Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.199

IJCSMC, Vol. 8, Issue. 5, May 2019, pg.219 – 233

Implementing a Mobile Cloud Computing Framework with an Optimized Computational Offloading Algorithm

Raji, Babatunde O.[1], Ajayi, Olusola O.[2]

[1,2]Department of Computer Science, Adekunle Ajasin University Akungba-Akoko, Ondo State, Nigeria
[1] rajahkonceptz001@gmail.com; [2] olusola.ajayi@aaua.edu.ng

Abstract— Most of the challenges of mobile cloud computing stemmed from the characters of mobile devices and wireless networks and their own restriction and limitation. All these challenges make application more complicated than on the fixed cloud devices. The entire limitations of mobile devices, quality of wireless communication and support from cloud computing to mobile are all important factors that affect accessing from cloud computing. So far, industrial and scientific communities have been doing various researches for responding to the above challenges. However, the open research issues are still focusing on identifying the method on how to provide suitable and friendly interactive services for mobile devices in mobile cloud computing world while considering their limitations. To this end, this study will focus on implementing an optimized computational offloading algorithm for mobile cloud computing. The need for friendly mobile users' experience when interacting with the cloud which include partitioning application such that parts that need more computation run on the cloud and remaining parts which is associated with the user interface run on the mobile device, is been considered in this work.

Keywords— cloud computing, mobile cloud, mobile cloud computing, computational offloading, algorithm

I. INTRODUCTION

Many authors, organizations and developers, based on their own perspective have come up with different definitions on what cloud computing is. However, according to [1], "Cloud computing is a parallel and distributed computing system, which is combined by a group of virtual machines with internal links. Such systems dynamically offer computing resources from service providers to customers according to their Service Level Agreement (SLA)." [2] introduced that the major function of a cloud computing system is storing data on the cloud servers and uses of cache memory technology in the client to fetch the data, those clients can be PCs, laptops, smartphones and so on.
The goal of cloud computing is to allow users to take benefits from all these technologies without the need for deep knowledge about or expertise with each one of them. The cloud aims to cut costs, and helps the users focus on their core business instead of being impeded by IT (Information Technology) obstacles [3]. The main enabling technology for cloud computing is virtualization. Virtualization

© 2019, IJCSMC All Rights Reserved

software separates a physical computing device into one or more "virtual" devices, each of which can be easily used and managed to perform computing tasks.

Mobility has become a very popular word and rapidly increasing part in today's computing area. An incredible growth has appeared in the development of mobile devices such as, smartphone, PDA, GPS Navigation and laptops with a variety of mobile computing, networking and security technologies. In addition, with the development of wireless technology like WiMax, Ad Hoc Network and WIFI, users may be surfing the Internet much easier but not limited by the cables as before. Thus, those mobile devices have been accepted by more and more people as their first choice of working and entertainment in their daily lives [4].

Mobile computing is based on a collection of three major concepts: hardware, software and communication. The concepts of hardware can be considered as mobile devices, such as smartphone and laptop, or their mobile components. Software of mobile computing is the numerous mobile applications in the devices, such as the mobile browser, anti-virus software and games. The communication issue includes the infrastructure of mobile networks, protocols and data delivery in their use [4].

Nowadays, both hardware and software of mobile devices get greater improvement than before, some smartphones such as iPhone 4S, Android serials, Windows Mobile serials and Blackberry, are no longer just traditional mobile phones with conversation, SMS, Email and website browser, but are daily necessities to users. Meanwhile, those smartphones include various sensing modules like navigation, optics, gravity, orientation, and so on. Which brings a convenient and intelligent mobile experience to users. In 2010, Google CEO Eric Schmidt described mobile cloud computing in an interview that 'based on cloud computing service development, mobile phones will become increasingly complicated, and evolve to a portable super computer' [5]. In the face of various mobile cloud services provided by Microsoft, Apple, Google, HTC, and so on, users may be confused about what mobile cloud computing exactly is, and what its features are.

Mobile cloud computing is a development of mobile computing, and an extension to cloud computing. In mobile cloud computing, the previous mobile device-based intensive computing, data storage and mass information processing have been transferred to 'cloud' and thus the requirements of mobile devices in computing capability and resources have been reduced, so the developing, running, deploying and using mode of mobile applications have been totally changed. On the other hand, the terminals which people used to access and acquire cloud services are suitable for mobile devices like smartphone, PDA, Tablet, and iPad but not restricted to fixed devices (such as PC), which reflects the advantages and original intention of cloud computing. Therefore, from both aspects of mobile computing and cloud computing, the mobile cloud computing is a combination of the two technologies, a development of distributed, grid and centralized algorithms, and have broad prospects for application [4] .

Since, most of power consuming mobile computing tasks are now being allowed to be offloaded to the cloud, then the need for an optimized offloading is crucial, in which the aim of this topic is to implement a mobile cloud computing framework with an optimized computation offloading algorithm.

During the 1960s, the initial concepts of time-sharing became popularized via RJE (Remote Job Entry); [6] this terminology was mostly associated with large vendors such as IBM and DEC. Full-time-sharing solutions were available by the early 1970s on such platforms as Multics (on GE hardware), Cambridge CTSS, and the earliest UNIX ports (on DEC hardware). Yet, the "data center" model where users submitted jobs to operators to run on IBM mainframes was overwhelmingly predominant.

In the 1990s, telecommunications companies, who previously offered primarily dedicated point-to-point data circuits, began offering virtual private network (VPN) services with comparable quality of service, but at a lower cost. By switching traffic as they saw fit to balance server use, they could use overall network bandwidth more effectively. They began to use the cloud symbol to denote the demarcation point between what the providers was responsible for and what users were responsible for. Cloud computing extended this boundary to cover all servers as well as the network infrastructure.

As computers became more diffused, scientists and technologists explored ways to make large-scale computing power available to more users through time-sharing. They experimented with algorithms to optimize the infrastructure, platform, and applications to prioritize CPUs and increase efficiency for end users [7].

© 2019, IJCSMC All Rights Reserved

Since 2000, cloud computing has come into existence. In August 2006, Amazon introduced its Elastic Compute Cloud, in April 2008, Google released Google App Engine in beta, in early 2008, NASA's OpenNebula, enhanced in the RESERVOIR European Commission-funded project, became the first open-source software for deploying private and hybrid clouds, and for the federation of clouds [8] .By mid-2008, Gartner saw an opportunity for cloud computing "to shape the relationship among consumers of IT(Information Technology) services, those who use IT services and those who sell them" and observed that "organizations are switching from company-owned hardware and software assets to per-use service-based models" so that the "projected shift to computing ... will result in dramatic growth in IT products in some areas and significant reductions in other areas." [9].

In February 2010, Microsoft released Microsoft Azure, which was announced in October 2008. In July 2010, Rackspace Hosting and NASA jointly launched an open-source cloud-software initiative known as OpenStack. The OpenStack project intended to help organizations offering cloud-computing services running on standard hardware. The early code came from NASA's Nebula platform as well as from Rackspace's Cloud Files platform. As an open source offering and along with other open-source solutions such as CloudStack, Ganeti and OpenNebula, it has attracted attention by several key communities. Several studies aim at comparing these open sources offerings based on a set of criteria [10].

On March 1, 2011, IBM announced the IBM SmartCloud framework to support Smarter Planet. Among the various components of the Smarter Computing foundation, cloud computing is a critical part. On June 7, 2012, Oracle announced the Oracle Cloud. This cloud offering is poised to be the first to provide users with access to an integrated set of IT solutions, including the Applications (SaaS), Platform (PaaS), and Infrastructure (IaaS) layers. In May 2012, Google Compute Engine was released in preview, before being rolled out into General Availability in December 2013.

However, due to limitation of mobile device in terms of battery life and processing power, Mobile Cloud Computing (MCC) has become an attractive choice to leverage this shortcoming as the mobile computation could be offloaded to the cloud, which is so-called mobile computation offloading, in which cloud computing provides services to mobile clients in forms of processing and storage without the need for them to install hardware on their side [11].

Recently, user preferences for computing have changed because of the latest developments and enhancements in mobile computing technologies. Several reports and studies have presented the importance of MCC and its impact on mobile clients and enterprises. For instance, and according to a recent study by ABI Research, more than 240 million business will use cloud services through mobile devices by 2015 and this will push the revenue of the MCC to $5.2 billion

Moreover, the usage of smartphones has increased rapidly in various domains, including enterprise, management of information systems, gaming, e-learning, entertainment, gaming, and health care. Although the predictions that mobile devices will be dominating the future computing devices, mobile devices along with their applications are still restricted by some limitations such as the battery life, processor potential, and the memory capacity of the SMDs [12]. Nowadays, modern mobile devices have sufficient resources such as fast processors, large memory, and sharp screens. However, it is still not enough to help with computing intensive tasks such as natural language processing, image recognition, and decision making. Mobile devices provide less computational power comparing to server computers or regular desktops and computation-intensive tasks put heavy loads on battery power.

Currently, there are several works and research in cloud computing that aim at enhancing the computing capabilities of resource constrained mobile client devices by providing mobile clients access to cloud infrastructures, software, and computing services. For example, Amazon web services are used to protect and save clients' personal data via their Simple Storage Service (S3) [13]. In addition, there are several frameworks that allow to process data intensive tasks remotely on cloud servers. For instance, the ASM computation offloading framework [14] showed that computation offloading helped to reduce the energy consumption cost of mobile devices by 33%, and the turnaround time of the application by 45% [15].In which the purpose of this study is to classify and analyse current computation offloading frameworks, identify their approaches and crucial issues and then work on the issues that require more attentions.

© 2019, IJCSMC All Rights Reserved

II. RELATED WORK

Although there are several offloading frameworks that have been implemented for offloading computation intensive components of mobile applications to the cloud, [15] classify these mechanisms into two broad categories:

i. Frameworks based on virtual machine cloning.
ii. Frameworks based on code offloading.

Frameworks based on code offloading offload intensive application components by invoking a remote procedure call (RPC) using annotations, special compilation or binary modification, whereas in virtual machine cloning, the mobile device's full image is captured and stored on the cloud server. During offloading, the mobile's execution is suspended and transferred to the VM clone in the cloud.

From the literature reviewed, we present the following related works on the subject of study. For each of the frameworks we identify the approaches used in the three steps introduced in the previous section under methodology.

[14] presented CloneCloud that supports a flexible application partitioner to empower the unmodified mobile applications to seamlessly offload the compute-intensive partitions to the trusted remote cloud. The system supports dynamic profiling and static analysis to partition the mobile application. The main goal of the partitioning is to optimize the energy usage and the execution time. Unlike its counterparts, the CloneCloud partitions the application at thread level. Although the CloneCloud provides the dynamic automatic application partitioning and user-transparent clone migration, CloneCloud needs installation of compatible clone VM along with the application on CloneCloud. The requirements of compatible clone VM and the application installation obstruct the seamless application execution, if the clone VM and the required application is not already installed.

[16] suggested a VM based cloudlet framework. Offloading to the cloud is not always a solution, because of the high WAN latencies, mainly for applications with real-time restrictions. Thus the cloud has to be moved closer to the mobile user in the form of cloudlets. A cloudlet can be defined as a hosting environment for offloaded tasks that is deployed to remote resources, as different as individual servers or parallel systems. Cloudlets are virtual-machine (VM) based on support scalability, mobility, and elasticity. They are located in single-hop close to mobile devices. In the preparation step, the framework requires the cloning of the mobile device application processing environment to a remote host. It offloads the entire application using VM as the offloading mechanism and more precisely it uses a technique called dynamic VM synthesis. The VM would encapsulate and separate the guest software from the cloudlet's host software. The mobile device serves as a thin client providing only the user interface, whereas the actual application processing is performed on the cloudlet infrastructure. Device mobility is the main critical issue for mobile users on the move while connected to cloudlets.

[17] proposed MAUI as an energy-aware fine grained, method level mobile application offloading mechanism that supports a semi-dynamic partitioning; wherein, programmers annotate an application with a considerably less effort. The profiler component of the MAUI assesses a method on call time for energy saving; whereas, the solver component takes the decision of migrating the method based on the input from the MAUI profiler. MAUI employs a time-out mechanism to detect the failures in connection with the server. Although MAUI significantly improves the battery life of a mobile device, it does not address scalability, QoS, and transmission latency.

[18] presented the mirror server framework that uses Telecommunication Service Provider (TSP) based remote services. A TSP is a type of communication service provider which provides voice communication services such as landline telephone services. Mirror server extends capabilities of smartphones by providing three different types of services: computation offloading, security, and storage. Mirror server is a powerful server which retains VM templates for different mobile device platforms. This framework does not require a partitioning as the entire application is offloaded. In the preparation step, a new VM instance is created. This VM is called mobile mirror and the mirror server takes care of managing and deploying the mobile mirrors on a computing infrastructure in the telecom network. Applications are executed in the mirror VM instances and results are returned to the SMD. The framework employs an optimized mechanism for offloading.

[19] presented Cuckoo framework a dynamic offloading framework that takes the offloading decision at runtime and offloads the well-defined components of the application. This framework offloads

© 2019, IJCSMC All Rights Reserved

mobile device applications onto a cloud server using a Java stub model. Cuckoo's objectives were to enhance mobile's performance and reduce battery usage. The framework was integrated using Eclipse development tool with the open source Android framework taking advantage of the existing activity model in Android. The framework requires the developer to write offloadable methods twice - one for local computations and one for remote computations. For this purpose, a programming model is made available to application developers. This programming model is used for dropped connection, supports local and remote execution, and combines all codes in a single package so the user will have a compatible remote implementation.

[20] presented a semi-automatic offloading framework called Phone2Cloud. The objective was to improve energy efficiency of smartphones and improve the application's performance. Unlike the previous frameworks, authors focus on conducting a fully quantitative analysis on energy saving of the system by conducting application experiments and scenario experiments. In order to run applications on the cloud and receive the results, applications need to be manually modified during preparation step to make it possible to be executed on cloud servers. The offloading decision is based on a static analysis while considering user's delay-tolerance threshold. For delay tolerant applications, the framework uses a simple model to expect WiFi connectivity. The threshold is defined based on predictions to delay transfers in order to offload more data on WiFi while respecting the application's tolerance threshold. The framework will wait for WiFi (only if 4G savings are expected within the application's delay tolerance) to become available, rather than sending data immediately.

Sharing the same concern but from a different perspective, [21] presented Jade, a system that monitors application and device status and that automatically decides where the code should be executed. The goal of Jade was to maximize the benefits of energy-aware computation offloading for mobile applications while minimizing the burden on developers to build such an application. During partitioning, applications are partitioned at the class level in Jade based on the collected information. As a preparation, the system checks the application and device status by monitoring the communication costs, work load variation, and energy status. The framework provides a sophisticated programming model with a full set of APIs, so developers have total control on: how the application is partitioned, and how remote code interacts with local code. The offloading decision is taken at runtime to decide where the code should be executed. Jade supports two types of servers:

I. Android servers and
II. Non-Android servers running operating systems such as Windows and Linux.

Non-Android servers must have Java installed in order to support Jade. Jade's runtime engine runs as a Java program on a non-Android server. Jade can dynamically change its offloading decision according to the device status and thus efficiently reduce energy consumption of mobile devices.

Having reviewed different existing computation offloading frameworks along with their main characteristics, Table 2.5 below presents an overall view about these frameworks and classifies them based on the following attributes:

- Preparation: Any necessary preparations before offloading.
- Partitioning: Partitioning supported or not.
- Decision: Dynamic or static.
- Offloading Mechanism: Mechanism used to offload intensive computations.
- Granularity Level: Granularity Level (i.e. class, method, thread).
- Annotation: Automation of partitioning process (Automatic or manual).
- Contribution: Solved problems?

III. METHODOLOG & SYSTEM DESIGN

A. METHODOLOGY

This study adopts the use of mobile computing methodology. For successful execution of this study, the following steps shall be followed:

i. **Application profiling:** In order to make correct offloading decisions, the first step entails getting updated information concerning the status of the application and the device. Application profiling is the process of collecting information about programs, such as energy consumption, data size, execution time, and memory usage. Similarly, device profiling collects information about devices status, such as battery level, CPU usage, and wireless connection.

ii. **Preparation:** The preparation step performs all actions required for both offloadable and non-offloadable components either to enable their use in mobile applications or for remote execution.

iii. **Partitioning**: This step divides the application into offloadable and non-offloadable components meaning which components to retain on the mobile device and which to migrate to the cloud server.

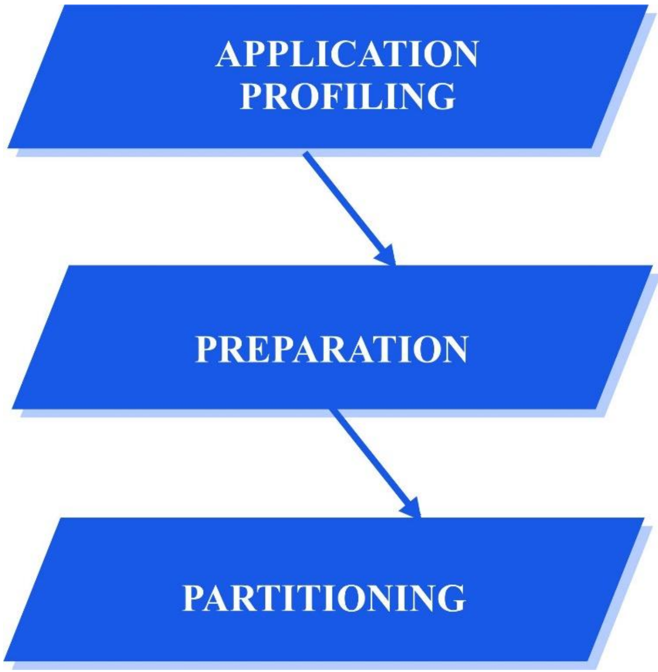


Figure 1: **Study Methodology**

B. SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development (Bentley *et al*, 2014). System design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user.

System Specification

The following specification will be discussed and highlighted in this section

i. I/O Specification
ii. Security Specification

I/O Specification

In computing, input/output or I/O is the communication between an information processing system, such as a computer, and the outside world, possibly a human or another information processing system. Inputs are the signals or data received by the system and outputs are the signals or data sent from it. The term can also be used as part of an action; to "perform I/O" is to perform an input or output operation.

Given an operation to translate from one language to another, the diagram below shows the input and the corresponding output.

© 2019, IJCSMC All Rights Reserved

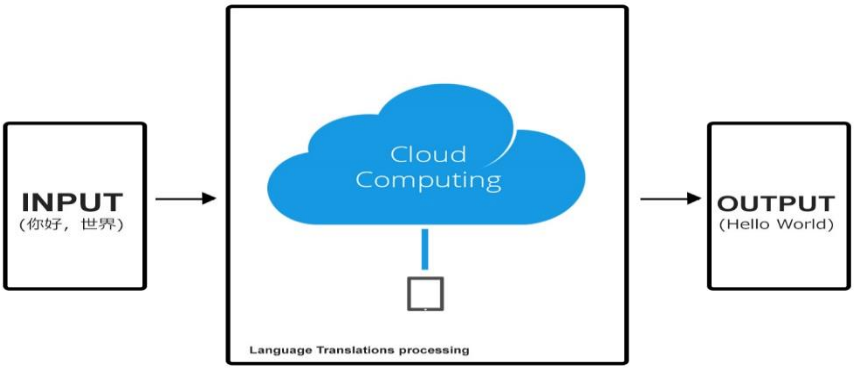


Figure 2 - I/O Specification

Security Specification

Security specification describes enhancements to data to provide quality of protection through integrity, confidentiality, and authentication. These mechanisms can be used to accommodate a wide variety of security models and encryption technologies. In mobile cloud computing, a token authentication is used, this mechanism uses a security token to validate the user and determine whether a client is valid in a particular context. A client can be an end user, machine, application, or import.

The diagram below shows the authentication process in mobile cloud computing.

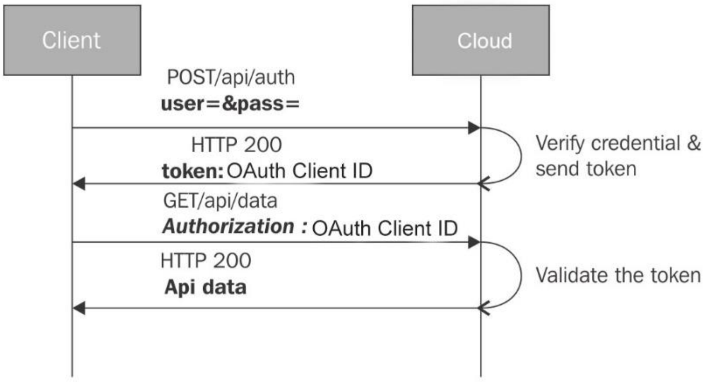


Figure 3 - Security Specification

System Organization (Operational) Structure

System Organization structure is the structure that depicts how the entire system is being set up, showing the hierarchy and how each component that makes up the system interacts to establish its pre-defined functionality.

The entire system will be structured using:

i. USE-CASE diagram
ii. Sequence diagram
iii. Activity diagram

USE-CASE Diagram

In the Unified Modelling Language (UML), a use case diagram is a diagram that summarizes the details of the system's users (also known as actors) and their interactions with the system. Use case diagrams serves as the blueprints for the system *Figure 4* shows the USE-CASE Diagram for the intended system implementation.

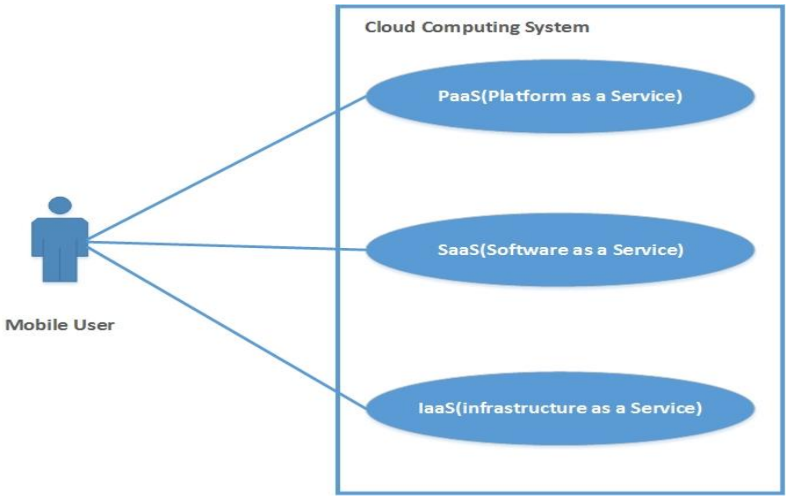


Figure 4: Use-Case Diagram for the intended system.

The Use-Case Diagram (Figure 4) given above shows the interaction of user to the cloud system, where user can perform and request three difference services, starting from PaaS(Platform as a service) where user (as a developer) can easily write and deploy application on cloud, the SaaS (Software as a Service), where user access the host applications provided by the cloud providers to the customers and also the IaaS (Infrastructure as a Service) where user can access the virtualized computing resources over the internet such as virtualization, servers, storage and networking.

Sequence Diagram

A sequence diagram, sometimes known as event diagrams or event scenarios is a type of interaction diagram that describes of how and in what order- a group objects works together. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. Figure 4.3.2 represents the Sequence Diagram for the proposed system.

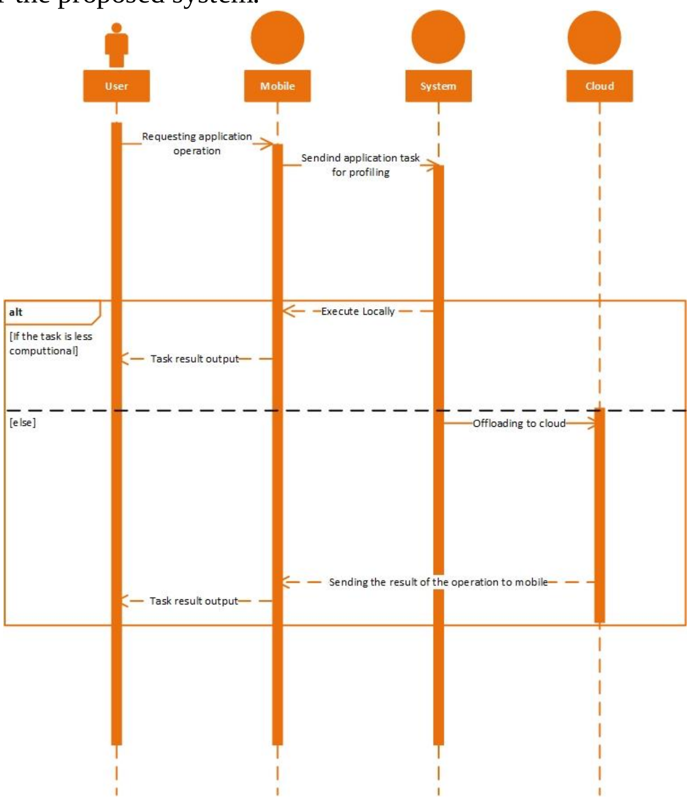


Figure 5 - Sequence Diagram for the intended system.

© 2019, IJCSMC All Rights Reserved

The Sequence diagram (*Figure 5*) given above describes how and in what order- a group objects which user, mobile and cloud that make up the system works together. The user interacts with the mobile by making a certain request, system operates based on the request made by the user, interact with the cloud only if the operations requires it and display the result of the operation for user to view on the mobile.

Activity Diagram
An activity diagram is essentially a flowchart that shows activities performed by a system. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities *Figure 6* represents the Sequence Diagram for the proposed system.

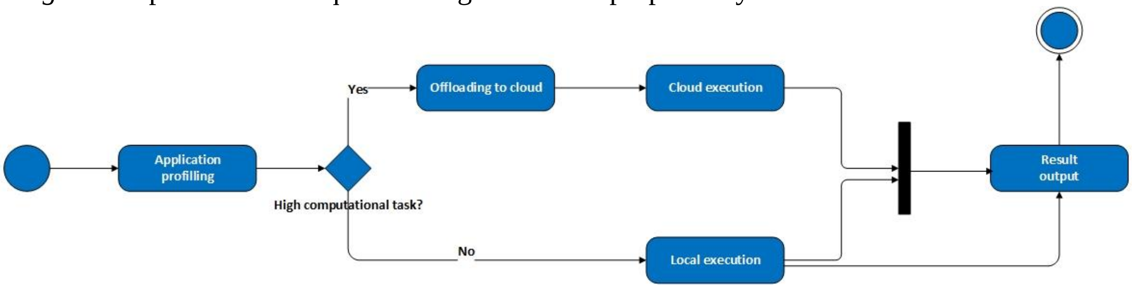


Figure 6 - Activity Diagram for the intended system.

The Activity Diagram (Figure 6) given above shows the flowchart of each activity perform in the system, starting from the operation profiling, offloading to cloud for execution or local execution based on the result criteria and the result of the profiling to the last activity of the system which is result output.

IV. IMPLEMENTATION

The application starts and presents the user with a splash screen as shown in the figure below. The splash screen which can also be referred to as welcome screen which contains a little introductory screen. The splash screen automatically discards itself after 3 seconds.

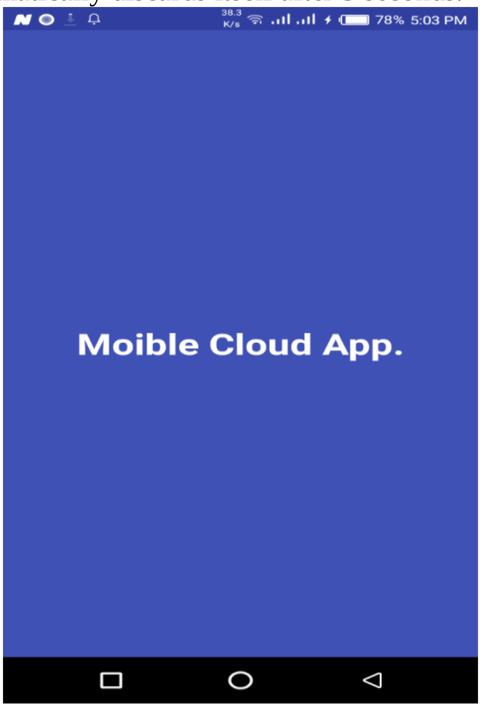


Figure 7 – Splash screens

Main Interface: After the spashscreen, it moves to the Main Activity, as show in the diagram below, the main interface display a floating action button for user to select image.

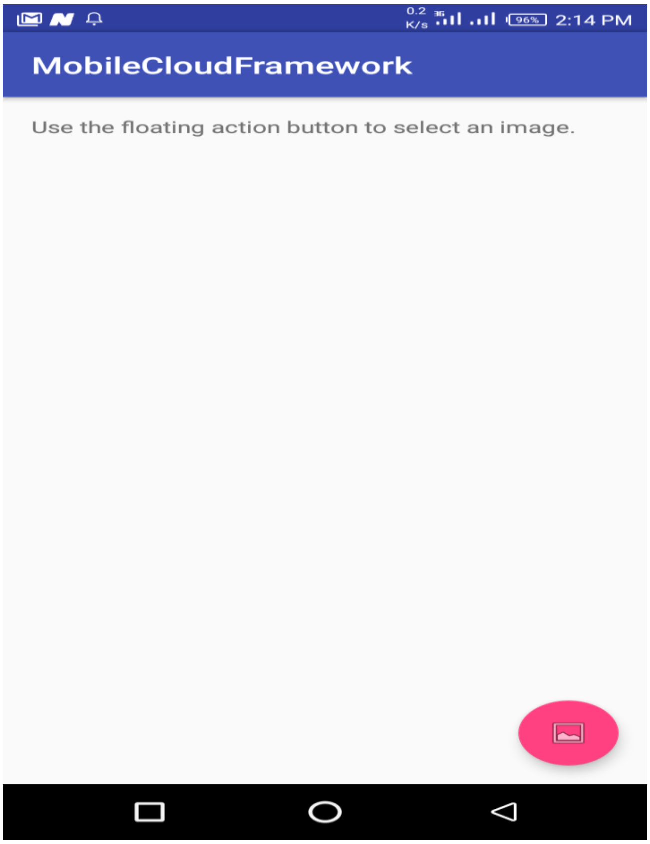


Figure 8 – Main Activity

Once the floating action button is clicked, a pop menu is shown for user to use the camera or access existing image in the mobile phone.

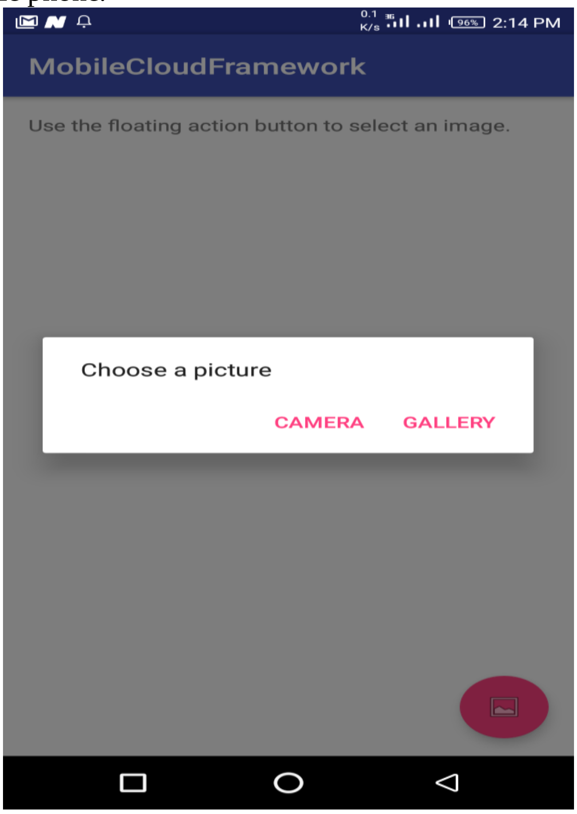


Figure 9 – Main Activity (Pup-up Menu)

Once any of the action is triggered, for instance “CAMERA”, a camera interface is show for user to snap picture as shown below

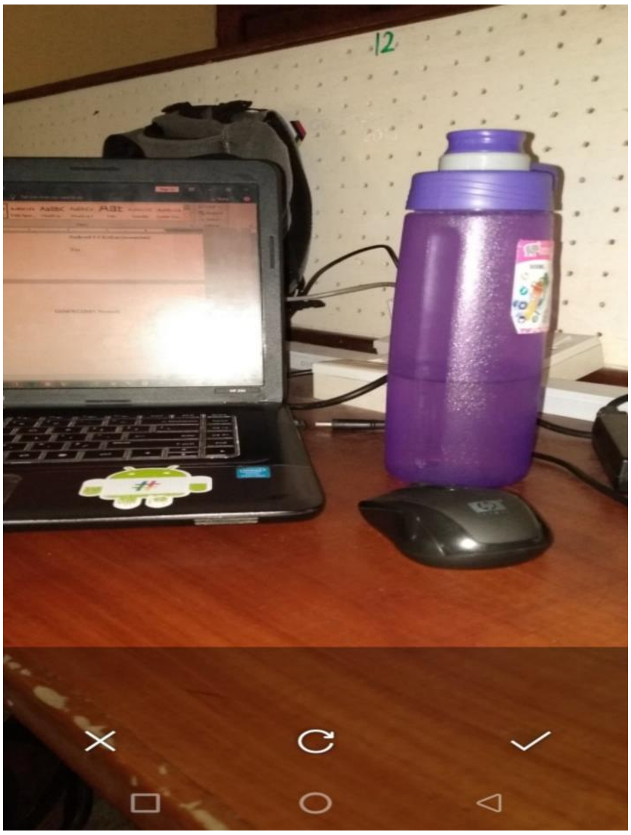


Figure 10– Main Activity (Camera Interface)

After the image is captured, the image is directly offloaded to the cloud for processing while waiting for results as show in the diagram below:

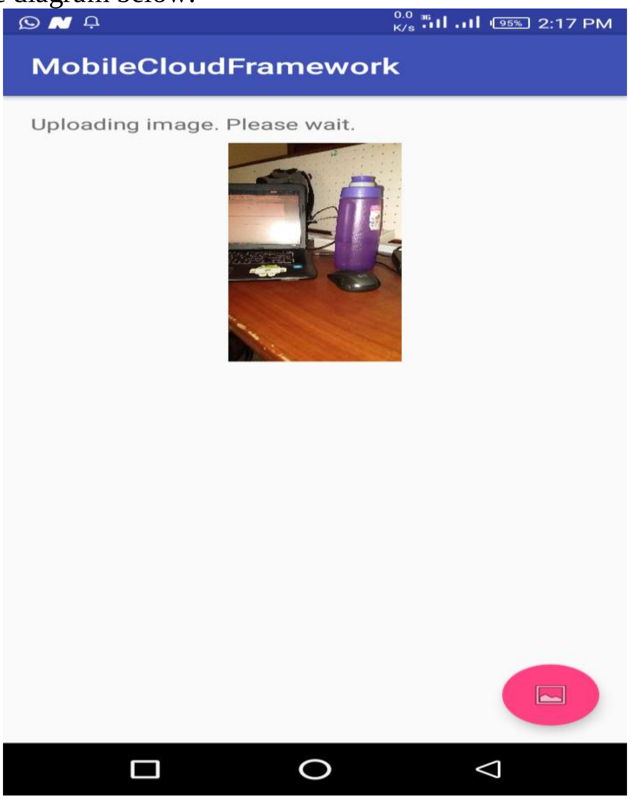


Figure 11 – Main Activity (with Image for processing)

After the operation the result (objects in the image) of the input (Images) is displayed as shown in the diagram below:

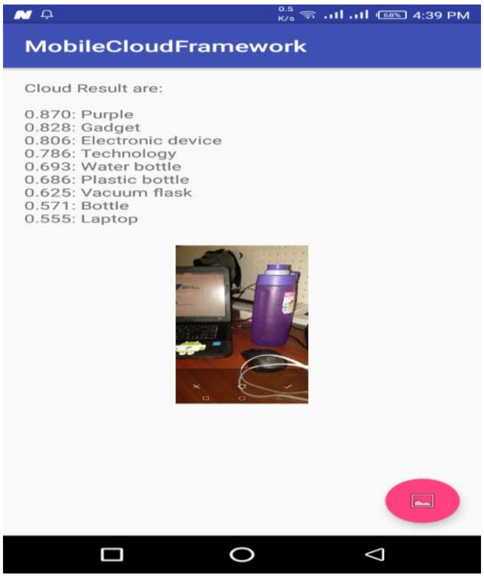


Figure 12– Main Activity (with results)

As shown in the above description the purpose of the operation is to return the object(s) found in the image uploaded to perform mobile cloud computing task.

Testing

System testing is another step-in system development. It is always necessary to test a program to ascertain whether it not it is able to produce correct result and that it conforms to the content of the requirement specification document and it is without errors. It involves the testing and running of the integrated program in their intended sequence using sample data. The sample data are selected so as to test various functions and sub-routines of the programs. System testing can be done through some stages. The first stage is called unit testing or component testing and this testing is done during the development of the system. Each component, class or function test isolates from other component or unit by checking the input and output for it. The second stage if called integration testing. The integration between components will be tested and in case there are any errors the components will be tested again. In testing the system, the authors make use of both valid and invalid data to test for the robustness of the system as well as the stability. In addition, the following testing method were carried out.

Unit and Integration Testing

Unit testing focuses on testing module, it is a software testing method by which individual units of source code, usage procedures, and operating procedures, are tested to determine whether they are fit for use. For example, unit testing was carried out on each function such as the modules which performs mutation, selection and cross over in our developed system. All this function is developed to perform different functions, so we need to test if it solves the task for which it was developed without errors.

After unit testing has been done with satisfaction for each component, the integration testing is started to ensure there is proper interaction between the system modules, that is, there is no conflict with the communication between one module to another, for example, the interaction between the log in module and the main interface module, integration testing ensures that there is no conflict in the data declaration in different modules. For the developed system to work effectively, an integration test was carried out between the selection, crossover and mutation functions, though is modules performs different functions, but for our developed system to achieve it aim, there must be adequate interaction between the modules, before genetic algorithm can function effectively.

System and Acceptance Testing

System Testing is a testing that is integrated in a product for total assurance that all requirements are met, while Acceptance Testing also known as User acceptance testing (UAT) is the last phase of the

© 2019, IJCSMC All Rights Reserved

software testing process. During UAT, actual software users test the software to make sure it can handle required tasks in real-world scenarios, according to specifications. UAT is one of the final and critical software project procedures that must occur before newly developed software is rolled out to the market.

The table below shows the result of all the tests carried out in this study:

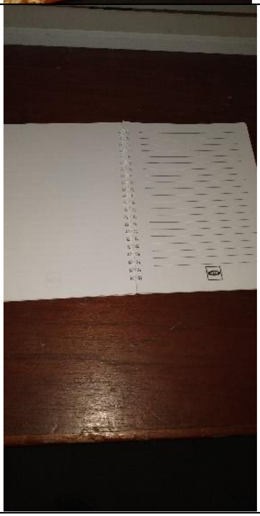
Test Nos	**Inputs**	**Response Time (Milliseconds)**	**Expected results**	**Actual results**
1		18894	Flower, Leaf	Flower, Plant, Shrub, Leaf, Flowering Plant, Groundcover, Tree, Plant Community, Arctostaphylos
2		11576	Mobile Phone, Device, Gadget	Mobile phone, Portable communication device, Gadget, Communication device, Feature phone, Electronics device, Technology, Smartphone.
3		6597	Book, White Paper, Notebook.	Text, paper, Material property, font, Document, Wood, Paper product, Book, Stationery.

Table 2 - Testing

© 2019, IJCSMC All Rights Reserved

V. CONCLUSIONS

With the high increasing of data computation in commerce and science, the capacity of data processing has been considered as a strategic resource in many countries. Mobile cloud computing (MCC), as a development and extension of mobile computing (MC) and cloud computing (CC), has inherited the high mobility and scalability, and become a hot research topic in recent years. We conclude that there are three main optimization approaches in MCC, which are focusing on the limitations of mobile devices, quality of communication, and division of applications services. Firstly, using virtualization and image technology can address it effectively, and immigrate task from terminal to cloud is also a good way to achieve better results. Secondly, as we know the quality of communication in wired network is better than in wireless network, so reducing the proportion of data delivery in wireless environment is an effective way to improve the quality. In addition, upgrading bandwidth is envisaged to be a simple way to increase performance but it incurs additional cost to users. Deploying an effective elastic application division mechanism is deemed to be the best solution to guarantee the application service in MCC; its complicated, but promising high impact results.

ACKNOWLEDGEMENT

This research work was supported by National Association of Science Student (NASS), Adekunle Ajasin University (AAUA) Chapter.

REFERENCES

[1] BUYYA, R., YEO, C., AND VENUGOPAL S. (2008), "*MARKET-ORIENTED CLOUD COMPUTING: VISION, HYPE, AND REALITY FOR DELIVERING IT SERVICES AS COMPUTING UTILITIES*," IN HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS, 2008. HPCC'08. 10TH IEEE INTERNATIONAL CONFERENCE ON. IEEE, PP. 5–13.

[2] Hewitt, C. (2008) "*Orgs for scalable, robust, privacy-friendly client cloud computing*," Internet Computing, IEEE, vol. 12, no. 5, pp. 96–99.

[3] HAMDAQA, Mohammad. (2012) *Cloud Computing Uncovered: A Research Landscape* (PDF). Elsevier Press, pp. 41–85.

[4] Qi, Han, G., Abdullah (2012) "*Research on mobile cloud computing: Review, trend and perspectives*", 2012 2nd International Conference on Digital Information and Communication Technology and its Applications, DICTAP, pp. 195-202.

[5] Marrapese, B. (2010, Dec.) *Google ceo: a few years later, the mobile phone becomes a super computer*, 2010.

[6] White, J.E. (2016) "*Network Specifications for Remote Job Entry and Remote Job Output Retrieval at UCSB*". tools.ietf.org. Retrieved 2016-03-21.

[7] Corbató, Fernando, J. (2012)"*An Experimental Time-Sharing System*". SJCC Proceedings. MIT.

[8] Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I.M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M. Emmerich, W., Galan, F. (2009) "*The Reservoir model and architecture for open federated cloud computing". IBM Journal of Research and Development. JRD.*

[9] Cearley D. *et al* (2009). *Hype cycle for applications development gartner group reporter number* G00147982, Gartner.

[10] Kostantos, Konstantinos, *et al* (2013). "*OPEN-source IaaS fit for purpose: a comparison between OpenNebula and OpenStack." International Journal of Electronic Business Management*.

[11] Shamim, M., Sarker, S., Angona, N., Ali A., Md. (2015)"*A Review on Mobile Cloud Computing*". vol. 113, pp. 4-9.

[12] Smit, M., Shtern, M., Simmons, B., Litoiu, M. (2012) *Partitioning applications for hybrid and federated clouds, in: Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research,* IBM Corp., pp. 27–41.

[13] Mathew, S. (2014*) Overview of Amazon Web Services, Amazon Whitepapers*.

[14] Chun B., Ihm S., Maniatis P., Naik M., and Patti A. (2011) "*Clonecloud: Elastic execution between mobile device and cloud*," in Proceedings of the sixth conference on Computer systems. ACM, pp. 301–314.

[15] Akherfi, Khadija, G., Micheal, H., Hamid (2016)"*Mobile cloud computing for computation offloading: Issues and challenges*" in Applied Computing and Informatics., vol. 14 pp. 1-16.
[16] Satyanarayanan, M. ,Bahl, P. ,Caceres, R., Davies, N. (2009)*The case for vm-based cloudlets in mobile computing,* IEEE Pervasive Comput. 8 (4) (2009) 14–23.
[17] Cuervo, E., Balasubramanian, A., Cho, D.-K., Wolman, A.S. Saroiu, R. Chandra, P. Bahl, (2010) *Maui: making smartphones last longer with code offload, in: Proceedings of the 8th International Conference* on Mobile Systems, Applications, and Services, ACM, pp. 49–62.
[18] Zhao, B., Xu, Z., Chi, C., Zhu, S., Cao, G. (2010) *Mirroring smartphones for good: a feasibility study, in: Mobile and Ubiquitous Systems: Computing, Networking, and Services, Springer*, 2010, pp. 26– 38.
[19] Kemp, R., Palmer, N., Kielmann, T., Bal, H., Cuckoo: (2010*) a computation offloading framework for smartphones,* in: Mobile Computing, Applications, and Services, Springer, pp. 59–79.
[20] Xia, F., Ding, F., Li, J., Kong, X., Yang, L.T., Ma, J. (2014) Phone2cloud: *exploiting computation offloading for energy saving on smartphones in mobile cloud computing,* Inform. Syst. Front. 16 (1) (2014) 9–
[21] Qian, H., Andresen, D. (2015) *Jade: reducing energy consumption of android app,* Int. J. Network. Distrib. Comput (IJNDC) (3) (2015) 150–158 (Atlantis Press).

© 2019, IJCSMC All Rights Reserved