www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 5 may, 2014 Page No. 6165-6172

A Literature Survey on Mobile Cloud Computing: Open Issues and Future Directions

Nitesh Kaushik, Gaurav, Jitender Kumar
Nitesh133@gmail.com
garry_theproud@yahoo.com
Jitenderk_107@yahoo.com
Computer Science and Engg. Department, DCRUST, Murthal

Abstract - Given the advances in mobile phones, users start to consider a mobile phone a personal information processing tool. So users want to execute its various operations on the top of mobile devices. Researchers have long recognized that mobile hardware is necessarily resource poor relative to static client and server hardware. Mobile cloud computing (MCC) which combines mobile computing and cloud computing is a good solution to this problem and has become one of the industry buzz words and a major discussion topic since 2009. This paper presents a review on the background and principle of MCC, characteristics, recent research works and future research trends.

Keywords – mobile cloud computing, partitioning, augmentation.

1. INTRODUCTION

Advances in mobile hardware and software have allowed users to perform tasks that were once only possible on personal computers and specialized devices like digital cameras and GPS personal navigation systems. But still mobile phones are not capable of running all type of applications on its own without any constraint [1], [5]. Customers prefer improvements in battery life, storage size, weight etc over computation capability. So to encounter these computation-intensive applications Mobile cloud computing is being introduced. Mobile cloud computing is combination of two well established computing schemes, cloud computing and mobile computing.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [3]. Cloud computing is a style of computing in which dynamically scalable resources are provided as a virtualized service [20]. It allows service providers and other users to adjust their computing capacity depending on how much is needed at a given time or for given task. Cloud computing delivers infrastructure, platform and software as services, which are made available as subscription-based services in a pay-as-you-go model to customers[2]. As shown in figure below, mobile devices are connected to the mobile networks via base stations (e.g., base transceiver station (BTS), access point, or satellite) that establish and control the connections and interfaces between networks and devices. Requests from users are transmitted to the central processors that are connected to servers. After that, the subscribers' requests are delivered to a cloud through internet. In the cloud request processing is done and then results are provided back to mobile devices.

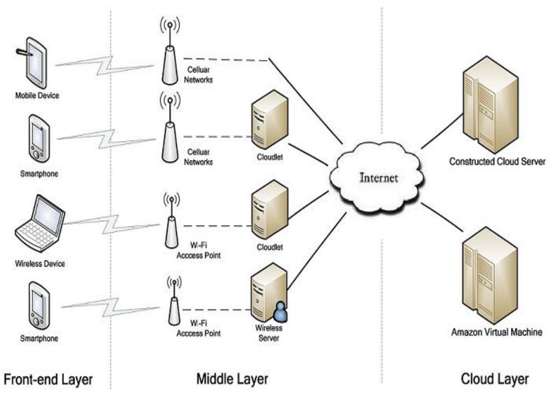

Fig 1. Mobile cloud computing architecture

Mainly there are three service models. **1.)*Cloud Software as a Service (SaaS)*** is the capability provided to the consumer to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a Web browser (e.g., Web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings. **2.)*Cloud Platform as a Service (PaaS)*** is the capability provided to the consumer to deploy onto the cloud infrastructure consumer-created or -acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations. **3.)*Cloud Infrastructure as a Service (IaaS)*** is a capability provided to the consumer to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

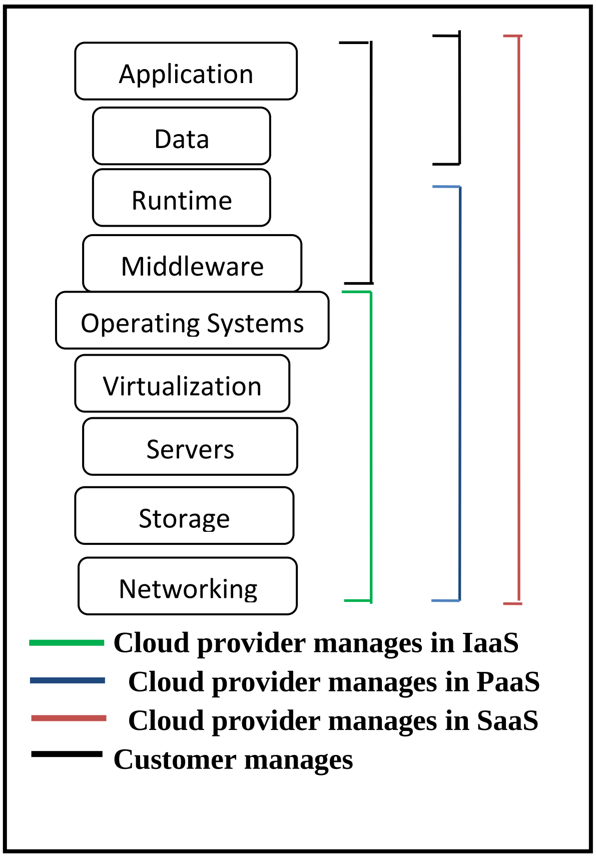

Fig 2. Responsibilities in accessing cloud using various service models

2. MOBILE CLOUD COMPUTING :BACKGROUND

Nowadays, both hardware and software of mobile devices get greater improvements than before, some smartphones such as iPhones, Android serials, window mobile phones and blackberry, are no longer just traditional mobile phones with conversation, SMS, Email and website browser, but are daily necessities to user. However at any given cost and level of technology, considerations such as weight, size, battery life, ergonomics and heat dissipation exact a severe penalty in computational resources such as processor speed, memory size, and disk capacity. Therefore three approaches have been proposed for mobile cloud applications:

1. Extending the access to cloud services to mobile devices. In this approach users use mobile devices often through web browsers, to access software/applications as services offered by cloud. The mobile cloud is most often viewed as a Software-as-a-service (SaaS) cloud and all the computation and data handling are usually performed in the cloud.
2. Enabling mobile devices to work collaboratively as cloud resource providers. This approach makes use of the resource at individual mobile devices to provide a virtual mobile cloud, which is useful in an ad hoc networking environment without the use of internet cloud.
3. Augmenting the execution of mobile applications on portable devices using cloud resources. This approach uses the cloud storage and processing for applications running on mobile devices. The mobile cloud is considered as an Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS) cloud. In this partial offloading of computation and data storage is done to clod from the mobile devices.

In next section we will see different approaches for dealing with computation intensive applications which are still challenging for executing at mobile side.

3. LITERATURE SURVEY

(i) Collaboration among mobile devices: As the mobile devices have certain resource constraints, there arises a need to get resources from external sources. One of the ways to overcome this problem is getting resources from a cloud, but the access to such platforms is not always guaranteed or/and is too expensive. Huerta-Canepa in [6] presents the guidelines for a framework that mimics a traditional cloud provider using mobile devices in the vicinity of users. The framework detects nearby nodes that are in a stable mode, meaning that will remain on the same area or follow the same movement pattern. If nodes in that state are found, then the target provider for the application is changed, reflecting a virtual provider created on-the-fly among users. In scenarios like downloading a description file at a museum, collocation increases the chances of people willing to perform common tasks[7]. To save the resources like energy and processing power, the collocated mobile devices can collaboratively act as a local cloud and split the task into smaller subtasks to be performed on different devices[8]. The results can then be aggregated and shared. The proposed approach allows avoiding a connection to infrastructure-based cloud providers while maintaining the main benefits of offloading.

Fernando et'al in [9] on the other hand propose to use all kinds of local resources (smartphones, PDA, even computers) to be used to collaborate in forming the local cloud to achieve a common goal. Their approach is to overcome the resource sparseness, energy consumption and low connectivity problems[10] faced in traditional mobile cloud computing. Sharing of workload is dynamic, proactive and depends on cost model to benefit all participants. The architecture consists of mainly a Resource Handler, a Job Handler and a Cost Handler. The resource handler discovers the collocated resources, the cost handler then calculates the costs to see what distribution of jobs will have most benefits and then the job handler distributes[11] the sub-tasks, run the jobs and collect them back on sender. Finally the cost handler handles micropayments among the participating devices.

SpACCE concept in [12] providing calculation capacity of PCs is proposed to facilitate distributed collaboration. A SpACCE is a sophisticated ad hoc cloud computing environment that can be built according to the needs that occur at any given time on a set of personal, i.e., non-dedicated, PCs and dynamically migrate a server[13] for application sharing to another PC. By migrating the server, redundant calculation capacity of PCs can be utilized for creating a SpACCE, where the response time of the application shared among users is improved.
A SpACCE provides the available calculation capacity of a PC as the server for collaboration to other PCs which have no application and/or not enough calculation capacity to be the server on demand.
.

Table 1. Comparison of approaches related to collaboration among mobile devices

Name of approach	Job Distribution Time	Performance matrix	Constraints	Applications used	Advantages	Disadvantages
Virtual Cloud Computing Framework	Static	Energy consumed	1. Sharing can't be done with relatively moving devices. 2. Basic framework with no cost consideration.	OCR (Optical Character Recognition) Software	1. Lightweight architecture. 2. Ad hoc 3. Much lesser energy consumed.	No fault tolerance. Very basic framework.
Ad hoc and Opportunistic Job Sharing	Opportunistic /Dynamic	Cost and device capabilities	Devices should be in close proximity.	Speech recoginition and synthesis	1. Benefit to all participants. 2. Ad hoc 2. Includes all types of local resources.	No fault tolerance
SpACCE	Dynamic	Calculating capacity of PCs	1. Servers must have more than 50% calculating capacity available. 2. Need of some existing network infrastructure.	CollaboTray[14]	1. Server can be migrated according to available calculating capacity. 2. Even with no high-spec PC, acceptable response time is maintained.	Required network infrastructure.

Each PC in this environment can become the server and client according to its available calculation capacity for the applications running. The level of server's available calculation capacity decides the migration of server to take place. For distributed collaboration occurring ad hoc, the migration of a server is executed with no management mechanism for application sharing.

In the scenario shown below, any one of PC1, PC2 or PC3 can become server and PC4 is capable of being a client only. Currently, PC1 is acting as a server which is serving all the clients namely PC2, PC3 and PC4. When PC1 feels a degree of latency in the application, the server can be migrated to either PC2 or PC3.

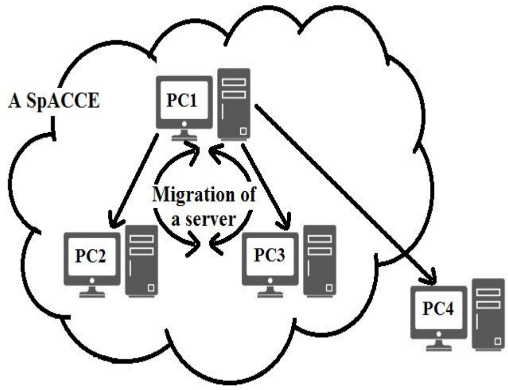

Fig 3. SpACCE architecture

(ii) Migrating execution from mobile devices to resource rich platform: Collaboration of mobile devices to work as a unit in a networked environment is a good solution for a common task.But sometimes work cannot be distributed among mobile devices and has to be offloaded to a resource rich platform. For that migration of executable block has to be done[15],[16],[17]. Ricky et'al in [18] has proposed stack-on-demand asynchronous exception (SOD_AE) execution mechanism for offloading of work to a nearby cloud. In this mechanism, a stack is being maintained for the storage of execution state and only the recent execution state that is on top of the runtime stack will be migrated. So in this approach no matter how big the process image is, SOD migrates only the required part of the data to the destination site. Capturing states in mobile devices in a portable manner has been done using asynchronous exception and is stored using Twin Method Hierarchy approach in order to minimize the overhead. However offloading to a distant cloud introduces latency as a factor.

A cloudlet [19] architecture proposed by M.Satyanarayan, advocates a two tier approach to decrease the latencies. Proposed architecture states that rather than relying on a distant "cloud", we might be able to address the mobile device's resource poverty via a nearby resource-rich cloudlet. Cloudlets are decentralized and widely dispersed Internet infrastructure components whose compute cycles and storage resources can be leveraged by nearby mobile computers. Access to a cloudlet can be provided by Wi-Fi that saves energy as well as has greater bandwidth as compared to other internet services.

Table 2. Comparison of approaches related to migration

Name of Approach	**Resource-rich platfrom**	**Migrated data**	**Internet Services**	**Proposed concept**	**Advantages**	**Disadvanteages**
SOD_AE	Cloud Provider	Only state at that time stored in stack	3G	Proposed to migrate less data as only by sending state stored at given time	1).Less data to migrate. 2).Java language increases code mobility.	1).Migration of all the tasks.
VM-based Cloudlets	Cloud Provider and Cloudlets	Computation intensive threads	3G for Cloud and Wi-Fi for cloudlet	Proposed to rather than relying on distant cloud try to offload to a nearby cloudlet.	1).Offloading to cloud reduces delay and energy consumed.	1).Response time increases if Coudlet denies the service. 2).No criteria for defining whether to offload or not.
HYRAX	Resource-rich nearby Computer	Work broken in parts	Networked	Break work in a components and distribute to slaves(mobile devices) via Master(Personal computer) node in the network	1).Better for data processing. 2).Work is under a centralised controller.	1).Not good for computation intensive. 2).No cloud provider for computation tasks.

Hyrax is also very similar to this type of concept, proposed by E. Marinelli in [20]. This architecture deploys mobile devices as nodes to create a mobile cloud computing platform. In order to enhance the performance of Hyrax, an extended version of Hadoop[21], mobile devices act as slave but master is still deployed on a PC (resource rich platform in contrast with smartphones). Distributed data processing is provided via Hadoop's MapReduce implementation, which divides jobs submitted by the user into independent "tasks" and distribute these tasks to slave nodes. The architectural model for Hyrax is shown in the figure below.

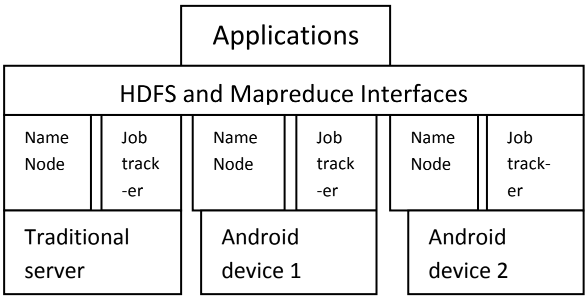

Fig 4. Hyrax architecture

(iii) Augmented Execution: In recent years, researchers have explored an era, in which offloading is being done partly to the cloud and rest is completed at mobile side, gives better results [23],[24],[22],[27]. B.Chun in [22] has defined an approach CloneCloud, with aim of offloading execution blocks from mobile device to the cloud dynamically to modify the execution performance of a mobile device. Approach describes that clone are made at the cloud side at each initiation of a service, that are mirror image of the smartphone. In contrast with Smartphones, clones are resource rich and do not have the battery constraint as well. Major advantage of the CloneCloud implementation is stated as the performance enhancement. Chun has taken Virus scanning, image search and behaviour profiling applications that are computation intensive for performance evaluation. Some considerations are also there as application control can be at either entry level or at exit level only. Also native methods cannot be migrated.

Another related approach is being proposed by L.Yang in [25], which performs the offloading decision dynamically based on the resources available at mobile device. This approach is based on elasticity of an application, which states that component can be offloaded to cloud and vice versa at any particular time.

Table 3. Comparison of approaches related to partitioning and offloading

Name of approach	**Partitioning Time**	**Performance matrix**	**Constraints**	**Applications used**	**Advantages**	**Drawbacks**
CloneCloud	Offline	Total execution time and energy expanded	1.Native methods cannot be offloaded. 2.Methods that access special features cannot be offloaded(e.g. camera).	Virus Scanning, Image processing, Behaviour modelling	1.Better execution results 2.Less energy consumed	Does not virtualize access to native resources Processing can get blocked if some thread is offloaded
Application partitioning problem for mobile datastream applications	Adaptive	Throughput	1.Maximum of computation time and communication time will be taken for throughput. 2.All components are independent	Tasks with different computation to communication ratio.	1.Throughput achieved is about 2X.	Energy consumption not being taken into account. Resources at cloud end are assumed to be abundant.
MACS	Adaptive	Execution time and Energy consumed(in joules)	1.Memory cost of resident service cannot be more than available memory on mobile device. 2.Energy consumption of offloading should not be greater than not offloading 3.Execution time at cloud should not be greater than execution time at mobile.	N-Queens problem and face recognition	Better cost function (consists of cost of transfer, cost of memory, cost of execution) in contrast with locally execution.	This approach is lagging in parallelism between threads.

Yang has advocated that the accuracy of many mobile data stream applications such as face/gesture recognition is determined by its throughput. The application can be broken into a number of independent components such that each component can be executed at cloud as well as mobile device without causing any blocking in the execution of the complete application. To determine throughput of an application the critical component is chosen from all the components in which an application can be divided. The component that is taking maximum time is the deciding component.

An approach Mobile Augmentation Cloud Services (MACS), proposed by D. Kovachev in [26], based on the adaptive computation and elasticity of executing blocks. MACS application consists of an application core (Android activities, GUI, access to device's sensors) which cannot be offloaded, and multiple services that encapsulate separate application functionality (usually resource-demanding components) which can be offloaded. Therefore partitioning is done taking considerations of such applications. The partition consists of a binary string that is a combination of 0s and 1s. If some component is corresponded by 0, that means component cannot be offloaded to cloud and if corresponded by 1 then that is offloadable. Some constraints are there to check on the components whether offloading will be beneficial or not according to binary string. Performance evaluation is done using N-Queens problem and Face recognition.

4. OPEN ISSUES

(i) Task division: It is been found that classifying the tasks or applications from mobile devices into multiple sub-tasks / modules and delivering some of them to run on cloud, can be an intelligent approach to the resource limited mobile devices. However, there is still a scope of improvement in the form of an optimal, effective strategy or algorithm on how to classify these tasks and modules and which module should be processed by cloud and which one by Mobile devices.

(ii) Quality of Service (QoS): When a mobile user need to access any services or resources then he needs to request to servers located in a cloud. In this case, the mobile users may face some issues such as congestion due to wireless bandwidths, network disconnection, and the signal attenuation caused by mobile users' mobility. Elements of network performance within the scope of QoS often include availability (uptime), bandwidth (throughput), latency (delay), and error rate and to overcome all these factor new research directions are expected.

(iii) Data delivery: It is analyzed that due to the feature of resource constrained, mobile devices such as PDAs in terms of memory, processing power, battery lifetime and screen size are vital point of concern. Applications for such devices need to be resource conserving and lightweight enough to achieve a level performance deemed usable. The application programmers also need to take into account the strain put on these resources during execution time, and there are often tradeoffs to be made as to where to execute processes and store information, whether it be locally on the mobile device or remotely on a more powerful device.

(iv) Low Bandwidth: As we have seen that many research scholars has propose the optimal and efficient way of bandwidth allocation or the bandwidth limitation is still a big concern because the number of mobile and cloud users is dramatically increasing. And to improve the bandwidth limitation the emerging technologies such as 4G network are used to overcome the limitation and bring a revolution in improving bandwidth.

(v) Architectural issues: A reference architecture for heterogeneous MCC environment is a crucial requirement for unleashing the power of mobile computing towards unrestricted ubiquitous computing.

(vi) Context-awareness issues: Context-aware and socially-aware computing are inseparable traits of contemporary handheld computers. To achieve the vision of mobile computing among heterogeneous converged networks and computing devices, designing resource-efficient environment-aware applications is an essential need.

(vii) Live VM migration issues: Executing resource-intensive mobile application via Virtual Machine (VM) migration-based application offloading involves encapsulation of application in VM instance and migrating it to the cloud, which is a challenging task due to additional overhead of deploying and managing VM on mobile devices.

(viii) Energy-efficient transmission: MCC requires frequent transmissions between cloud platform and mobile devices, due to the stochastic nature of wireless networks, the transmission protocol should be carefully designed.

5. CONCLUSION

As mobile devices have become our primary data processing devices nowadays, mobile cloud computing has emerged as a great extension to cloud computing field. In this paper, we present an in-depth survey of research work done in mobile cloud computing. Open issues have also been covered, with some primary issues being discussed along with the research done around them. Section III contains the detailed survey around the key categories of mobile cloud computing which points out at some of the approaches focusing on collaborative working of mobile devices, migrating the execution from mobile devices to resource rich platforms and partitioning of applications for offloading them to the cloud. We have also concluded that for a number of applications local resources are not sufficient to execute on mobile device.

We conclude that the available local resources of a group of devices residing in the same area can be used to form a virtual cloud to overcome the resource constraints of our mobile devices. This way, need of internet availability can also be suppressed. In computation intensive applications, sometimes the local resources cannot provide enough support to deliver the required quality of service. Such applications can be migrated to be executed on cloud. Application partitioning approaches can also be used to augment the execution of certain mobile applications on cloud resources. While some of the discussed approaches might seem quite complicated, these fields offer some promising scope of research for future.

6. REFERENCES

[1] M. Satyanarayan, "Fundamental Challenges in Mobile Cloud Computing," Proceedings ACM Symposium Principles of distributed Computing, ACM Press, 1996.

[2] N. Tolia, M. Kaminsky, D. G. Andersen and S. Patil. An Architecture for Internet Data Transfer. In NSDI, 2006.

[3] A. Weiss. Computing in the Clouds. NetWorker, 11(4):16-25,Dec 2007.

[4] A. Zahariev, "Google App engine," Helsinki University of Technology, 2009.

[5] Wikipedia, "Mobile phone – Wikipedia , the free encyclopedia, http://en.wikipedia.org/wiki/Mobile_phones.

[6] Gonzalo Huerta-Canepa, Dongman Lee, "A Virtual Cloud Computing Provider for Mobile Devices", *ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond. MCS'10*, 2010.

[7] J. Kangasharju, J. Ott, and O. Karkulahti, "Floating Content: Information Availability in Urban Environments," *Proceedings of the 8th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom) (WiP)*, Mannheim, Germany: 2010.

[8] A. Berl, H. Meer, H. Hlavacs, and T. Treutner, "Virtualization in energy-efficient future home environments," *IEEE Communications Magazine*, vol. 47, 2009, pp. 62-67. December 2009.

[9] Niroshinie Fernando, Seng W. Loke, Wenny Rahayu, "Dynamic Mobile Cloud Computing: Ad Hoc and Opportunistic Job Sharing", *Fourth IEEE International Conference on Utility and Cloud Computing*, Victoria, NSW: December 2011.

[10] M. Satyanarayanan. "Fundamental challenges in mobile computing." In *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, PODC '96, pages 1–7, New York, NY, USA, 1996.

[11] K. A. Hummel and H. Meyer. "Self-organizing fair job scheduling among mobile devices". In *Self-Adaptive and Self-Organizing Systems Workshops, 2008. SASOW 2008. Second IEEE International Conference on*, pages 230–235, Venice: October 2008.

[12] Tatsuya Mori, Makoto Nakashima, and Tetsuro Ito, "A Sophisticated Ad Hoc Cloud Computing Environment Built by the Migration of a Server to Facilitate Distributed Collaboration", in *26th International Conference on Advanced Information Networking and Applications Workshops*, pages 1196-1202, 2012.

[13] Z. Zhang, J. Bu, C. Chen, and J. Bo, "A new collaboration architecture for highly organized work group," *Proc. the 2008 Fourth International Conference on Networked Computing and Advanced Information Management (NCM '08)*.

[14] Y. Abe, K. Matsusako, K. Krimura, M. Nakashima, and T. Ito, "Tolerant sharing of a single-user application among multiple users in collaborative work," *Companion Proc. the ACM Conference on Computer-Supported Cooperative Work (CSCW'10), ACM Press*, 2010.

[15] R. F. Lopes, and F. J. D. S. E. Silva. "Migration Transparency in a Mobile Agent Based Computational Grid." In Proc. Of the 5th WSEAS Intl. Conf. on Simulation, Modelling and Optimization, pp. 31-36, August 17-19, 2005.

[16] E.Truyen, B.Robben, B.Vamhaute, T.Coninx ,W.Joosen, and P. Verbaeten, "Portable Support for Transparent Thread Migration in Java." In Proceedings of 2nd International Symposium on Agent Systems and Applications and 4th International Symposium on Mobile Agents 2000. Zurich, Switzerland, Sept. 13-15, 2000.

[17] T.Sakamoto, T. Sekiguchi, and A. Yonezawa, "Bytecode transformation for portable thread migration in java," In proceedings of 2nd international Symposium on Mobile Agents, Zurich, Switzerland, Sept. 13-15, 2000.

[18] Ricky K.K. Ma, Cho-Li Wang, "Lightweight Application-level Task Migration for Mobile Cloud Computing ." In Proceedings of 26th IEEE International Conference on Advanced Information Networking and Applications, 2012.

[19] M. Satyanarayan, P. Bahl, R. Caceres, and N. Davies. "The Case for VM-based Cloudlets in Mobile Computing," IEEE Pervasive Computing, 8(4), 2009.

[20] E. Marinelli, "Hyrax: Cloud Computing on Mobile Devices using MapReduce." Thesis Report to School of Computer Science, Pittsburgh, PA. Sept 2009.

[21] Hadoop apache, Borthakur[2007].

[22] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. "CloneCloud: Elastic execution between mobile device and cloud," In Proceedings of EuroSys 2011.

[23] R. K. Balan, M. Satyanarayan, S. Park, and T. Okoshi. "Tactics-based remote execution for mobile computing. In MobiSys, 2003.

[24] E. Cuervo, A. Balasubramaniam, D. Cho, A. Wolman, S. Sariou, R. Chandra. And P. Bahl. "MAUI: Making smartphones last longer with code offload. In MobiSys, 2010.

[25] L. Yang, J. Cao, S. Tang, Tao Li, Alvin T. S. Chan, "A framework for Partitioning and Execution of Data Stream Application in Mobile Cloud Computing." IEEE Fifth International Conference on Cloud Computing, 2012.

[26] D. Kovachev, Tian Yu and Ralf Klamma. "Adaptive Computation Offloading from Mobile Devices into the Cloud." 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, 2012.

[27] K. Kumar and Yung-Hsiang Lu, "Cloud Computing for Mobile Users: Can Offloading computation saves Energy?." IEEE Computer Society. April 2010.