VOL. 11, NO. 23, DECEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SWARM ORIENTED HYBRIDIZED APPROACH TO INTENSIFY THE STORAGE SPACES IN SMARTPHONES

Solomon Babatunde Olaleye and Ishan Ranjan
Department of Computer Science and Engineering, Sharda University, Greater Noida, India
E-Mail: olaleye3@yahoo.com

ABSTRACT

The advent of the mobile cloud computing is the recent exploration among the users. It is a service obtained from the cloud computing that shares the resources, software and information to the peers and with the smartphone as a utility service. Resources namely storage space, memory and processing time are quite discontented at the end users. In this paper, we make a study about enhancing the storage spaces of the smartphones using swarm prediction approach. We propose a hybrid algorithm named Gravitational Search Algorithm (GSA) with the aid of Binomial Formation of Arbitrary Numbers (BFAN).The task of this hybrid algorithm is to efficiently allocate the cloud resources to its intended mobile cloud users. Better positioning of the sub-jobs in the cloud servers may enhance the storage space is the thought of this research. Key formation is the basic building block of the security issues in cloud computing. Using BFAN, the best and secured fitness value is obtained. The parameter settings and its results proved the effectiveness of our proposed approach.

Keywords: mobile cloud computing, resource utilization, gravitational search algorithm, binomial distribution and fitness functions.

1. INTRODUCTION

In the modern environment, the usage of mobile phones reached a greater demand among the users. Every user bragged by the mobile phones [1]. The widely use of mobile phones lead to the plethora of mobile services. “Dreaming of accessing the data anywhere at anytime” is the vision of the cloud computing. The arrival of cloud computing in mobile web makes the mobile users to utilize the infrastructure, platform and software with the aid of cloud providers on demand basis [2]. The outgrowth of cloud computing in mobile device is known as Mobile Cloud Computing (MCC). Mobile Cloud Computing is a service that allows resource constrained mobile users to adaptively adjust processing and storage capabilities by transparently partitioning and offloading the computationally intensive and storage demanding jobs on traditional cloud resources by providing ubiquitous wireless access. The thought of migrating the data processing and storage from mobile devices to the cloud was proposed by Aepona. As mobile devices doesn’t need high configuration due to the exaggerate power of cloud platforms. Cloud possesses high computational ability whereas it is restricted in mobile. To incorporate the mobile functions into the cloud holds some issues such as:

- Constrained resources
- Networks issue
- Security oriented issues.

Constrained resources: Computational offloading is an important factor to enhance the responsiveness from the computational process [2].There are various offloading techniques are available; yet the partition task between cloud and mobile applications is a complicated one. This issue exists due to two practical facts: a) Unpredictable user’s number using the cloud applications b) Limited resources supplied by the application providers c) Limited Battery power d) low quality display.

Networks issue: Latency, Bandwidth, availability and heterogeneity were the main issue in mobile networks related issues.

Security oriented issues: Security is a major concern when considering storing part or all of your data in the cloud. Privacy considerations are very important before a user or organization opts for cloud. Before using the cloud for sensitive data, be certain that the cloud provider has a very strong mechanism for protecting sensitive data in its custody.The paper is organized as follows: Section I depicts the basic definitions in Mobile Cloud Computing (MCC) and current issues and challenges in MCC. Section II describes the various techniques studied by other experts. Section III portrays the proposed work. Section IV discusses about the experimental actions carried out. At last, concluded in Section V.

2. LITERATURE SURVEY

Byung*et al* [1] analyzed the importance of Clonecloud model. They established the restriction associated with the Clonecloud model. From the investigation, the Clonecloud model is not applicable for all mobile clients. A virtualized screen based is established towards an elastic model development to enhance the storage utilization [2] [3]. Kaur *et al* [4] suggested an offloading computational scheme in dynamic approach. Anyhow, there are a less number of studies focused on the storage space enhancement with the aid of online server. A continuous advancement in technologies of smartphones treated as the most important part of daily life [5]. The mobile appliance possesses the functions to SMS and to make calls, updating the status of the social media sites, accessing the news and playing different games. A different type of storage consumes a variant level of power. Kumar and Lu [6] portrayed an explosive growth of multimedia schemes that causes greater demands for Android Smartphones. Because of variant applications installed in smartphones, the user experiences constrained resources like network bandwidth, battery life, heavy data processing and inadequate storage space. Carroll *et al* [7] estimated the energy consumption breakdown by the major hardware tools in a smartphone. Their direct measurements of every component's present and voltage are utilized to evaluate power.

Balasubramanyam *et al* [8] suggested that the software capability of storage is not properly constructed. This leads to inefficient use of reduced energy features of hardware of mobile storage. According to the statistics, the software storage depletes a greater amount of energy than the mobile hardware.Storage is viewed as important performance criteria for several mobile applications. Storage overhead is compared with background applications such as messaging services, file synchronization and other software updates can depletes the heavy storage computation. The author in [9] developed an insight on storage enhancement using flash or non-volatile memory.Liu *et al* [10] studied the performance of the storage and file systems to create a better user experiences. As file systems can consume the energy level, it's been studied. Let us consider an e-commerce websites that holds huge number of images of the smaller products. Some games may acquire a vast amount of storage to initiate the games. Thus the storage performance is widely studied among the various researchers to design effective systems. Kim *et al* [11] studied about the storage performance of applications such as web browsing, application installation, maps, facebook and email. Similarly Guo*et al* [12] mentioned that smartphones have become much and much familiar in the past couple of years. Although certain hardware design of several smartphones have been comparable to a 10 year old PC such that the computation and storage limitation still a great drawback for smartphones if its target is to become an alternative platform for traditional PC.Although several smartphones enhance processors up to 1 gigahertz, their storage and memory remains restricted compared to desktops, servers and laptops. On the other side data centers and cloud computing are capable to offer unlimited storage and abilities of computation. Smartphones are capable to access storage of cloud through third party providers and also libraries of application such as Android market of Google and Apple's App-Store and these cloud computing abilities can be accessed through special applications or web browsers.

Cuervo*et al* [14] suggested the enhancement of storage performance using fragmentation techniques. The key parameter in updating the android version is difficult in processing the hardware constraint. A virtual machine with the advent of cloud computing is studied by Oberheide*et al* [15]. He reported that more than one cloud server can be used for enhancing the storage performance. Their target is to use the non-constrained resources for mobile users. Satyanarayan*et al* [16] reported that when the storage in android is full, a transparent storage data model is used to accumulate the temporary data files.

3. PROPOSED WORK

A. Preliminaries

Android is an open source platform that comprised of Operating systems, Applications key and Middleware. It has been emerged from Open Handset Alliance. Google is one of the active key participants. The primary components were listed as follows:

- Activities
- Services
- Content Providers
- Broadcast Receivers

Each process operates in lifecycle within the system. The role of Activities components is for user interaction and performing some basic computing operations. A service is used for network intensive operations and doesn't have GUI. Content Provideris used for accessing and sharing the data among the applications. At last, broadcast receivers are activated by the events that are broadcasted by the alternate components of the system. Figure-1 describes the working process of IPC.

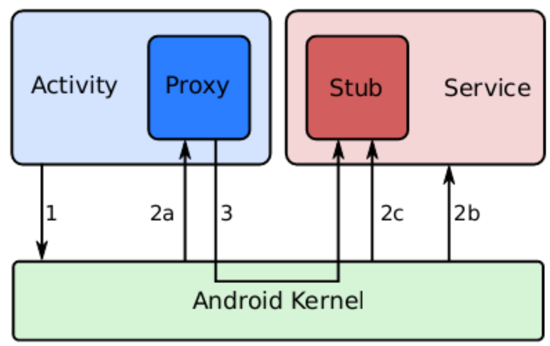

Figure-1. Mechanism of Android IPC [5].

The working steps as follows:

1. Activity gets bind to the service
2. a) Then it gets a proxy object back from the kernel.
   b) The kernel sets up the service

   c) Containing the stub of the service.

3. Subsequent method invocations by the activity on the proxy (3) will be routed by the kernel to the stub, which contains the actual implementation of the methods.

The building process of android includes Builders such as:

- **Android resource manager:**It generates Java file that enables us to access the assets like Images, Sounds, and layout definitions in code.
- **Android pre-compiler:** It generates java file using Activities Interface Definition Language (AIDL).
- **Java builder:** It compiles source code and the generated java code.

- **Package builder:** It bundles the resources, the compiled code and the application manifest into a single file. The single file is known as Android Package File (.apk).

B. Hybrid GSA +BFAN algorithm

The novel hybrid GSA + BFAN algorithm is to study about the enhancing the storage space without compromising the security. The GSA is used for determining the optimal positions of the cloud servers. Based on the output of GSA, the BFAN is used for evaluating the goodness of fit. Each user's requests and its data are forwarded to the servers using mobile network services. The user's data like ID, location etc. is forwarded. The network operator in mobile provides services like authentication, authorization and accounting. By doing these functions, the request is transformed to the cloud via internet. Everycloud owns a group of networks N= $\{N_1, N_2, N_3, N_4…N_n\}$. A group of application servers S= $\{S1, S_2….S_s\}$ with a group of resources R= $\{r_1, r_2….r_r\}$ is assigned for every cloud users. Based on user IP, the network manager communicates with the intended cloud server.BinomialFormation of Arbitrary Numbers (BFAN) is the building block of generating algorithms and protocols.Key Formation is the most important form of data encoding process. An efficient security lies in non-repetition key formation, ensure better results.The key formation is done using Binomial Formation of Arbitrary Numbers (BFAN).The proposed work process is explained as follows: Assume that there are M agents (servers) resides in the cloud. Each server S_i is picked up as an agent that symbolizes a unique solution (servers) which are assigned for every client's sub-jobsSJ_i. Let SJ= $\{SJ_1, SJ_2….SJ_n\}$ of n mobile users to be processed in a stipulated time. The agent (server) position is updated at every phase. The direction and velocity of an agent is used for predicting the user's next position. If an agent's velocity is positive, then it moves towards positive direction or vice versa.

A) Initialization of the agents (Cloud Servers):

$$CA_i = (a_i^1, ...a_i^d, ..., a_i^n) \text{ For i= 1, 2… N} \quad (1)$$

a_i^d represents the positions of i^{th} agent in d^{th} dimension; n is the space dimension.

B) Evaluating the best fitness estimation:

The fitness estimation is done by Binomial Formationof Arbitrary Numbers (BFAN).Binomial distribution belongs to the class of probability distributions. The potential solution is obtained by the repeated tests. Each test holds two outcomes either opting the best agents or worst agents. Let us assume that all of the tests in process are independent. The probability of choosing the best agents is declared as the probability of the success rate of the agents. In maximization problem, the fitness function gravity (Constant weight), best and worst of all agents are estimated. The fitness function is given as:

$$Fit_j(t) = {}_nC_x * p^x * (1-p)^{n-x} \quad (2)$$

Where n= number of agents (servers); x= total number of successes to pick the best agents; p= probability of a success on an individual trial.

$$Best(t) = \max fit_j(t) \; j \in (1…N) \quad (3)$$

$$Worst(t) = \min fit_j(t) \; j \in (1…N) \quad (4)$$

Where fit $_j$ (t) denotes fitness value of j^{th} agent (server) at iteration t; N is the list of available agents. Best (t) and Worst (t) denotes the best and worst fitness at iteration t.

C) Computing the masses& accelerations of the agent's:

The mass of the agent's is defined as the number of jobs, Processing power and Request/ Response to the tasks that an agent can hold. The agent's mass is estimated using twomasses namely, Active mass and Passive mass.The active mass represents the group of task executing by the server. Passive mass represents the group of task waiting in a queue to be executed after the successful completion of active mass.

$$M_{ai} = M_{pi} = M_i \quad \text{where i= 1, 2… N.} \quad (5)$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (6)$$

Where mi (t) represents the current mass of the i^{th}agents at iteration t:

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \quad (7)$$

Where M_i (t) denotes the aggregation of the masses of all agents.

Then the acceleration of an agent is calculated as:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \quad (8)$$

$F_i^d(t)$ is the total force on i^{th} agent and it is estimated by the formula:

$$F_i^d(t) = \sum_{j \in Sbes, j \neq i} rand_j F_{ij}^d(t) \quad (9)$$

VOL. 11, NO. 23, DECEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Sbes is the group of S agents (server) comprised of best fitness value and highest mass.$Rand_j$is the uniform random variable with the interval [0, 1].

$$F_{ij}^{d}(t) = G(t).\frac{M_{aj}(t)}{R_{ij}(t)+\varepsilon}.(x_j^d(t) - x_i^d(t)) \tag{10}$$

$F_{ij}^{d}(t)$ is the processing time on agent i from agent j at d^{th} dimension and t^{th} iteration. $R_{ij}(t)$ is the Euclidean distance between two agents i and j at iteration t. Epsilon (ε) is constant. $M_{aj}(t)$ represents the task executing at the server.

D) Determining the velocity and position of agents (servers):

This step determines best agents for executing the task.

$$v_i^d(t+1) = rand_i * v_i^d(t) + a_i^d(t) \tag{11}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{12}$$

v_i^d represents the velocity of the agent in i^{th} iteration at d dimensional space. x_i^d represents the position of the agent in ith iteration at d dimensional space.

E) The step (ii) - (iv)are processed until the best position of the server is found. Once the best agent is picked up, the query response to the mobile users will be easily. The direction and velocity of an agent is used for predicting the user's next position. If an agent's velocity is positive, then it moves towards positive direction or vice versa.

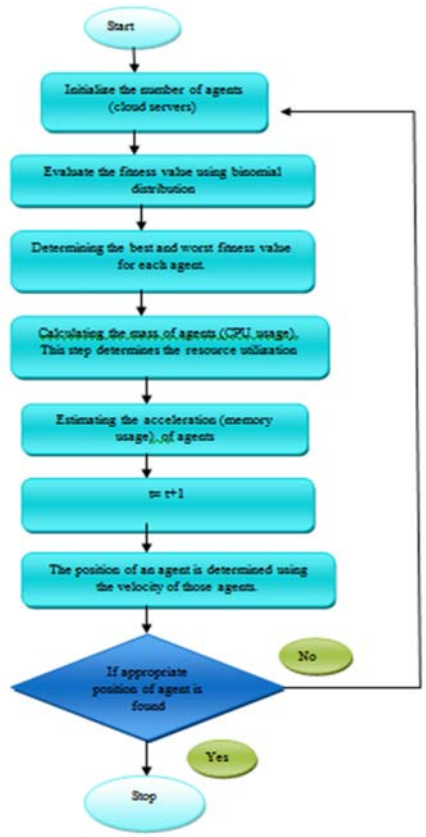

Figure-2. Proposed workflow.

4. PERFORMANCE EVALUATION

The mobile users aim for optimal communication with the cloud. A high degree of interoperability should ensure between mobile users and cloud servers. The parameter settings of the improved GSA were showed in Table-1.

Table-1. Estimating the BFAN.

Parameters	Value
No. of agents (servers) (a)	50
Max. no. of iterations (t)	100
No. of resources (R)	32
No. of Sub-jobs (SJ)	32
No. of dimensions (d)	2
Gravitational Constant G(t)	10
Epsilon (ε)	10
No.of virtual machines	3

Table-2. Estimating the fitness value using BFAN.

No.of agents (n)	Formation of successful agents (x)	Probability that at least single server will exist on anyparticular iteration (p)	Standard deviation	Exactly p out of n	P or fewer out of n	P or more than out of n	Binomial ratio
50	10	0.5	3.535	0.000009123 616	0.000011930666	0.99999719295	-4.1
50	20	0.5	3.535	0.041859149 253	0.101319375532	0.94053977372	-1.27
50	30	0.5	3.535	0.041859149 253	0.94053977372	0.101319375532	+1.27
50	40	0.5	3.535	0.000009123 616	0.99999719295	0.000011930666	+4.1
50	50	0.5	3.535	0	1.0	0	+6.93

$Fit_j(t) = {}_nC_x * p^x * (1-p)^{n-x}$

Where n= number of agents (servers); x= total number of successes to pick the best agents; p= probability of a success on an individual trial. From the table 2, it is inferred that the probability of selecting the best cloud servers is estimated. The total of 50 servers, the formation of 10 servers with atleast single server is found to be -4.1. Similarly, the formation of 20, 30, 40 and 50 servers with atleast single server is estimated.

Best (t) = max fit_j (t) j € (1…N) is estimated at +6.93.
Worst (t) = min fit_j (t) j € (1…N) is estimated at -4.1.

Let us consider the positive fitness functions from 0.0, 2.0, 4.0, 6.0 and 8.0 executes for 100 iterations.Then the performance metrics studied were the Resource CPU usage, Resource memory usage, and Average execution time.

a) Resource CPU usage

The performance analysis of VM1, VM2 and VM3 were analyzed. The application is accessed by VM1. Initially, 20% of CPU usage is consumed by VM1. When the threshold level of 55% is increased by VM1, the next machine VM2 is identified. Similarly, the threshold level of VM2 is reached then the next machine VM3 is used. Based on the masses of the agents, the cloud resources are effectively utilized.

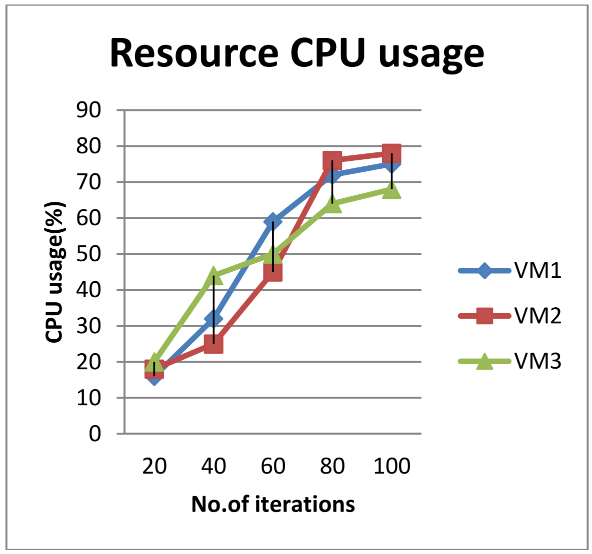

Figure-3. CPU Usage.

b) Resource memory usage

Figure-4 denotes the memory usage of using three virtual machines VM1, VM2 and VM3. Initially, all the three machines execute an average level of memory performance. When the threshold level of VM1 is reached, the next VM2 is utilized. Similarly, VM2‘sthreshold is reached, the next VM3 is used. Based on the forceimpulseby agentsusing the formula F_{ij} (t), the memory load is equally balanced.

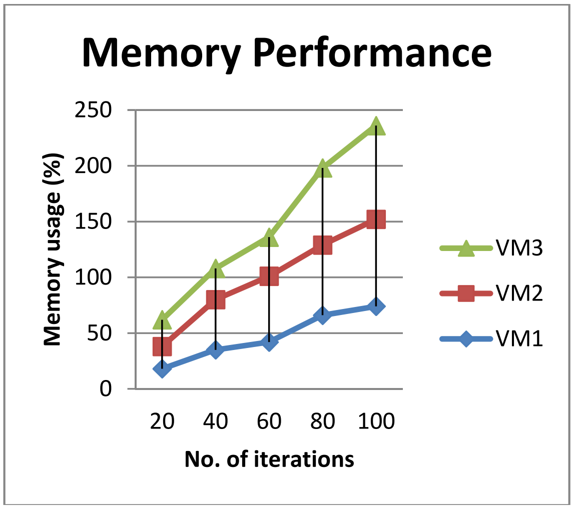

Figure-4. Memory usage.

c) Average execution time

Figure-5 denotes the average execution time taken by the cloud resources. Once the velocity and position of an agent is determined, the time taken between mobile users and cloud servers is estimated.

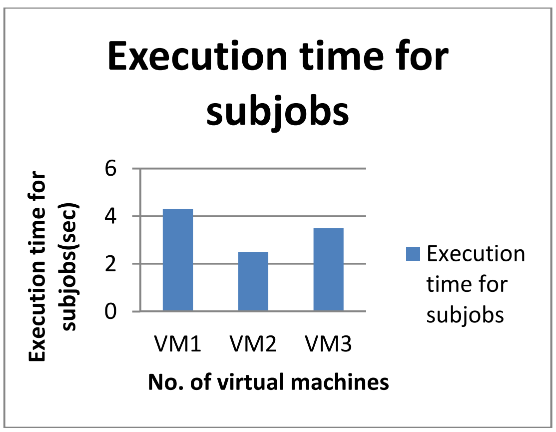

Figure-5. Average execution time taken by the sub-jobs.

5. CONCLUSIONS

Nowadays, there exists a great demand for cloud servers. The plethora of cloud servers among the huge number of mobile cloud users should ensure a greater reliability and availability of storage space. The availability of the data is ensured by the proper allocations of the cloud servers based on their physical locations. Increase the rate of data availability, Increase the rate of reliability is achieved. This research studied the proper allocations of sub-jobs to the cloud resources. We proposed a novel algorithm named GSA+ BFAN, whichaims for the better positioning of the cloud servers in order to enhance the storage space. The parameter setting depicts the quantitative measurement used for validating the proposed work. We carried out an extensive performance analysis on the CPU usage, memory usage and average execution of time taken by the sub-jobs. From the analysis, we were able to judge that the enhancement of the storage space purely depends on the effective capacity planning between the mobile users and cloud servers without solely depending on mobile applications and battery life. As a future work, we shall make an attempt to enhance the storage spaces in the view of various cryptographic techniques.

REFERENCES

[1] Byung as cited in Qi H. and Gani A. 2015. Research on Mobile Cloud Computing: Review, Trend and Perspectives. Retrieved On: Retrieved From: http://arxiv.org/ftp/arxiv/papers/1206/1206.1118.pdf.

[2] Jeong S., Zhang X., Kunjithapatham A. and Gibbs S. 2010. Towards an elastic application model for augmenting computing capabilities of mobile platforms.Mobile Wireless Middleware, Operating Systems, and Applications.pp.161-174.

[3] Lu Y., Li S. and Shen H. 2011. Virtualized screen: A third element for cloud-mobile convergence. Multimedia, IEEE. 18(2): 4-11.

[4] Kaur. K, Sharma. S and Arora R. 2014. Performance Enhancement of Android Phones by Offloading Computation Dynamically. International Journal of Advanced Research in Computer Science and Software Engineering. 4(5):1182-1186.

[5] Murmuria R., Medsger J., Stavrou A. and Voas J. 2012. Mobile Application and Device Power Usage Measurements. In Proceedings of Sixth International Conference on Software Security and Reliability (SERE) 2012 IEEE. 147-156.

[6] Kumar K. and Lu Y. H. 2010. Cloud Computing for Mobile Users: Can Offloading Computation Save Energy? IEEE Computer Society. 43(4): 51-56.

[7] Carroll A. and Heiser G. 2010. An Analysis of Power Consumption in a Smartphone. In Proceedings of the 2010 USENIX Annual Technical Conference, Boston, MA, USA.

[8] Balasubramanian N., Balasubramanian A. and Venkataramani A. 2009. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications.In IMC '09 Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, ACM IMC, Chicago. 280-293.

[9] Mendon A. A. and Sass R. 2008. A Hardware File System Implementation for High-Speed Secondary Storage. In Reconfigurable Computing and Fpgas. IEEE International Conference on ReConFig'08. pp. 283-288.

[10] Liu W., Hu Y., Rong H. and Li R. 2015. Optimizing File System Performance for Android-based Consumer Electronics by an Experimental Method. Accessed. http://ss.hnu.cn/Lnml/UpLoadFolder/Private/viqz/OtherFile/JCIT1-6476JE-revision.pdf.

[11] Kim H., Agrawal N. and Ungureanu C. 2011. Examining Storage Performance on Mobile Devices.Proceedings of the 3rd Association of Computing Machinery (ACM), Symposium on Operating Systems Principles (SOSP) and Workshop on Networking, Systems, and Applications on Mobile Handhelds, New York, USA.No.6.

[12] Guo Y., Zhang L., Kong J., Sun J., Feng T. and Chen X. 2011. Jupiter: Transparent Augmentation of Smartphone Capabilities through Cloud Computing. In Proceedings of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handheld,New York, NY, USA.2-6.

[13] Chun B. G. and Maniatis P. 209. Augmented Smartphone Applications through Clone Cloud Execution.In Journal of Hot Topics on Operating Systems (HotOS). 9: 8-11.

[14] Cuervo E., Balasubramanian A., Cho D. K., Wolman A., Saroiu S., Chandra R. and Bahl P. 2010. MAUI: Making Smartphones Last Longer with Code Offload.In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (ACM MobiSys ’10) San Francisco, CA, USA: ACM. pp. 49-62.

[15] Oberheide J., Veeraraghavan K., Cooke E., Flinn J. and Jahanian F. 2008. Virtualized In-Cloud Security Services for Mobile Devices. In Proceedings of the First Workshop on Virtualization in Mobile Computing, MobiVirt. ’08: 31-35.

[16] Satyanarayanan M., Bahl V., Caceres R. and Davies N. 2009. The Case for VM-based Cloudlets in Mobile Computing. IEEE Pervasive Computing, Special Issueon VM. 8(4): 2-11.