ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

Survey on Application Models using Mobile Cloud Technology

Vinayak D. Shinde[1], Usha S Patil[2], Anjali Dwivedi[3]

H.O.D., Dept of Computer Engg, Shree L.R. Tiwari College of Engineering, Mira Road, Mumbai, India [1]

M.E., Scholar, Dept. of Computer Engg, Shree L.R. Tiwari College of Engineering, Mira Road, Mumbai, India[2,3]

Abstract: Nowadays, Smartphones supports a vast range of applications but in return demands a very high computational power increasing with time. But as we know Smartphone's computation power, memory, storage, and energy are limited solution to this problem can be Cloud Computing that offers a very wide computational power, memory, storage and energy. Hence, Researches are aiming to get Cloud Computing for smartphones to fulfil its requirements. But the traditional Smartphone application models not supporting the cloud computing features and requires specialized mobile cloud application models has became a challenge for the researchers. This article covers mobile cloud architecture, decision affecting entities on offloading, classification of application models, the latest mobile cloud application models and their critical analysis.

Keywords: Cloud, Mobile Cloud, Application Models, Survey of Mobile Cloud.

I. INTRODUCTION

A lot of popularity has been gained by Cloud computing in the recent years being a mixture of many computing fields .It provides storage, services, computing, and applications over the Internet and facilitates to reduce capital cost, deduct services from underlying technology, and is flexible in case of resource provisioning. Developers and manufacturers needs working together making the smartphones energy efficient and computationally capable also the major hardware and software level changes needed [1]. Computation offloading is defined as a procedure of migrating resourceful computations to the resource-rich cloud from a mobile device. The applications that are unable to run due to less Smartphone resources can be executed with the help of computation offloading based on cloud computing enhancing the applications performance and also battery power consumption reduction. Mobile cloud computing can be defined as an integration of cloud computing technology and mobile devices for making them resource-full in terms of context awareness, computational power, energy, memory and storage[7]. Mobile cloud computing is the outcome of interdisciplinary approaches comprising mobile computing and cloud computing Hence, it is also referred as MobiCloud computing. In this survey, the selection of application models is limited to the infrastructure based mobile cloud. Hence, ad-hoc mobile cloud based application models and associated issues and mobility of cloud infrastructure are not the scope of this survey.

The further paper is structured as follows. Section II explains mobile computation offloading workflow, cloud architecture, and entities affecting computation offloading process. Section III discusses the mobile cloud application models comparison criteria. Section IV highlights advantages and discusses the mobile cloud application models. Section V presents comparison of mobile cloud applications and application models. Section VI is a discussion of critical outstanding issues.

II. ARCHITECTURE OF MOBILE CLOUD & COMPUTATION OFFLOADING

The primary objective of mobile cloud computing is to provide enhanced user experiences to mobile users that may be in terms of computation time, battery life, communication, services and mobile device resource enhancement. However, based on the standard cloud service the mobile cloud applications are modelled that includes Infrastructure as a Service (IaaS) [30], Software as a Service (SaaS) [32], [33] and Platform as a Service (PaaS) [31].Some of the well-known services for mobile cloud computing include Google App Engine [31], Amazon Elastic Compute Cloud (EC2) [34], and Microsoft Azure [35].

1. Architecture Of Mobile Cloud

As shown in Figure 1, mobile devices can access cloud services in two ways i.e., through access points or through mobile network (telecom network).In the mobile network (telecom network provider) case, Base Station (BS) or a satellite link is used to connect the mobile devices such as cellular/satellite smartphones [36] to a mobile network. External satellite communication devices [37] are used, if the smartphones are not equipped with the satellite communication module.

Users will get Internet connectivity from telecom network which are connected to the Internet. User gets access to cloud based services through Internet, if and only if the users mobile have Internet connection. In the access point case, through Wi-Fi the mobile users connect to the access points Wi-Fi is further connected to the Internet service provider to provide Internet connectivity to the users. Moreover, compared to 3G connections Wi-Fi based connections provide low latency and consume less energy [14].

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

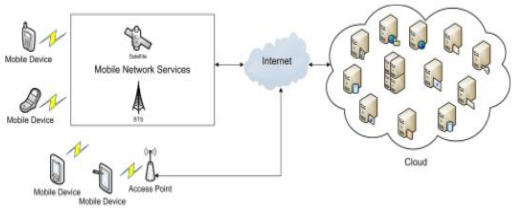

Fig. 1: Architecture of Mobile Cloud

2. Computation Offloading Decision Making

Computation offloading technique is proposed with the objective to migrate the large computations and complex processing from resource-limited devices (i.e., mobile devices) to resourceful machines (i.e., servers in clouds). This lowers the execution time and power consumption [1].

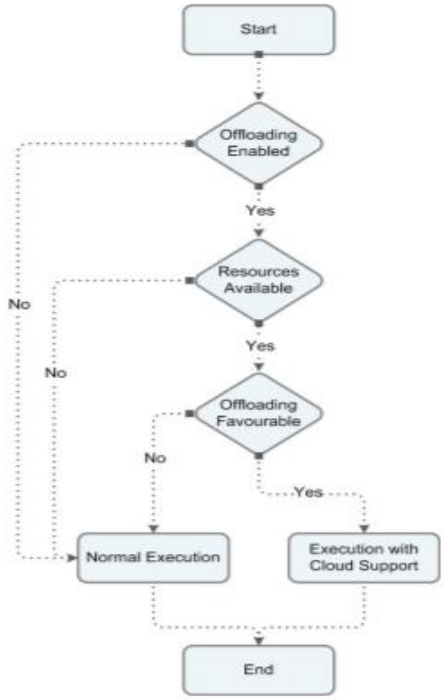

Fig. 2: Process of computation offloading

Figure 2 presents the basic workflow of the computation offloading process. The workflow starts with the execution of an application followed by checking the user's offloading permission. If offloading is enabled, then connection is checked to the cloud resources by application and notes the available/assigned resources.

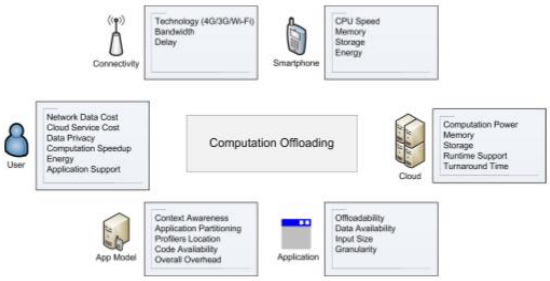

Fig. 3: Entities affecting computation offloading

The next step involves deciding whether offloading is favourable, depending on the users' desired objective (discussed in Section IV). If it is favourable, then the computation offloading is performed. Otherwise, the application performs all computations locally. Taking decision of computation offloading is an really complex process and is affected by different entities.Figure3 presents various entities that can affect the computation offloading decision in multiple ways.

1. User: computation offloading may be enable or disable by user based on cloud service cost, network data cost, job turnaround time and importance of data privacy. Moreover, User's objectives are considered for making decision.
2. Connection: Higher bandwidth and shorter delays will be provided by Wi-Fi based connections. Compared to Wi-Fi connections, 3G connections suffer from higher delays and provide lower bandwidth [14].Therefore, User prefers to use Wi-Fi connection. Hence, from a connection point of view, delay, network bandwidth and cost affect the computation offloading decision.
3. Smartphone: The latest smartphones are equipped with memory, high performance processors, storage and sensors. Users with high performance smart phones may need less support from mobile cloud compared to those users having low performance smartphones and runs out of resources very quickly.
4. Application Model: The application models may also differ in terms of application partitioning, context awareness, code availability in the cloud, profilers and overhead.
5. Application: when the application data is not available in the cloud and the input data size is too large, the computation is done by Smartphone. Alternatively, transferring a large amount of data may incur higher time and high energy is consumed in terms of communication, which may offset the benefits of offloading.
6. Cloud Service: In order to gain advantage of computation offloading, the leased cloud service must be rich in resources. For example, User may not get any improvement in the application performance, if a Virtual Machine (VM) [40], [41] (deployed in the cloud) and Smartphone have the same specification (computational power, memory).Although the scenario may be favourable in terms of energy (depending on data/code size), it is not favourable for enhancing application performance. In fact, the performance of the application may decrease due to the added computation and delay concerned in the offloading process.

III. CRITERIA FOR COMPARISON OF APPLICATION MODELS

Mobile cloud application models addressing most of the following mentioned parameters are said to be preeminent.
1. Context Awareness: The entities and parameters affecting the decision of computation offloading refer Context awareness of an application model. To be context aware is very important for an application model because static offloading is sometimes not beneficial, and there also may occur cases where there is degradation of

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

application performance with computation offloading [4].
2. Latency: In mobile cloud computing, latency can be defined as the total time required in the process of offloading the computation and then getting back the results referred as turnaround time.
3. Bandwidth Utilization: Bandwidth utilization is the amount of data migrated to computation offload. Hence, if the computation offloading needs a large amount of data transfer on runtime, then latencies can occur.
4. Generality: The support for a range of applications is said to be its generality. Practically, it has multiple applications with different resource demands with behaviour.
5. Privacy: Many advanced applications require user location to provide location based services. They are usually user-invoked for getting information related to location or invoked by service-provider for delivery of location-based services.
6. Complexity: The applications developed must be able to execute in both modes (i.e. Online and Offline) for the mobile cloud platforms. Moreover, the applications should utilize minimum bandwidth along with considerable delay. Hence, some models as discussed partition the applications into off-loadable components (i.e. sub-partitions) and manageable that can move to the cloud without more bandwidth requirement.
7. Security: Security is one of the most important factors in the adopting cloud computing [5]. Cloud computing can have a number of security issues like data access control, secure communication, data distribution in a distributed infrastructure, service availability, and data integrity
8. Programming Abstraction: Smartphone run different operating systems that consists variable hardware and software requirements. Therefore, the cloud platforms and heterogeneities in smartphones make development of mobile cloud applications difficult.
9. Scalability: Scalability is an important feature of cloud computing. The mobile cloud application models must support applications development that can scale to meet unpredictable user demands.

IV. CRITERIA FOR COMPARISON OF APPLICATION MODELS

Mobile cloud application models addressing most of the following mentioned parameters are said to be preeminent.
A. Performance Based Application Models: By utilizing cloud resources we can enhance the performance of mobile device applications. This becomes the primary objective of performance based application models.

1. Clone Cloud: Figure 4 shows high level architecture of the Clone Cloud. The main components of the Clone Cloud are migratory, node manager, profiler (dynamicprofiler), database and partition analyser. The migrator is responsible for resuming, suspending, merging and packaging the thread states on both sides. The node manager performs image synchronization, provisioning and handles communication between the migrated threads. For three tasks Chun et al. tested Clone Cloud prototype i.e., virus scan, behaviour profiling and image search. The advantage of this model is the recovery of data when a Smartphone is lost or destroyed.

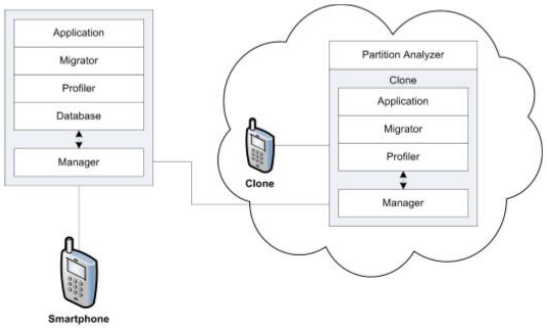

Fig. 4: Architecture of CloneCloud

2. Zhang et al. Model: Based on elastic applications technique, Zhang et al. [6] propose a model where multiple components are made by partitioning a single elastic application is called weblets. . A weblet can be defined as an independent functional unit of an application that can compute, store and communicate while keeping its execution transparent. The cost of migrating mobile apps to the cloud can be calculated considering four attributes into consideration: performance attributes, power consumption, security, monetary cost and privacy.

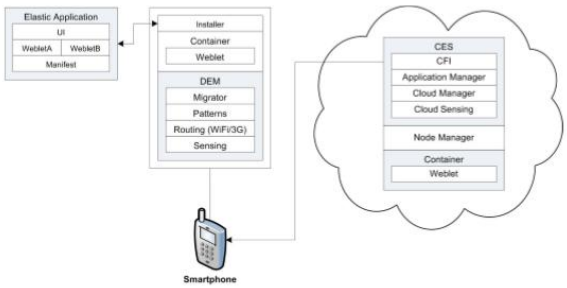

Fig. 5: Architecture of Zhang et al. Model

Figure 5 illustrates the Zhang et al. Model's main architecture. Cloud Elasticity Service (CES) consists of four sub-components, (2) An application manager that maintains and launches the weblets on the cloud platform, (1)a cloud manager that is responsible for provisioning and monitoring of resources provided to the weblets on the cloud,(4) a Cloud Fabric Interface (CFI),provides service to the elastic applications/smartphones and (3) a cloud sensing that collects information about resource consumption by weblets and provides that information to the cloud manager. Moreover, migration of weblets between the smartphones and the cloud is facilitated by CFI. Also, the node manager monitors the overall cloud node (server) resources. Consequently, the offloading decisions of the weblets are based on a cost model that accounts for various parameters, such as application performance, energy consumption and data privacy. Elastic applications are analysed by Zhang et al. for various security threats, which includes authorization, trustworthiness (of the weblet containers), authentication, auditing and communication.

B. Application Models based on Energy: Energy based application models are prepared to reduce mobile device

DOI 10.17148/IJARCCE.2015.41052

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

applications energy consumption by utilizing cloud resources that is achieved by reduction of the application's computational overhead through computation offloading. The computational tasks that are resource intensive are performed in cloud, hence also consumes lesser energy of mobile devices.

1. µ Cloud: The µ Cloud model focuses on the framework of applications from heterogeneous components to support reusability, flexibility and configurability. Therefore, the application frameworks are represented in form of graphical view to achieve organization of applications from heterogeneous components, where each component may execute on a mobile phones, cloud or smartphones (called hybrid components). The hybrid components require a middleware to have multiple implementations, such as WebOS, for execution. Due to this, on execution of an application graph each component insert its output into the resultant components.

C. Constraint Based Application Models: Energy Constraint based application models are prepared for execution of applications in resource constrained environment by using cloud resources. For instance, consider a mobile device with insufficient local resources for executing an application. In these models, the application components with light weight execution mobile device while the components that are resource intensive execute using cloud. Consequently, these models also enable high resource-demand applications to execute on resource limited devices.

1. eXCloud: eXCloud (Extensible Cloud) supports VM instance level computation offloading to the cloud. The main components of eXCloud are communication manager, migration manager, class pre-processor, object pre-processor and resource manager. The class pre-processor holds the responsibility of adding state capturing and restoration of code to the Java applications' before loading it to the JVM. The migration manager helps the migration requests, along with the object pre-processor.

D. Multiple objective Application Models: Energy The purpose of these models is to achieve multiple objectives mainly energy efficiency and performance at the same time.

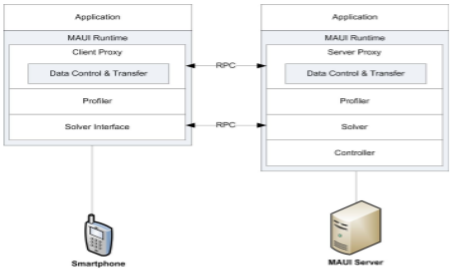

Fig. 6: Architecture of MAUI

1. MAUI: (Memory Arithmetic Unit and Interface): Programmer intervention is least. The main objective of this model is to minimize energy consumption of mobile devices, which is the first main challenge of the mobile industry. MAUI offloads all the resource intensive methods to the nearby infrastructure or cloud, provided the offloading is beneficial in terms of energy. MAUI uses a profiler (optimization engine) that helps in analysing energy consumption involved in the local and remote execution of the code. In MAUI, the application partitioning is dynamic. The offloading is done on the basis of methods instead of complete application modules to minimize the offloading delay. For local and remote execution MAUI creates two versions of Smartphone application, which uses Microsoft .NET Common Language Runtime (CLR). The architecture of MAUI is shown in Figure 6. In MAUI, the mobile device consists of three main components, i.e., solver interface, profiler and client proxy. The interaction with the solver is provided by solver interface (decision engine) and improves the offloading decision making. The information regarding the application energy consumption and data transfer is collected by profiler. The client proxy concentrates on the method offloading and data transfer. Similarly, the server side consists of server proxy, profiler, solver and controller. However, the profiler and server proxy working is similar to the Smartphone. The solver is the main decision engine of the MAUI that holds the scheduled methods and call graph of the applications. Lastly, the controller is responsible for resource allocation and authentication for incoming requests. The energy consumption is involved in the offloading procedure and MAUI focuses on it. MAUI will not be able to offload those methods which are not marked by programmer (for remote execution). Also, MAUI uses online design to create an energy consumption model and it also saves information about the offloaded methods (for future decisions).

2. ThinkAir: ThinkAir supports method-level offloading to a Smartphone clone. Desired QoS is achieved by executing multiple clones of the Smartphone simultaneously. ThinkAir requires minor changes in the source code of the applications. Therefore, it is the responsibility of the programmers to search all resource intensive methods that can be offloaded to the cloud for remote execution. ThinkAir framework's main architecture is illustrated in Figure 7.

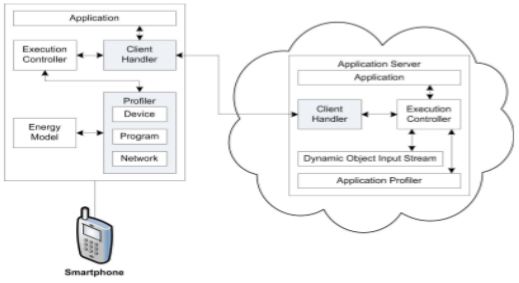

Fig. 7: Architecture of ThinkAir framework

The client handler is responsible for the execution of the communication protocols and management of the connection between the client and the cloud. The device profiler monitors the energy consumption of the device hardware resources, such as display screen, processor and antennas, etc. The program profiler monitors the program

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

parameters, for instance, acquired memory, execution time, and number of instructions, thread CPU time, and method calls. Lastly, network related parameters are monitored by the network profiler, for instance, delay, bandwidth and connectivity. ThinkAir on server side consist of a application server, server handler and dynamic object input stream. The server handler is in charge for getting computational requests and reporting the results back to the clients. Parallelism is supported by providing additional resources. Offloaded code is handled by the application server. Besides, it is light-weight and facilities the process of replication. Lastly, the dynamic object input stream controls the exceptions generated during the execution of the offloaded code. The main advantage of ThinkAir considers the energy consumption while making the offloading decisions supports parallelism, on-demand resource allocation and minimizes execution delays.

3. Cuckoo: Cuckoo is on partially offloading the applications to the cloud or nearby infrastructure, and programming is made easy for the developers by integrating the existing development tools that are familiar to the developers. Moreover, Cuckoo is made for Android platform and supports both local and remote method implementations. The development of Cuckoo application process and architecture of model is illustrated in Figure 8. A developer writes the source code in created project for development of Cuckoo application. Next, by using the existing activity or service model of Android, interactive parts (activities), computation intensive (services) and the application are separated. The separation is done with the help of an interface definition language called AIDL. Further, remote service and an interface are created by the build system and that contains a dummy implementation done by the Cuckoo Remote Service Deriver (CRSD). Moreover, the Cuckoo Service Rewriter (CSR) generates a stub or proxy for each AIDL interface so that on the basis of information provided by the Cuckoo Resource Manager a method can be invoked either locally or remotely. Implementation is done as both remote and local service. The local and remote interface implementation may look identical, but behaves in different way, based on the location of the service execution, algorithms and libraries may vary. Later, using CSR the developer codes the local service implementation and remote service implementation is overwritten.

Finally, the code is compiled by build system and provides an .apk file to the users which are easily installable. The Cuckoo based applications can offload their computation to any Java Virtual Machine (JVM) residing on the nearby infrastructure or cloud. Therefore, it is the smartphones' application responsibility to install the service(s) on the server. After installing the service, the address of the server is passed to the resource manager running on the Smartphone in the form of two dimension barcode or resource description file. Finally, the address registrar registers the address and the remote resource becomes functional for the Smartphone applications. The main advantage of Cuckoo is that it supports partial offloading of the applications to the cloud and uses renowned tools for application development.

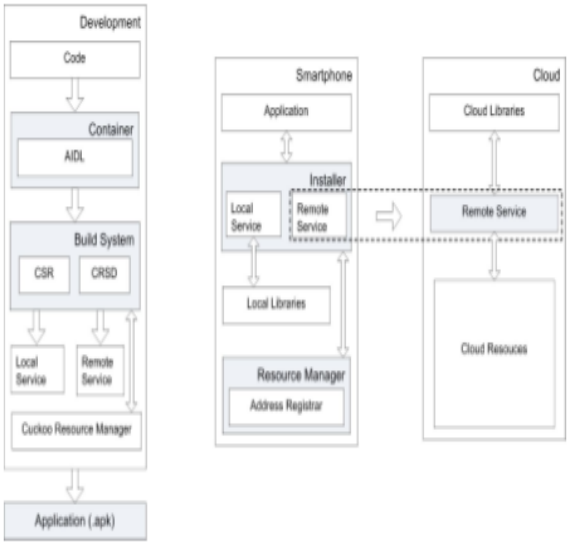

Fig. 8: Architecture of Cuckoo and application development process

Furthermore, it lacks security features to limit users from installing malfunctioned codes on the server and control illegal access to the resources.

V. MODELS COMPARISSON AND APPLICATION EXAMPLES

Objectives, Cloud From the analysis presented in Section IV, it is evident that none of the application models have considered all parameters highlighted in Section III. All models except have ignored the security and privacy issues of the applications' data, and smartphones' clone that resides in the cloud for augmented execution. None of the application models have focused the application piracy control in the cloud environment.

Table I. COMPARISION OF NEARBY INFRASTRUCTURE AND CLOUD PLATFORM

	Nearby Infrastructure (Virtual Cloud)	**Real Cloud**
Programming Abstraction	Neutral	Neutral
Scalability	Low	High
Security	Favourable	Favourable/ Unfavourable
Bandwidth Utilization	Neutral	Neutral
Latency	Favourable	Unfavourable
Complexity	Neutral	Neutral
Generality	Neutral	Neutral
Privacy	Favourable	Unfavourable
Context awareness	Neutral	Neutral

A. Comparison of Application Models: Table I presents a evaluation of nearby infrastructure and public cloud platform according to criteria quoted in Section III.

DOI 10.17148/IJARCCE.2015.41052

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

B. Application Examples for Mobile Cloud Computing: Here, we present few more application examples for mobile cloud computing to show how this technology can prove beneficial for different domains.

1. Mathematical Tools: Composite mathematical calculations, such as multiplication of a very large matrix involve large computations and eat considerable amount of energy. Therefore, mobile devices can offload such computations to cloud that may help to achieve energy efficiency and better performance due to high computational power of the cloud.
2. File Search: The most recent smart phones include storage capacity of up to 80 gigabytes. Therefore, smartphones can store large amount of files due to which look for functions may take only few minutes. However, by using mobile cloud computing the search functions can implement on the Smartphone clone (in the cloud) that may result as better performance and energy efficiency (assuming that the Smartphone and Smartphone clone in the cloud are pre-synchronized).
3. Imaging Tools: Image processing tasks insist large computations and the operations may take up to little minutes for completion, for example, when rendering a 3D image from a source file. Therefore, the imaging tools can offload intense computational operation to the cloud and benefits from improved performance and energy efficiency depending on the runtime environment and existing resources (network, cloud).
4. Games: Gaming applications usually require intense computation (on large datasets) and rapid response time for user communication. Therefore, computation offloading is not compulsory as it may reduce the game performance. For instance, First Person Shooting (FPS) games are not appropriate for computation offloading. However, some games involve large computation using small datasets that allow quick computation offloading and direct to energy efficiency.
5. Download Applications: Downloading files at low download rate consumes more energy compared to high data rate. Hence, it is beneficial in terms of energy to download files in the cloud and then transfer files to the mobile device with high speed. For instance, mobile cloud
6. Antivirus Applications: Antivirus applications are becoming vital part of smartphones due to the increasing threats from viruses and malwares, however, scanning a Smartphone for viruses requires computation that consume high amount of energy. Using mobile cloud computing, the Smartphone clone can be scanned in the cloud to save energy (assuming that the Smartphone and its clone are pre-synchronized). This type of applications can be developed using Satyanarayanan et al., and Clone Cloud Model.
7. Security: To execute wide range of applications, software and hardware level enhancements of smartphones are enabling. Installing, number of applications may increase the threat from malwares that can leak user's personal information from the Smartphone. The mobile device applications can execute their services in the cloud that may reduce the threat from attackers by using mobile cloud computing. The mobile devices execute more trusted and less complex applications with energy efficiency and enhanced security. The mentioned security aspects are the focus of the MobiCloud framework along with other important issues, such as secure routing and trust management.

VI. CONCLUSION

The applications developed usually supports one execution platform, thus, limiting the offloading of the elements (applications, components, clones) to other platforms. The execution platforms of mobile cloud need standardization to ease computation offloading for mobile cloud platforms. New energy consumption models are needed to facilitate making accurate decision by consideration of the main entities by involvement in offloading process. To confine mobile user access to optimum resources, or revoke access and timely identification of the fake users, new policies are required. Need to refine the data management laws and standardize the mobile cloud computing platforms accordingly, so that mobile users can truly benefit from the cloud computing technology and the mobile cloud computing can flourish.

REFERENCES

[1] C. Mascolo, "The power of mobile computing in a social era," IEEE Internet Computing, vol. 14, no. 6, pp. 76–79, 2010.
[2] Rackspace cloud. Accessed April 10th, 2012. [Online]. Available: http://www.rackspace.com/Fakhar, Faiza, and Muhammad AwaisShibli. "Comparative analysis on security mechanisms in cloud." Advanced Communication Technology (ICACT), 2013 15th International Conference on. IEEE, 2013.
[3] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in Proc. 8th international conference on Mobile systems, applications, and services. ACM, 2010, pp. 49–62.
[4] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" Computer, vol. 43, no. 4, pp. 51–56, 2010.
[5] Z. Zhou and D. Huang, "Efficient and secure data storage operations for mobile cloud computing," in Network and Service Management (CNSM), 2012 8th International Conference on. IEEE, 2012, pp. 37–45.
[6] X. Zhang, S. Jeong, A. Kunjithapatham, and S. Gibbs, "Towards an elastic application model for augmenting computing capabilities of mobile platforms," in Mobile wireless middleware, operating systems, and applications. Springer, 2010, pp. 161–174.
[7] Khan, Ab Rouf, Marini Othman, Sajjad Ahmad Madani, and Samee U. Khan. "A survey of mobile cloud computing application models." Communications Surveys & Tutorials, IEEE 16, no. 1 (2014): 393-413.