International Journal of Computing and Mathematics
Vol:3 Issue:2 June 2019

Energy Efficiency MCC through Calculation Offloading

Wasan Abdulwahab Mohammed
Directorate of Education in Karkh 1, Ministry of Education, Iraq

Abstract: Mobile cloud computing (MCC) has been implemented as a potential technology with the explosive development of mobile apps and the changing idea of cloud computing. MCC is the integration of cloud computing into the mobile technology environment. It refers to an infrastructure where both data processing and data storage are removed from the mobile device into the cloud. The aim is to overcome barriers in mobile computing linked to battery life, storage, bandwidth, heterogeneity, and safety. To throw away the computational burden of mobile devices, the method of offloading execution is suggested to migrate the process between computers. In execution offloading the choice where the software areas to be performed are chosen statically or dynamically depending on the execution moment and process state. MCC's success relies on the effective offloading method that meets energy consumption demands.

Keywords: MCC, Offloading, Cloud computing, Energy efficiency

1. INTRODUCTION

In recent years have seen a remarkable growth of mobile devices such as smartphones, PDAs. With the great improvement of the mobile app, users are not confined to using cell phones with only calling and sending SMS. The multi-dimensional and helpful distinct application markets (Apple iTunes, Google Play, Nokia's Ovi suite) paved the way for users of smartphones to feel the atmosphere, create social connections, use convenient apps (e.g., health care, e-commerce, mobile learning). Also, the cumulative impact of mobile devices allowed people to communicate beyond moment and space limitation with different apps, environments, social community. Mobile application revenues are expected to exceed $25 billion by 2015 [1]. It is estimated that 982 million smartphones will be transmitted worldwide by 2015 [2], according to the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker.

However, it is still challenging to create highly advanced apps on mobile devices due to resource constraints such as finite battery power, low CPU speed, insufficient storage space, low network bandwidth, and inadequate sensing capabilities [3]. Also, to minimize device fragmentation [4], nearly all smartphones embrace cross-platform runtime environments such as Java ME,.NET CF, and Android to create and operate apps. Thus the difficulties of mobility management, quality of service (QoS) insurance, energy management, and safety problems come to the front.

These problems influenced the researchers in searching for the architecture that could provide the necessary resource for mobile devices [5]. Cloud computing (CC) has been used to resolve the above-mentioned conflicts. CC has been widely accepted as the technology of future computing. It enables users to use facilities (e.g. servers, networks, and storages), platforms (e.g. middleware services and operating systems), and software (e.g. application programs) supplied by cloud providers (e.g. Google, Amazon, and Salesforce) on-demand computing, utility computing, or pay-as-you-go computing. The idea of the CC is based on offloading computing over the web to remote resource providers. As CC allows users to use resources on-demand elastically, mobile apps can be delivered quickly with minimal effort and more computational power. CC can provide a variety of services for a large number of users of mobile apps. This introduces Mobile Cloud Computing (MCC) to integrate CC into a mobile computing environment.

In MCC, the main concept is to unload a job from the mobile setting to the cloud process the job there and transmit the outcome to the mobile device. This technique of offloading execution decides which code region is to execute where during their runtime. Smartphones benefit in terms of energy and execution time by relieving loads. Several models have been developed to search for optimal partitioning of code in distributed systems. Some researchers [6] suggest a static partitioning scheme that refers to a fixed job partition between machines during its compilation time. The user's mobility, however, cannot always ensure constant configuration terms. So some other work [7-9] directed the offloading of dynamic and semi-dynamic execution. In the dynamic partitioning system, the portion of the code region chooses which portion is to be transmitted to the server based on the mobile device's current status.

When the executable portion is offloaded in runtime, the present state and control command is saved for further consideration. Three kinds of assignments can be recognized in MCC (1) they are processed locally, (2) they are processed remotely, (3) tasks are partitioned between mobile devices and cloud. Here, (2) and (3) need to consider multidimensional network characteristics (WLAN, LTE, 3 G, 4 G) and their transmission expenses, prices, accessibility.

There are trade-offs between the offloading methods suggested for execution. The questions that may emerge here are:
(1) How can computation be downloaded?
(2) How are the tasks effectively transferred to the cloud?
(3) In what ways does this differ from traditional distributed computing
(4) Which initiatives can be taken to convince surrogate devices to share the workload?

In this article, we discussed the present studies and results of energy-efficient execution offloading methods in detail and presented a mobile cloud-dependent application model. First, we describe MCC, execution offloading and why this is useful. Then we illustrated different offloading systems and their benefits. We studied and paired the systems under suitable conditions to achieve the highest results of MCC. The effective offloading and proper use of CC in the mobile setting contributes to the development of more advanced and richer apps and services such as mobile location [10], voice, keyword, image searching [11-13], mobile sensing [14] and mobile games [15], healthcare, e-commerce and much more. MCC plays a crucial role in the development of mobile apps. Finally, we design these wealthy MCC application models and examples.

2. OVERVIEW OF MOBILE CLOUD COMPUTING (MCC)

The term "mobile cloud computing" comes to the front with the emerging development of cloud computing. It attracted entrepreneurs by developing powerful mobile apps with lower price and effort. MCC has attracted the researchers' attention and has been a quickly increasing industry. This chapter presents MCC, its architecture, and its benefits.

A. What is MCC

MCC is the latest paradigm that removes data processing and storage from the mobile device to the remote cloud and decreases the mobile device's processing power and energy load. After the processing is performed in the cloud server, the resulting information is sent back to the computer. Recently, with the development of data transmission rate, centralized cloud server access has been simple. The cloud servers are big and accessible as they provide all cloud computing services. MCC is, therefore, the mixture of mobile computing and cloud computing. Mobile computing limitations have been addressed through this mixture of technology. With the development of MCC, consumers can now use strong apps to achieve multiple facilities with their mobile devices.

B. The architecture of MCC

Mobile devices can connect to the cloud server in two ways in mobile cloud networking architecture (e.g. via mobile network or access point network).

In the event of mobile networks, mobile phones are linked via base stations to mobile network providers. The base station (e.g. transceiver station, access point, satellite) regulates the connection velocity, connection capability, and functional interfaces between the network and mobile users. Requests from the user are sent to the server-connected central processor. This server holds a mobile network service including approval, authentication, and account validation based on data stored in the database. Then the requests will be sent to the clouds where the processing is done and the resulting data will be sent back through the mobile network. The cloud servers develop utility computing, virtualization and service-oriented architecture (Figure 1).

International Journal of Computing and Mathematics
Vol:3 Issue:2 June 2019

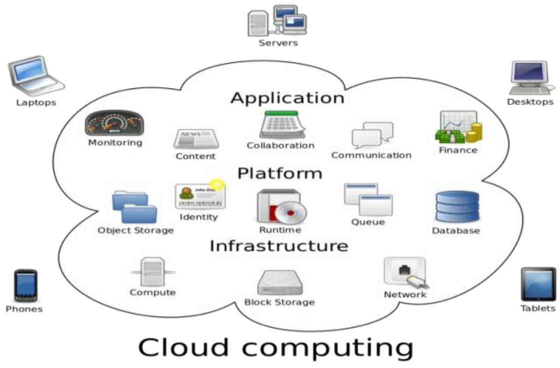

Figure 1. Cloud computing architecture

In the event of the access point, mobile devices connect to the access point via closer Wi-Fi, which is further attached to the Internet Service Provider (ISP) to provide internet connectivity. Users can access cloud services without using the telecom network that can pay for information traffic. Furthermore, the connectivity of the telecom network (e.g. 3G/HSDPA or 4G) requires more energy compared to the connection point [16]. Therefore, when available, mobile devices also prefer to use Wi-Fi connectivity to access cloud services.

The cloud computing architecture can be distinct in distinct situations. For instance, [17] suggested architecture to create market-oriented clouds [18] developed a web-based business-oriented architecture. A four-layer architecture is described in [19] and cloud computing has been compared with grid computing. In this article, we concentrated on CC's layered architecture. CC's usability and effectiveness are usually described in this architecture.

Generally, CC is a large-scale architecture where numerous data centers and servers provide continuous service in the concept of "pay as you go," "on-demand" and "usefulness." Cloud providers (e.g. Google, Amazon) provide as their cloud service infrastructure (e.g. servers, networks, and storages), platforms (e.g. middleware services and operating systems), and software (e.g. application programs). Three layers are frequently stacked in CC's layer idea (Figure 2).

They are
(1) Service Infrastructure,
(2) Service Platform,
(3) Service Software.

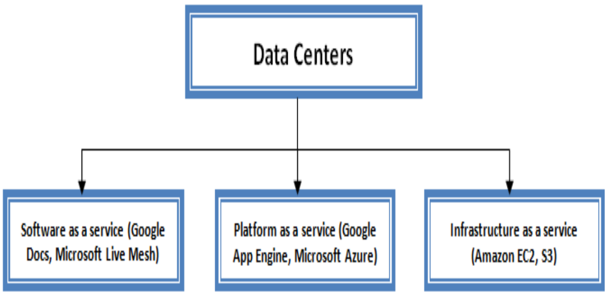

Figure 2.Service-based cloud computing architecture

3. TERMINOLOGY OF FIGURE 2

Data centers: This layer offers all other layers with the hardware unit. In this layer, all the storage and key computing are performed. This layer requires a constant supply of energy and a heavy cooling system. Data centers are linked to very high-speed connectivity.

IaaS: Infrastructure as a service offers suppliers with the ability to use server storage and high computing capacity. No service provider control is placed on this service. The service gainers use the infrastructure they need. They can stretch the limit at any time. They only have to pay for the portion they used for a specific moment. So this is very cost-effective for designers of applications. Examples include Amazon Elastic Cloud Computing (EC2) and Simple Storage Service (S3).

PaaS: Platform as a Service is an integrated environment for developers to construct customized apps. A specific development environment connects developers to work with specific instruments and services. This is a simple and quick development platform that allows users to access effective apps. The PaaS example is Google App Engine, Microsoft Azure and Amazon Map Reduce/Simple Storage Service.

SaaS: Software as a Service is one type of built application for cloud platform users. Through the internet, users can access these applications. Service providers secure application modules and services. The users only have to pay for their use following the providers' play rules. Sales Force is one of the pioneers in this model of cloud service. Microsoft Azure is a SaaS model for sharing files and folders across different devices. Google Docs is just another example.

Although the architecture model says about the layers, it is feasible to construct one layer without the assistance of the previous layer. The layers can be autonomously

International Journal of Computing and Mathematics
Vol:3 Issue:2 June 2019

introduced. Users pick the service and layer according to their need for flexibility in the implementation.

Mobile Cloud Computing Advantages

The integration of cloud computing has solved mobile computing issues. Portability, energy limit, and communication gap were overcome through mobile cloud computing. We define the benefits of MCC in the following.

Energy-saving: Mobile phones have restricted battery life. Because of huge battery usage, the development of apps with big and complicated computing becomes unfeasible. Several kinds of studies have suggested the concept of saving energy in computing or managing disk and screen [20, 21]. But the suggested scheme needs fresh hardware setup or a shift to the current architecture of mobile devices that is not practical or feasible. Thus, the computing offloading method is suggested that will take away a big computing code region to the distant cloud. Rudenko et al. [22] evaluates the offloading of the computation and shows that up to 45 percent of energy savings are possible. Additionally, in an image processing application, the offloading computation can save 41 percent energy. Memory arithmetic unit and interface [16] offers 27 percent power usage in a computer game and 45 percent in a chess game. Increasing processing power: computation-intensive apps require high processing power and velocity. Since mobile devices are poor in terms of processing power, high-scale computing cannot be done locally. Although some of them can be processed locally, it requires a long time and energy. Mobile cloud computing helps operate these computing-intensive codes in the cloud and retransmit the resulting information to mobile devices. For example, clouds can be used to play chess or broadcast multimedia services to mobile devices. In all cases, in the resourceful clouds, extreme computation such as computer chess moves or voice searching is done.

Storage capacity: Mobile phones have storage limits. The cloud is the answer to necessarily upload and access a big quantity of information from distinct devices. Amazon Simple Storage Service (S3) is an example of a file storage service. A famous cloud storage service, Dropbox uses Amazons S3 cloud as its back-end storage.
The AES-256 encryption scheme is used to secure user data in Dropbox. Another instance is the image sharing cloud, which occupies a big room for uploading your pictures from mobile devices. The pictures are automatically taken when someone gets a picture through the internet.

Security and reliability: The clouds can effectively inspect the uploaded information and code of the mobile user whether it is malicious or containing a virus. Also, the cloud service can safeguard customers from unauthorized and abused digital content (e.g., video, audio, photo, article). Furthermore, there is no risk of data loss since the data is copied and protected in the cloud. The cloud may obtain reviews and opinions from another user to validate public information to improve reliability.

Computation offloading: The general idea is to offload computation to the cloud. Due to its low processing speed and setup, the processing-intensive tasks are hard to operate on mobile devices. Also, battery restriction is another major problem for running time-consuming apps. The concept is to offload the code areas to the cloud. Due to its strong and big processors, the cloud can perform the tasks effectively and rapidly comparably with the mobile device. In the latest times, it has been feasible to construct computing-intensive apps by offloading methods. However, problems such as system heterogeneity, physically distributed mobile devices, and cloud, the inconsistency of the data transmission rate has made the offloading decisions hard in a distinct setting.

A. Offloading Decision Making

Offloading decisions for a specific computation can depend on several problems. Offloading can sometimes consume more energy and time in the event of a small amount of code. Then it's wise to calculate that region locally. The user may not want to offload again because of information charges. The decision-making and development of a useful offloading method are therefore crucial for effective mobile cloud computing.

A mobile cloud application goes through the steps shown in Figure 3. First, the request decides that either the execution will be static or dynamic. The part of the software that will be offloaded is predefined in a static method. So, some tasks are handled locally after dividing the code areas and others are offloaded. In static offloading, there is any change in the choice of offloading with the change of link velocity or another process impact. On the other side, the choice of offloading is made in dynamic offloading based on the present environmental circumstances. If the offload is activated and the resource is accessible in the cloud, the favorite portion will be offloaded to the cloud. Otherwise, the code will be executed in the mobile devices.

International Journal of Computing and Mathematics
Vol:3 Issue:2 June 2019

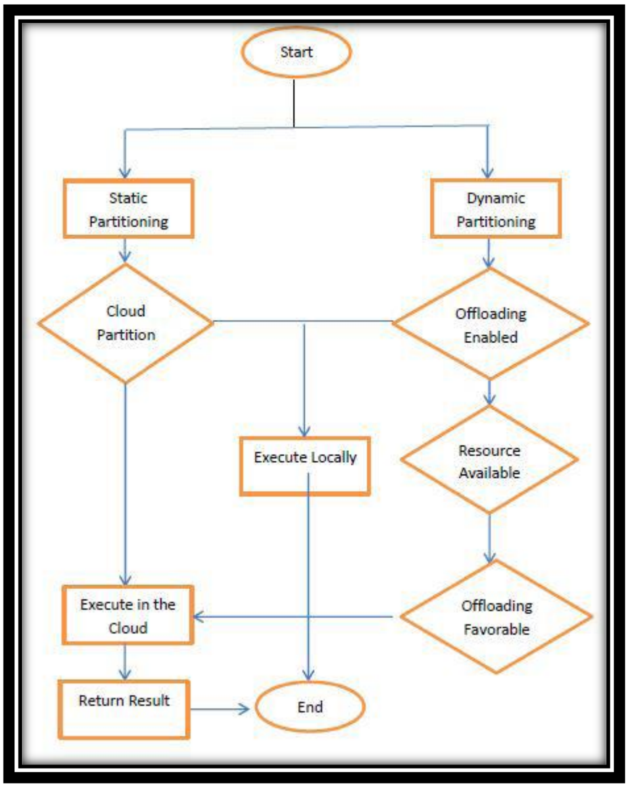

Figure 3.Computation offloading workflow

Decision-making in offloading execution is a complicated job. Network, mobile device, application model, cloud traffic and a user may affect the decision to offload. The decision on offloading can be changed in a different environment. Our goal is to reduce power utilization and time for execution. The entities that may affect the decision making of offloading are listed below:

Network: Due to its mobility, a mobile device can use a distinct network (e.g. 3 G, 4 G, LTE, and Wi-Fi) at different moments. The separate network utilizes distinct transmission technology with varying link speeds and energy requirements. This heterogeneity of the network affects the system of offloading. Mobile devices, for example, prefer to connect Wi-Fi available in spite of mobile networking (3G / HSDPA, 4 G). Smartphones require less battery consumption in Wi-Fi connection [16].

Mobile device: Various mobile devices have different processing power and configuration. Smartphones and tablet PCs become more powerful every day. One of the latest HTC One X smartphones has a 1.5 GHz Quad-core processor, 1GB RAM, 32GB data storage assistance and 1800mAh battery. Thus, the dominant smartphones may require fewer offloading.

Application model: The application's design and goals describe its model. The application model may also vary in terms of application security, cloud dependence, network availability and application partitioning. So the choice to offload the computation relies on the nature of the application. Suppose the application is a voice search software that processes the required object from the user's speech input via his / her computer. Recognition of voice patterns requires millions of training data that cannot be stored locally. The application model, therefore, impacts the discharge choice for multiple reasons.

Cloud traffic: The cloud should provide ongoing service to an infinite amount of demands. Nevertheless, applications sent from a lot of mobile devices at a moment can generate a lengthy queue. A new request will have to wait a certain amount of time before processing the other requests. Therefore, the offloading decision is affected by the traffic that hits the cloud. Moreover, the nature of the cloud, its resource, and the quality of service is essential to offloading decision-making.

User: A user may not wish to use the cloud service for data privacy, communication costs, cloud service costs, etc. This allows the user to enable and disable the cloud service. Furthermore, the goal of a user such as energy-saving, quality service, the efficiency of application can affect the offloading decision.

B. Offloading Method

Several kinds of studies have been tried to discover an effective method for offloading. Different techniques of offloading constitute varying network architecture, device setup, and protocol.

Client-Server communication: Offloading from mobile devices is performed using Remote Procedure Call (RPC), Remote Method Invocation (RMI) and sockets in the client-server communication method. Both RPC and RMI have well-supported APIs to interact between surrogate devices but need pre-installment on communicating devices through the protocol. This Client-Server Method disadvantage restricts the usability of mobile devices.

Spectra: Spectra is an instance that utilizes pre-installed service on mobile devices and servers to offload execution via RPC. Applications are intended to use RPC to invoke features in Spectra servers. When an application needs to be offloaded, it contacts a database that keeps the server load data, process queue, etc.

Previously, the application services are mounted in the Spectra servers and thus the application is intended. The developers partition the code region manually for offloading.

Hyrax: Marinelli [23] presented Hyrax for Android applications that support Hadoop-based data distribution and computing. It uses as a resource provider a cluster of mobile devices. They have proposed "Hyrax Tube" as a sample application for multimedia search and sharing. The goal of "Hyrax Tube" is to search files from the local devices according to user requests depending on their location, time and quality. Apache Hadoop is an open-source Map Reduce implementation that provides a virtualized implementation on various devices. A central server controls information and job offloading in "Hyrax." Mobile devices are connected via an isolated 802.11 g network. The main server is not liable for any processing assignment but does the task of device coordination. Like other implementation of Hadoop, "Hyrax" has Name Node and Job Tracker operating on the main server with access to mobile devices. Every Android device is running Data Node and Task Tracker. The devices are also run threads for storing data on the Hadoop Distributed File System (HDFS). The Data Node and Task Tracker create heartbeats sent to the main Name Node and Job Tracker. The heartbeats show the device is alive and can process or communicate information. Then the main server chooses the choice to unload and process a specific machine based on their answers.

Cuckoo: The Cuckoo framework, based on the Java stub / proxy technique, offloads employment to the cloud. In this context, all offloaded jobs are executed on the cloud rather than on the cluster of mobile devices. Any cloud resources such as commercial Amazon EC2 or personal cluster PCs running Java virtual machines can be used to process offloaded duties. Cuckoos' application of the Android operating system targets the capacity of efficiency and battery consumption. The Ibis High-Performance Programming Interface is used for the communication element of Cuckoos. The application is rewritten to run on the cloud. A programming model for the application of Cuckoo is to be created. The current activity-service model that promotes task separation is used for an Android phone. The services (resource-intensive tasks that are a candidate for offloading) and the activities (interactive implementation techniques) are segregated for making the offloading choice. When some tasks are offloaded to the cloud, due to processor change, Cuckoo generates the same version of the code for cloud processing. Offloading choice may also rely on the network link criteria. If the link is not formed, all duties are processed locally.

Mobile Message Passing Interface (MMPI): The MMPI framework is a portable variant of the conventional MPI network that uses Bluetooth to interact over computers. Mobile phones are the MMPI resource supplier. Mesh network interconnects equipment despite the typical star network. Libraries handle tasks so that particular Bluetooth codes do not need to be written explicitly. The framework is introduced by Java and the BlueCove third party library is used for Bluetooth activities. The master unit moves the duties for execution to the slave systems. MMPI network setup requires three measures: device discovery, service discovery, and network discovery.

Grid model: Deboosere et al. [24] suggest a grid model that is effectively a client-server protocol where a mobile device connects to the server as a thin client. To create a link, a thin client protocol such as Virtual Network Computing (VNC) or streaming protocol is used. The request from the user is sent to the server through a wireless network. The request will then be processed on the server and the response will be sent back to the mobile device. The study focuses on the effective algorithm to locate an appropriate server based on the place of the device. Efficiency relies on choosing the nearest server to reduce battery consumption and bandwidth usage.

Virtual Machine (VM) Migration

VM migration refers to transfer the memory image of a VM from one server to another server without stopping the execution. This is a live migration method where memory pages are pre-copied to prevent interrupting OS or any application. VM migration is a seamless offloading method that requires no code change. It is safe in the sense that in service the instruments are insulated. However, the time-consuming and heavy workload can be proven to run on mobile devices.

Cloudlet: Communication delay is a significant consideration during offloading functions from the cloud from mobile devices. The rich application requires high-speed communication with the main cloud. Cloudlet is an evolving framework for reducing communication delays. A Cloudlet is a resource-rich server that is well linked to mobile devices and has a high-speed Local Area Network (LAN) link. Mobile phones use Cloudlet nearby to download assignments. This allows mobile consumers to reach the cloud through one hop, high bandwidth, and low latency wireless network. If there is no Cloudlet nearby, the default mode tasks will be offloaded to the central cloud or processed locally in the worst case. Satyanarayanan et al.[25] create an architecture that exploits virtual machine migration to accelerate offloading using wireless local area network. Their alternative is to use temporary customization of cloudlet infrastructure using VM hardware technology. The technology can overcome CC's drawbacks. However, problems that need to be considered before practically implementing Cloudlet. Cloudlet's difficulties are network capability, distribution processing, storage. Security, management strategy and cost-effectiveness should be taken into consideration in future research.

MAUI: Cuervo et al.[16] suggests an architecture that utilizes both virtual migration and code partitioning in a vibrant setting. The application code is offloaded to the cloud in three steps. First, two versions of the application are being constructed. One is for local execution and the other is for remote execution. Today's smartphones use distinct instruction set architecture (Advanced RISC Machine, ARM) from PCs and servers. MAUI is intended to operate a distinct version of the application in a distinct setting, preferably without access to the source code of the program. Secondly, MAUI utilizes code reflection to indicate that a portion of the code is' remote' or not.

Clone cloud: Clone Cloud also uses virtual migration to offload part of the application to the cloud using Wi-Fi or 3 G network. Since the clone of the physical device is used for migration, it is not necessary to modify the application. Also, designers do not annotate techniques such as MAUI. Clone Cloud uses a ' cost model ' to analyze the cost to offload and running remotely comparable to local execution. The model was tested on Android phones, clones running on Ubuntu's Dell desktop. Speedups were reported up to 21.2 times with Wi-Fi over 3G.

Mobi Cloud: Mobi Cloud model proposes a safe use of cloud computing technology for MANETs (Mobile Ad Hoc NET works). Mobi Cloud can move MANETs to a service-oriented framework. Every mobile phone should be a service provider or service broker in Mobi Cloud. These service nodes are regarded as a virtualized component and reflected in the cloud. The Extended Semi-Shadow Images (ESSIs) are not the same as the virtual machine migration because the service nodes can be a clone, a partial clone, or just an image of the real device that can extend some functionality.

Mobile agent: Scavenger is another agent-based framework that uses Wi-Fi connectivity to use cyber foraging. It utilizes a mobile code strategy to spread employment to replace mobile devices. It presents a cost evaluation scheduler. The price evaluation is based on network velocity and a benchmarking technique is used to do this. Distributed jobs are performed in parallel on surrogate devices that enhance efficiency. It does not, however, provide any technique for a tolerance of fault. Since Scavenger only offloads jobs on the surrogate and doesn't share execution, it isn't dynamic.

4. DISCUSSION

Recent research has concentrated on virtual machine migration or mobile code as an offloading method. Cuckoo other frameworks do not use client-server architecture without Hyrax, Virtual Cloud. Although virtual machine migration and mobile code use older Hadoop and Ibis frameworks intended for distributed and grid computing, the techniques assist VM migration over standard client-server architecture. The reason is that Hadoop and Ibis have more benefits than RPC. Although client-server communication has well-supported APIs, it is not so robust in the case of a distributed mobile network. It needs pre-installed device services and disconnected activities are not developed. Considering the heterogeneity, the ongoing continuous connection and ad hoc nature of mobile computing researchers emphasize the frameworks of VM migration and mobile code.

Cloudlet, Clone Cloud, MAUI, Mobi Cloud use VM migration and reduce the developer's burden of rewriting 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 utilizes a mix of virtualization and partitioning of software. Again, for device mobility, the VM migration can sometimes be proven to be heavyweight, so Scavenger utilizes the mobile agent to offload devices to a surrogate in a vibrant setting.

Applications of MCC

- Augmented reality
- Mobile healthcare
- Mobile commerce
- Mobile learning
- Mobile gaming
- Web applications

5. CONCLUSION

Since mobile devices are becoming more potent through MCC, more studies can be conducted to discover a useful approach. The mobile cloud can alter the technology trend and bring an enormous improvement to our daily life by solving many real-life issues. We revealed the mobile cloud computing in this article. We illustrated the methods of offloading decision making and offloading to create the best use of MCC. Later, we outlined the application models that can be the future mobile apps and can alter the perspective of today's implementation.

REFERENCES

[1] https://www.marketsandmarkets.com/Market-Reports/mobileapplications-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) Crowd sourcing 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, Wether all 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 Gener Comput Syst 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 ACMSIGMOBILE Mob Comput Commun Rev 2: 19-26.

[23] Marinelli EE (2009) Hyrax: cloud computing on mobile devices using Map Reduce, 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 Gener Comput Syst 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. JMobile 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 hand held devices: a partition scheme. Proceedings of the 2001international conference on Compilers, architecture, and synthesis for embedded systems.

