-Grow-55-2011.
3. Conti M, Chong S, Fdida S, Jia W, Karl H, et al. (2011) Research challenges towards the Future Internet. Computcommun 34: 2115-2134.
4. Gavalas D, Economou D (2010) Development platforms for mobile applications: status and trends. IEEE Software 28: 77-86.
5. Yang D, Xue G, Fang X, Tang J (2012) Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. Proceedings of the 18th Annual International Conference on Mobile Computing and Networking.
6. Newton R , Toledo S, Girod L, Balakrishnan H, Madden S (2009) Wishbone: profile-based partitioning for sensornet applications. Netw Sys Design Implemen 9: 395-408.
7. Cuervo E, Balasubramanian A, Cho DK, Wolman A, Saroiu S, et al. (2010) MAUI: making smartphones last longer with code offload. MobiSys 10: 49-62.
8. Chun BG, Ihm S, Maniatis P, Naik M, Patti A (2011) Clone Cloud: elastic execution between mobile device and cloud. Euro Sys 11: 181-194.
9. Ra MR, Sheth A, Mummert L, Pillai P, Wetherall D, et al. (2011) Odessa: enabling interactive perception applications on mobile devices. Mobi Sys 11: 43-56.
10. Kwok YK, Hwang K, Song S (2007) Selfish grids: game theoretic modeling and NAS/PSA benchmark evaluation. IEEE Trans Parallel Distrib Sys 18: 621-636.
11. Subrata R, Zomaya AY, Landfeldt B (2008) A cooperative game framework for QoS guided job allocation schemes in grids. IEEE Trans Comput 57: 1413-1422.
12. Ghosh P, Basu K, Das SK (2007) A game theory-based pricing strategy to support single/multiclass job allocation schemes for bandwidth- constrained distributed computing systems. IEEE Trans Parallel Distrib Sys 18: 289-306.
13. Danezis G, Lewis S, Anderson R (2005) How much is location privacy worth?

Proceedings of Workshop on the Economics of Information Security.

14. Lee JS, Hoh B (2010) Sell your experiences: a market mechanism based incentive for participatory sensing. Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications.

15. Duan L, Kubo T, Sugiyama K, Huang J, Hasegawa T, et al. (2012) Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing. Proceedings of the Annual IEEE International Conference on Computer Communications.

16. Cuervo E, Balasubramanian A, Cho DK, Wolman A, Saroiu S, et al. (2010) Maui: making smartphones last longer with code offload. Proceedings of 8th international conference on Mobile systems, applications, and services.

17. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. J Future GenerComputSyst 25: 599-616.

18. Huang Y, Su H, Sun W, Zhang JM, Guo CJ, et al. (2010) Framework for building a low-cost, scalable, and secured platform for web delivered business services. IBM J Res Dev 54: 1-14.

19. Foster I, Zhao Y, Raicu I, Lu S (2009) Cloud computing and grid computing 360-degree compared. Proceedings of Workshop on Grid Computing Environments.

20. Davis JW (1993) Power benchmark strategy for systems employing power management. Proceedings of the IEEE International Symposium on Electronics and the Environment.

21. Mayo RN, Ranganathan P (2003) Energy consumption in mobile devices: why future systems need requirements aware energy scale-down. Proceedings of the Workshop on Power-Aware Computing Systems.

22. Rudenko A, Reiher P, Popek GJ, Kuenning GH (1998) Saving portable computer battery power through remote process execution. J ACM SIGMOBILE Mob ComputCommun Rev 2: 19-26.

23. Marinelli EE (2009) Hyrax: cloud computing on mobile devices using MapReduce, Carnegie Mellon University, USA.

24. Deboosere L, Simoens P, Wachter JD, Vankeirsbilck B, Turck FD, et al. (2011) Grid design for mobile thin client computing. Future GenerComputSyst 27: 681-693.

25. Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for VM- based cloudlets in mobile computing. IEEE Pervasive Comput 8: 14-23.

26. Varshney U (2007) Pervasive healthcare and wireless health monitoring. J Mobile Net App 12: 113-127.

27. Yang X, Pan T, Shen J (2010) On 3G mobile e-commerce platform based on cloud computing. 3rd IEEE International Conference on Ubi-Media Computing.

28. Zhao W, Sun Y, Dai L (2010) Improving computer basis teaching through mobile communication and cloud computing technology. Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering.

29. Li Z, Wang C, Xu R (2001) Computation offloading to save energy on handheld devices: a partition scheme. Proceedings of the 2001 international conference on Compilers, architecture, and synthesis for embedded systems.

ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-3, ISSUE-10, 2016
DOI:10.21276/ijcesr.2016.3.10.4