International Conference on Information Engineering, Management and Security 2016 [ICIEMS]

ISBN	978-81-929866-4-7
Website	iciems.in
Received	02 – February – 2016
Article ID	ICIEMS016

VOL	01
eMail	iciems@asdf.res.in
Accepted	15 - February – 2016
eAID	ICIEMS.2016.016

Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost

Long CAI[1], Kokula Krishna Hari Kunasekaran[2], Prithiv Rajan[3]
1 – Research Scholar, Hong Kong
2 – Secretary General, Association of Scientists, Developers and Faculties
3 – Global President, Techno Forum Group, Australia

Abstract: *The increased usage of mobile devices caused them to face a large amount of resource, memory and processing speed scarcity. Of all other constraints, energy is the major problem for this to carry out a task. The concept of offloading gets into play for mobile devices i.e., the task or the computation which needs to be performed involving more service in the android systems will be shifted to resourceful server (for ex cloud) and getting back the results done from the cloud. The decision of whether to offload a computation or not will depend on the task accounting to the energy spent by the device while working with the application versus the amount of energy spent by the same device for uploading the task to the cloud and getting the result back from the cloud. As this concept depicts energy is the major constraint for whether to offload a task or not, there is a model called CUCKOO framework which acts as an interface between the cloud and the android environment to support for the task offloading to the cloud. Thus this framework bridges the gap between the smartphones as well as the cloud environment so that computation intensive task can be performed with less amount of energy consumed. In this work two applications are used to detect the amount of energy consumed in the cloud as well as the smartphones namely eyedentify and Photoshoot.*

Keywords: *Mobile Cloud Computing, Cuckoo Framework, Offloading, eyeDentify and Photoshoot.*

1. INTRODUCTION

The cloud computing is playing a major role in Mobile Cloud Computing. The mobile cloud computing consist of a cloud server, provided by a variety of servers. It is the responsibility of cloud servers to be able to cope up with the type of requirements of the users [1]. The mobile environment is facing a lot of constraints namely memory, speed, energy and computation. These constraints are decided by the amount of service utilised by the mobile devices [2]. The task in the android systems can be of two types as computationally intensive or ordinary task. A task can be computationally intensive if it involves multimedia processing, GUI applications etc.

Task offloading is a critical technique because in some cases it increases the energy consumption of smartphones. This is due to the fact that it involves the computation as well communication overhead to perform a task. The task can be anything depending on the amount of service (energy) consumed. This technique can be four variants depending on the task and data involved in the particular application involved [3]. In the first case, the input data is available locally on the smartphone and task execution occurs on the smartphone as well. This case seems to be normal because there is no offloading [4]. The second case is where the task execution happens on the cloud but the task data exists locally on the smartphone. In this case, the smartphone needs to upload the task data to the cloud and then download the task results. The third scenario consists of the task execution is being performed locally on the smartphone, but the task data exists on the cloud. To perform the required execution, the smartphone is allowed to download the task data and perform the task execution locally. And in this final case, the input data needs to be injected is available on the cloud and task execution occurs

Cite this article as: Long CAI, Kokula Krishna Hari Kunasekaran, Prithiv Rajan. "Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost". *International Conference on Information Engineering, Management and Security 2016*: 83-91. Print.

in the cloud as well. Therefore, the work needed to be performed by the smartphone is just downloading the task results which are being carried out by the cloud.

The task offloading will offer a better service when it is coupled with a framework called CUCKOO model [5]. The Cuckoo framework, which simplifies the development of smartphone applications that benefit from computation offloading and it is able to decide upon offloading [6]. It provides a dynamic runtime system that can, at runtime, decide whether a part of an application will be executed locally or remotely.

1.2 Objective

Providing uninterrupted service to the smartphone is evitable because of its energy consumption and low resource. To enhance the handheld device to next dimension a hybrid platform called Mobile Cloud Computing is used to minimise energy consumption is introduced. The cuckoo framework is used to analyze the energy cost consumption for task offloading to the cloud. Dynamic offloading decisions based on cuckoo model are worked out is based on the equation 1 computation. The energy consideration for each of the cases is calculated.

ECuckoo <ELocal Execution ---> (1)
ECuckoo- Energy consumed by the smartphone using Cuckoo framework (With Offloading)
ELocal Execution-Energy consumed by the smartphone (Without Offloading)

1.3 Scope of the Work

The aim of the work is to reduce the amount of energy consumed by the individual tasks in the smartphones, So that the smartphones battery lifetime can be enhanced. Moore's law states that the integrated circuit is accommodating number of transistors every year twice that of the previous year. In contrast, battery capacity increases only by 5% every year. Annually the gap between the energy supply and demand increases by 4%. This concept offers a better conjunction between the battery life and the task handling. For a task performed locally, it involves the energy consumption based on the application. In my consideration, two tasks are used namely eyeDentify and Photoshoot.

2. Related Work

Bowen Zhou et al. and Amir Vahid Dastjerdi et al. [7] stated the objective of the context sensitive offloading scheme is to derive an optimal offloading decision under the context of the mobile device and cloud resources to provide better performance and less battery consumption. The proposed framework adopts client-server communication model, in which the cloud resources (e.g. mcc cloud, public and private cloud) used will be acting as servers and the mobile device can be the client to access the services on servers. The client side framework should be consist three components, namely a context monitor, a communication manager and a decision engine. Next, the server side includes server side communication manager, a program profiler and a task manager. The cost model consists of three parts, namely the task execution time denoted by D, wireless channel energy consumption denoted by E and monetary cost denoted by M when related. Then the total cost of executing task ti is as follows:

$P(ti) = \alpha 1 * D(ti) + \alpha 2 * \rho d * E(ti)$ ->(2)

In the context of cloud computing according to Antti P. Miettinen et al. and Jukka K. Nurminen et al. [8], in energy trade-off analysis as given in equation 2, the critical aspect for mobile clients is the trade-off between energy consumed by computation and the energy consumed by communication. Bowen Zhou et el considered the energy cost for performing the local computation (Elocal) versus the cost of transferring the computation input and output data (Ecloud) as provided by K. Yang, K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava et al S. Ou, and H.-H. Chen in "A Survey of Computation Offloading for Mobile Systems," [2012] [9]. If D is the amount of data to be transferred in bytes and C is the computational requirement for the workload in CPU cycles then

Ecloud =D/Deff -> (3)
Elocal =C /Ceff -> (4)

Where Deff and Ceff are device specific data transfer and computing efficiencies. In equation 3, Deff is a parameter meant for measuring the amount of data that can be transmitted with given energy (in bytes per joule) whereas in equation 4, Ceff is a parameter used as a measure for the amount of computation that can be performed with given energy. By making use of this, he derived the relationship between computing and communication for offloading to be beneficial.

The basic idea of COSMOS given by Cong Shi et al, Karim Habak et al, Pranesh Pandurangan et al [10] is to achieve good offloading performance at low monetary cost by sharing cloud resources among mobile devices. Specifically, in this paper our goal is to minimize

Cite this article as: Long CAI, Kokula Krishna Hari Kunasekaran, Prithiv Rajan. "Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost". *International Conference on Information Engineering, Management and Security 2016*: 83-91. Print.

the usage cost of cloud resources under the constraint that the speedup of using COSMOS against local execution is larger than $1-\delta$ of the maximal speedup that it can achieve using the same cloud service, where $\delta \in (0, 1)$. This process consists of mainly three components: a COSMOS Master running on a VM instance that manages cloud resources and exchanges information with mobile devices; a set of active COSMOS Servers each of which runs on a VM instance and executes offloaded tasks; and a COSMOS Client on each mobile device that monitors application execution and network connectivity and makes offloading decisions.

Eemil Lagerspetz and Sasu Tarkoma stated the main purpose of mobile cloud computing (MCC) [11] is to enable the development of computational intensive mobile applications by leveraging the application processing services of computational clouds. Contemporary distributed application processing frame works use run time partitioning of elastic applications in which additional computing resources are occurred in runtime application profiling and partitioning. There are provisions for enabling computationally intensive mobile applications to have software level technique called Distributed application processing on SMDs. A number of augmentation algorithms have been proposed for alleviating the resources limitations of SMDs — energy augmentation, memory augmentation (G. Bianchi, 2005 [12]), and an application processing augmentation. The elastic applications allows for separating intensive components of the mobile applications in the currents APAs.

In Graph-based application partitioning algorithms the elements of graph — vertices and edges — are used to represent the parameter or context of an application (Xinwen Zhang, Won Jeon, Simon Gibbs, and Anugeetha Kunjithapatham., 2010) [15]. The parameters can be available resources, data size, communication overhead, computation cost, and memory cost depending on the SMDs used and also the type application being used. APAs adopt the graph model for modelling execution states, cost models, internal dependency, data flow, and control flow. In graph-based APAs, obtaining the optimal partitioning decision is a Non-deterministic Polynomial-Complete (NPC) problem. Linear programming (LP) is a mathematical method for determining away to achieve the best result such as maximum profit or lowest cost with a list of requirements which are represented as a linear equation in a mathematical model. Guetal et al, Abebeand Ryan and Verbelen et al. [13] (2013) devised a hybrid partitioning algorithm. In this algorithm, a method for the allocation of components to server in the computational cloud while minimizing required bandwidth. This work is further extended by including dynamic runtime adaptation to the frame- work and a programming model based on annotations which aim to minimize the programmers' burden.

In the following, Majid Altamimi et al and K. Naik "A Survey of Software Based Energy Saving Methodologies for Handheld Wireless Communication Devices [2010]" [14] modelled the energy usage in two distinct cases, namely, file upload and file download. For simplicity, he assumed that the mobile device transceiver uses only two power levels, namely, PRX when it is idle, in back off mode, or receiving and PTX when it is transmitting. For File Download Case, the mobile device is mostly receiving. Here, we address first the general situation where there is no limitation on the file download rate from the cloud. Next, we address the situation where the cloud restricts the file download rate. For every MAC frame to be received, the mobile device has to send a CTS and an ACK frame. The mobile device has to send a TCP ACK for every received TCP segment. During downloading a file, a smartphone will be receiving a data frame for a time T +3SIFS+TPHY +TRTS and it has to wait for the AP back off time ϑ/t.

3. Existing System

Existing system consists of two major parts, smartphones (i.e., user equipment, UE) and Cloud Computing (CC), both linked to the Internet. The smartphones are connected to the Internet through a WLAN access point or a cellular data network base station. These smartphones provide all of mobile computing functionalities to the end users via different applications. On the other hand, the CC part consists of cloud data center and cloud provider, which are accessible through the Internet. The cloud supports the end users (e.g., smartphone users) by providing all the CC functionalities that are needed for mobile computing. In the offloading technique, smartphones access the cloud via the Internet. Therefore, offloading is considered as a Network Related Application (NRA). At the beginning of studying NRA, network interfaces (i.e., 3G/4G and WLAN) should be considered because each of these interfaces has its own characteristics, such as supported data rate. As a result, each network interface consumes unequal amount of energy. In addition, the Internet protocols, namely, the Hypertext Transfer Protocol (HTTP) and the File Transfer Protocol (FTP) need to be taken into account. The network interfaces and protocols are the major factors that affect the energy costs of task offloading. They are taken into account for energy cost modelling.

3.1 The Experiment

They experimentally evaluate the energy cost on smartphones when the offloading technique is used over different network interfaces and Internet protocols. They conducted experiments in four broad experimental scenarios related to the location of the task data. The first scenario corresponds to S1, where there is a local task execution and the task data exists on the smartphone. The second scenario corresponds to S2, where uploading the task data, doing the task computation (encoding) by the cloud, and downloading the task result is presented by the "Upload + CC encoding + Download". The third scenario corresponds to S3, where there is a local task execution and the task data is downloaded from the cloud, as shown by the "Download + Local encoding". The fourth scenario corresponds to S4, where the task data exists in the cloud and the task executed on the cloud, and the task result is simply

Cite this article as: Long CAI, Kokula Krishna Hari Kunasekaran, Prithiv Rajan. "Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost". *International Conference on Information Engineering, Management and Security 2016*: 83-91. Print.

downloaded, as presented by the "CC encoding + Download". For uploading and downloading files to and from the cloud, we consider the energy implications of: (i) using the HTTP and FTP protocols at the application level; and (ii) using the 3G and WLAN communications at the wireless interface level.

4. Proposed System

The proposed system is defined to work in a Mobile Cloud Computing environment as briefed out earlier. The cloud environment is able to serve the variety of requests from the users based on the needs of them. Likewise the mobile device is able to serve the users by installing the applications specific to the needs of the users.

4.1 The Cloud Modelling

Even though the cloud environment is able to support all kinds of users, it is based on pay as you go based service model. But we need little consideration before taking the smartphones to the cloud. The general problems of the smartphones include energy, speed and the memory. For all these considerations the cloud environment is brought into these devices to minimize the power consumed. The service provider provides provision for the cloud computing service model, giving out a very large amount of high performance computing resources and high-capacity storage devices that are shared among end users as required. There can be many kinds of cloud service models. Depending on the end users subscribing to the service may or may not have their data hosted by the service, and also computing resources allocated on demand from the pool. The service providers can also be giving service in the form of software applications offering as required by the end user. In order to be successful, the high speed network deserves a major criterion for an offloading to the cloud service model to provide connection between the end user and the service provider's infrastructure.

The run time computational offloading involves deployment of a delegated application on the virtual machine of the cloud server node and is a challenging aspect for this process. Current COF focus on partitioning an elastic mobile application dynamically and offloading the intensive partitions at runtime. The reconfiguration of delegated applications on the virtual device instance of the machine is a critical feature of current COFs on the cloud server node. Therefore, the execution of the offloaded applications to the cloud server nodes requires the deployment of virtual phone instance(s) on the virtual machine of the cloud data center. The operating system platforms implement platform-specific application frameworks; due to the hardware architecture and operating system platform of the mobile devices are different. This interface is deployed in the ADSL layer of the android devices so that the application is coupled with this interface to make computations with regards to energy consumption.

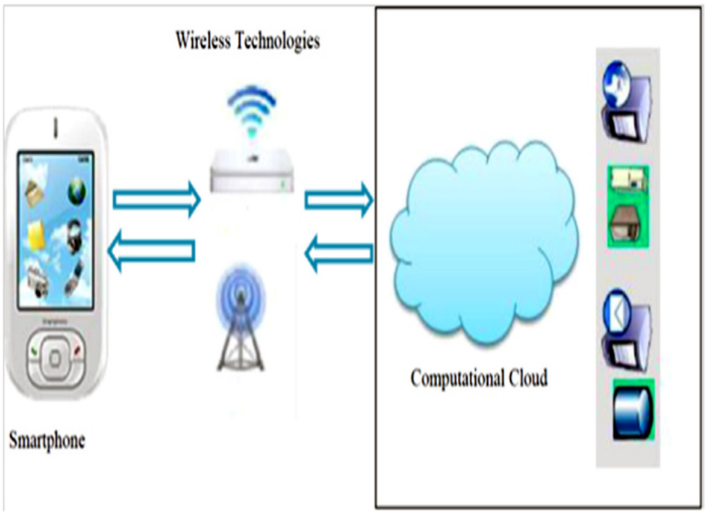


Fig. 1 Overview of task offloading

The figure1 is depicting the MCC consisting of smartphone and cloud environment with wireless communication links. The communication links can be WiFi, 3G/4G. In our proposed work, an interface is used to connect between the smartphone and the cloud environment. So the energy consumed in this case is minimized. This interface is deployed in the ADSL layer of the android devices so that the application is coupled with this interface to make computations with regards to energy consumption. Thus the amount of energy saved is

$$PC=C/M-Pi*C/S-Ptr*D/P \quad \text{---> (5)}$$

Cite this article as: Long CAI, Kokula Krishna Hari Kunasekaran, Prithiv Rajan. "Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost". *International Conference on Information Engineering, Management and Security 2016*: 83-91. Print.

S: measures the speed of cloud to compute C instructions being given by the SMDs.
M: measures the speed of SMDs to compute C instructions
D: the data need to transmit
Pc: the energy cost per second when the mobile phone is doing computing

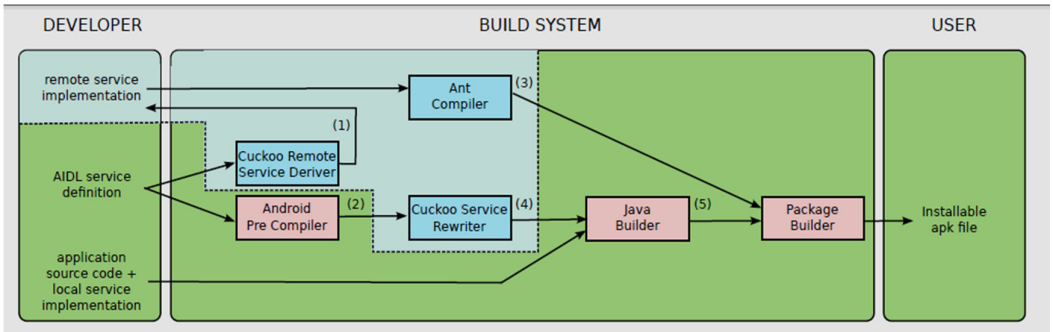


Fig. 2 Overview of Cuckoo Framework

Pi: measures the energy cost per second when the mobile phone is idle.
Ptr: measures the energy cost per second when the mobile is transmitting the data.

4.2 Cuckoo Framework

Cuckoo Framework represented in figure 2 focuses on minimizing effort to enhance the performance and reduce the battery usage of applications with heavy weight computation. The amount of energy saved by a device is given in equation 5.In connectivity drops offering a very simple programming model that supports for both local and remote execution and is able to bundle all codes in a single package. Next step is to integrate the system with existing development tools that are familiar to developers allows for automating large parts of the development process. This service offers a simple way for the application user to collect remote resources, which includes laptops, home servers and other cloud resources.

This model used the existing activity/service model in Android that makes a separation between compute intensive parts (services) and interactive parts of the application (activities), through an interface defined by the developer in an interface definition language (AIDL). If not, the code can be extracted from an interface. A the activity, the interface can be implemented as a local service when used it acts as a proxy object which is being present with it. The Cuckoo framework provides two Eclipse builders and an Ant build file that can be inserted into an Android project's build configuration in Eclipse. The Cuckoo Service Rewriter is also called first Cuckoo builder. The order of invocation can be first the Android Pre Compiler, next the cuckoo service rewriter and at the last the Java Builder. The work of Cuckoo Service Rewriter is to rewrite for each AIDL interface on the generated Stub, so that the decision of whether to execute locally or remotely is made at runtime by the Cuckoo framework. The remote implementation for the interface is called the Cuckoo Remote Service Deriver and derives a dummy implementation which acts as a second Cuckoo builder. This programmer reserves the right to implementation of the remote interface.

After the generation of the dummy remote implementation, the Cuckoo Remote Service Deriver will also generates an Ant build file, which then be used to build a Java Archive File (jar) that contains the remote implementation, which is installable on cloud resources. The order of execution of the service plays an important role. Now the Package Builder is invoked later the java builder is invoked finally the Cuckoo Remote Service Deriver and the resulting Ant file have to be invoked, so that the jar(java archive) will be part of the Android Package file that results from the build process.

4.3 Intelligent Offloading

Resource Manager Application is a part of the Cuckoo framework and it is able run on the smartphone. The registering process is very essential in order to make a remote resource known to a phone. The job of the remote resource is to register its address to this Resource Manager using a side channel. If a resource has a display, starting a server will result in showing a two dimensional barcode – a QR code on the resources' screen. The address of the server is fed into the QR code. Scanning the QR code is another task here. This is done by the smartphones because they are typically equipped with a camera and scan this QR code using a special resource manager application. The resource description file can be created from the resource if in case it does not have a visual output, the resource description file can be copied from the resource to the phone to register the resource. The application which uses the cuckoo computational offloading may be applied to all of its process over the long run, once the resource is known to the Resource Manager application, it can be used repeatedly for any application.

Cite this article as: Long CAI, Kokula Krishna Hari Kunasekaran, Prithiv Rajan. "Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost". *International Conference on Information Engineering, Management and Security 2016*: 83-91. Print.

When a method of a service is invoked by an activity, the work of the Android IPC mechanism is to direct this call through the proxy and the kernel to the stub. Normally, the invocation of the method is done by the stub which is always a local implementation of the method and then returns the result to the proxy. The Cuckoo system intercepts all method calls and evaluates whether it is beneficial to offload the method invocations or not, using heuristics, context information and history.

4.3.1 eyeDentify

Our first example application is eyeDentify, a multimedia content analysis application that performs object recognition of images captured by the camera of a smartphone. The purpose of this application is similar to that of the Google Goggles application which can recognize contact info, places, logos, landmarks, artworks, and books. This game can be offloaded by making computation offloading which allows the speed up of the computation with a factor of 60 along with the reduction of battery consumption with a factor of 40 and increase the quality of the recognition.

4.3.2 PhotoShoot

The second example that we will consider is a distributed augmented reality game, called PhotoShoot, with which we participated in the second worldwide Android Developers Challenge and finished at the 6th place in the category 'Games: Arcade & Action'. There are two players in this innovative game duel and they take part in the real world. The working mechanism for this game involves two Players each having 60 seconds and 6 virtual bullets to shoot at each other with the camera on their smartphone. The goal involves Face detection which is used to determine whether a shot is a hit or not. The duel will be recorded as the winner and he is first player that hits the other player. The face detection is the major compute intensive operation in this game. The face detection algorithm is present as an inbuilt package in the Android framework, so it is possible to create a local implementation to detect faces in an image. If we assume the process that happens without offloading, and if the processor of the smartphone is slower, the longer time it takes to analyse the shot, which causes the user of a slow smartphone a significant disappointment. However Offloading can, be used to make the game fair again over multiple times.

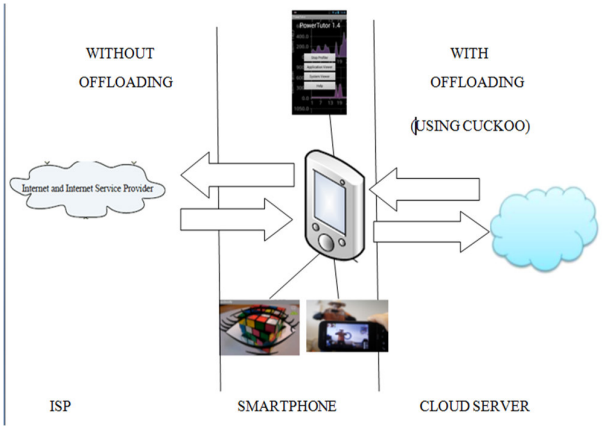


Fig. 3 Task offloading using Cuckoo framework

5. System Architecture

The System architecture pictured in figure 3 describes how the smartphone is connected to the internet and the cloud. While if the system is connected to the internet it will not be offloading the task rather it extracts the information needed to perform computation.

5.1 No Offloading Scenario

This scenario deals with the computation of energy consumed by the application during its working over the particular task. These two tasks are computation intensive since they involve image processing. At the earliest results shows off little difference over things but improves rapidly when there are huge records that need to be processed is given in equation 6 and 7. So the energy consumed in case of no offloading scenario is that the task to be executed locally on the smartphone itself. If Ttask is the time taken for executing the compute intensive algorithm, then time taken by the client for no-offloading scenario is:

Tclient = Ttask ---> (6)

Cite this article as: Long CAI, Kokula Krishna Hari Kunasekaran, Prithiv Rajan. "Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost". *International Conference on Information Engineering, Management and Security 2016*: 83-91. Print.

The total energy consumption Eno_offload is given by:

Eno_offload = Eclient = Pclient * T client ----> (7)

5.2 Offloading Scenario

For offloading scenario, the task is executed on remote server. Hence, in this case Tserver = Ttask and total time returned at the client is given by.

Tclient = Ttask + Tcomm ---->(8)

Where, Tcomm is communication time for sending the input file to server and getting the result back from server is given by equation 8. In this case, the offloading was done using Wi-Fi as well as 3G interface which is given in equation 9 and 10. For offloading, total energy consumption Eoffload_wifi and Eoffload_3G respectively are given by:

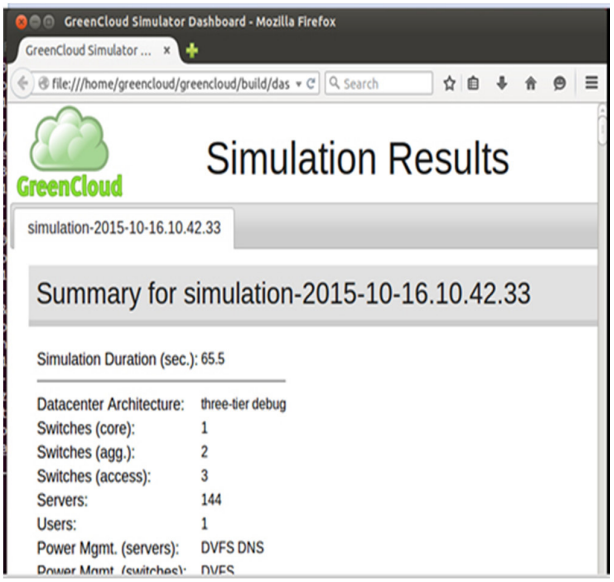
Eoffload_wifi = Eclient + Einternet_wifi + Edatacenter --> (9)
Eoffload_3G = Eclient + Ebase_station + Einternet_3G + Edatacenter ------->(10)
Where, Eclient = Energy consumed by the client device,
Edatacenter = Energy consumed at the datacenter.
Einternet = Energy consumed by the internet infrastructure
Energy consumption at the datacenter is given by:
Edatacenter = Eserver + Eoverhead ----> (11)

Where, Eoverhead = Energy consumed by HVAC, power supply and other overheads at datacentre which provides the energy of datacentre in equation 11.

6. Implementation

This constitutes energy computation for both the mobile as well as the cloud environment. This can be done by installing the cloud environment using green cloud and mobile environment can be brought into the device using android SDK and eclipse with built- in java configurations.

The figure 4 represents the green cloud installs the cloud environment along with its data center. The energy consumption for the devices which are connected can also be listed. So that the task installed in the cloud will be represented the graph for parameters like performance, response time, memory access, storage access, failure rate etc., are represented. While installing the mobile environment power tutor application also needs to be installed, which is helpful for Calculating energy consumption of each of the applications.



Cite this article as: Long CAI, Kokula Krishna Hari Kunasekaran, Prithiv Rajan. "Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost". *International Conference on Information Engineering, Management and Security 2016*: 83-91. Print.

Fig. 4 Green cloud simulation

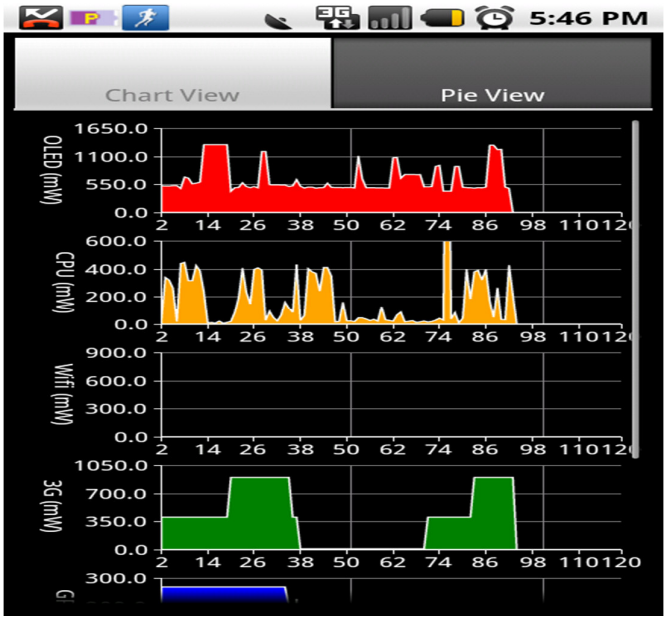


Fig. 5 Power tutor Application

The eclipse installing involves java RT environment to be installed onto the device. But the green cloud virtual disk is being coupled with the ns2, eclipse and c, c++ coding. Power tutor represents with all the parameters in the form of graphs. With all these parameters of energy responses of all the applications having graphs of all the cases are depicted. And from the results the application is decided upon whether to offload or not. In this case, both the applications needs to be offloaded since they are involving image processing. The figure 5 represents the application power tutor which allows the user to find the power consumption of individual applications.

GreenCloud simulator is an extension of NS-2 network simulator. It comes as an archived source tree, or as a pre-configured VM, which works with VirtualBox and VMWare Player. The VM also includes a pre-configured Eclipse environment, so it is the easiest way to download GreenCloud and start running simulations and/or modifying the source code.

7. Conclusion

Task offloading is one of the emerging topics in Mobile Cloud Computing. Task offloading serves as better job of taking the task to be done on to the cloud by creating communication link between the cloud and the mobile device. The communication links also plays a important role in task offloading. WiFi serves better offloading than 3G/4G. Because they don't deserve a physical layer communication link as wifi. Likewise offloading gives better yield when there are number of tasks to be done. This process includes the integration of three major components namely Cuckoo framework along with popular open source Android framework and at the last Eclipse development tool. The major advantage of this framework is that it provides a simple programming model, familiar to developers, and it allows for a single interface with a local and a remote implementation. The speciality of this Cuckoo model is that will decide at runtime where the computation can be local or remote. In addition to this, the Cuckoo framework adds an extra generic remote server, that allows for hosting the remote implementations of compute intensive services. A smartphone application to collect the addresses of the remote servers is also included. In this paper I evaluated the Cuckoo framework with two real world smartphone applications, an object recognition application and a distributed augmented reality smartphone game and showed that little work was required to enable computation offloading for these applications using the Cuckoo framework.

8. Future Enhancement

As of wireless network is concerned security breaches may occur. (1) Ensuring security for the communication link can also be taken as a next step. (2) Offloading from smartphone involves certain security violation that can also be analysed and rectified. (3) Sometimes incompatibility in both the device and cloud system may result in different results, it should be checked out and a fixed framework can also be devised. (4) 5G communication link can also be tested. (5) This offloading can also be extended as data,

Cite this article as: Long CAI, Kokula Krishna Hari Kunasekaran, Prithiv Rajan. "Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost". *International Conference on Information Engineering, Management and Security 2016*: 83-91. Print.

application partitioning etc., in order to perform the variants of offloading. This future enhancement may allow the device to have a better scope in the Mobile Cloud Computing environment. The energy as well as memory and computation cost can also be reduced.

9. References

1. A. Albasir, K. Naik, and T. Abdunabi, "Smart Mobile Web Browsing," in the 6th IEEE International Conference on Ubi-Media Computing, Aizu-Wakamatsu, Japan, Nov. 2-4 2013.
2. Android. http://developer.android.com/.
3. ADT Eclipse plugin. http://developer.android.com/sdk/eclipseadt.html.
4. ADC2.http://code.google.com/android/adc
5. A. Kansal and F. Zhao, "Fine-Grained Energy Profiling for Power- Aware Application Design," SIGMETRICS Perform. Eval. Rev., vol. 36, pp. 26–31, Aug. 2008.
6. Apache Ant. http://ant.apache.org/.
7. Bowen Zhou and Amir Vahid Dastjerdi A Context Sensitive Offloading Scheme for Mobile Cloud Computing Service 2015 IEEE 8th International Conference on Cloud Computing
8. A. P. Miettinen and J. K. Nurminen, "Energy Efficiency of Mobile Clients in Cloud Computing," in Proc. of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud'10), 2010, p. 4.
9. K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A Survey of Computation Offloading for Mobile Systems," Mobile Networks and Applications, pp. 1–12, 2012.
10. Cong Shi et al, Karim Habak et al, Pranesh Pandurangan "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Std. 802.11a-1999, 1999.
11. Eemil Lagerspetz and Sasu Tarkoma "Survey on Energy Consumption Entities on the Smartphone Platform," in Proc. IEEE 73rd Vehicular Technology Conf., 2011, pp. 1–6.
12. G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," IEEE Journal on Selected Areas in Communications, vol. 18, no. 3, pp. 535–547, 2000.
13. Guetal and Abebeand Ryan, Verbelen, "Energy Scavenging for Mobile and Wireless Electronics," Pervasive Computing, IEEE, vol. 4, no. 1, pp. 18 – 27, January-March 2005.
14. Majid Altamimi, K. Naik, "A Survey of Software Based Energy Saving Methodologies for Handheld Wireless Communication Devices," Dept. of ECE, University of Waterloo, Waterloo, ON, Canada, Tech. Rep. 2010-13, 2010.
15. Xinwen Zhang, Won Jeon, Simon Gibbs, and Anugeetha Kunjithapatham "Exhausting Battery Statistics: Understanding the energy demands on mobile handsets," in Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile hand- helds, ser. MobiHeld '10. ACM, 2010, pp. 9–14.

Cite this article as: Long CAI, Kokula Krishna Hari Kunasekaran, Prithiv Rajan. "Task Offloading to the Cloud by Using Cuckoo Model for Minimizing Energy Cost". *International Conference on Information Engineering, Management and Security 2016*: 83-91. Print.