Volume 5, No. 5, May-June 2014

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Mobile Cloud Computing: concept, research trends and open issues

Gaurav
Department of CSE
DCRUST, Murthal Haryana, India

Nitesh Kaushik
Department of CSE
DCRUST, Murthal Haryana, India

Abstract: In this modern era of ubiquitous computing, mobile devices have become a vital part of our day-to-day chores. With the continuous evolution of mobile applications and advantage of portability, these devices have become our primary information processing devices. Despite the ever increasing technological advances, these devices are still not able to go neck-to-neck with the traditional computing infrastructure (such as personal computers, laptop etc.). So, researchers have been stressing upon the resource constraints of mobile devices since long. Mobile Cloud Computing can be seen as an appropriate solution to this problem because of its features like instantaneous scalability, on-demand services, ubiquitous network access. It has been a reverberation in technological scenario in the recent past. This paper presents a review on basic principles, characteristics and the recent research trends in the field.

Keywords: mobile cloud computing, collaborative computing, partitioning, off loading.

I. INTRODUCTION

Because of ever increasing usage of mobile device applications in our daily life, the devices like smartphones, PDA, tablets, palmtop etc. have become a must for almost everyone. But because of the small size and portability, these devices put forward certain restrictions on the resources to be put in them. Due to these resource constraints, they are still mobile phones are not capable of running all type of applications on its own without any constraint [1], [5]. Manufacturers in the industry are constantly coming up with improvements in battery life, storage capability, weight, computational capability etc. to cop-up with the computation-intensive applications but the restrictions such as size, weight and cost make huge bumps in the improvements infeasible. So, Mobile Cloud Computing can be seen as a life-saver. 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 fig. 1 shows, the mobile devices are connected to the base stations which in turn connect the devices to the mobile networks. These base stations (e.g., base transceiver station (BTS), access point, or satellite) establish and manage the connections and interfaces between the mobile networks and devices. User requests are sent to the servers via central processors. Then, the requests of the subscribers' reach a cloud infrastructure using internet. Afterwards, the cloud processes the request and the results are transmitted back to the respective mobile devices.

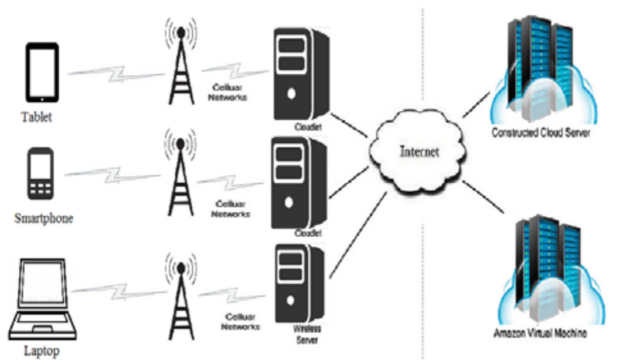

Figure 1. Architecture of Mobile Cloud Computing

The services offered by cloud computing can be broadly classified into 3 models:

a. ***Cloud Software as a Service (SaaS):*** It is the service offered to the consumers for using the service provider's applications running on the cloud infrastructure. These applications can be accessed from a number of client devices via a thin client user interface like a Web browser (e.g., Web-based email). The consumers do not need to manage/control the underlying infrastructure which includes the cloud network, servers, operating systems, storage, or even independent application properties, with some possible exceptions such as the user-specific application configuration settings.

b. ***Cloud Platform as a Service (PaaS):*** It is the service offered to the consumers to install consumer-specific applications on the top of the cloud infrastructure. These applications can be created using programming languages and tools supported by the service provider. The consumers do not need to manage the underlying cloud infrastructure which includes the network, servers, operating systems, or storage, but have to control the self-installed applications and possibly application hosting environment configurations.

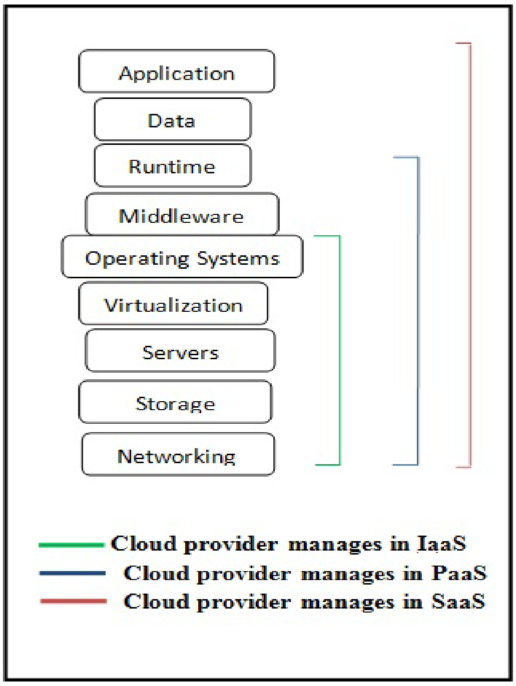

Figure 2. Responsibilities in accessing cloud using various service models.

c. ***Cloud Infrastructure as a Service (IaaS)***: It is a service offered to the consumers to provision processing, storage, networking, and rest of the fundamental computational resources where the consumers are allowed to install and execute arbitrary software, which include operating systems and applications. The consumers do not need to manage the underlying cloud infrastructure but have the control over operating system, storage, installed applications, and possibly limited control of select networking components (e.g., host firewalls).

II. BACKGROUND - MOBILE CLOUD COMPUTING

Nowadays, both hardware and software of mobile devices get greater improvement than before, some smartphones such as iPhone, 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:

a. 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 All the computation and data handling are usually performed in the cloud.
b. 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.
c. 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.

III. LITERATURE SURVEY

a. ***Collaboration among mobile devices:*** As the mobile devices are resource poor in comparison of our personal systems, a need to get resources from external sources is felt for working efficiently. One way to overcome this problem is using resources of a cloud, but access to such platforms is not always guaranteed or/and is often very expensive. Huerta-Canepa in [6] puts forward the guidelines for a framework that acts as a traditional cloud provider using mobile devices in the close proximity of user's device. The framework detects nearby nodes that are either stationary or moving as a group, means they remain within each other's range or follow the same movement pattern until the work is not completed. If a number of nodes are found in that state, then the target provider for the application is changed, resulting in the creation of a virtual cloud on-the-fly among users. In scenarios like downloading an introduction file at a museum, collocation increases the chances of people willing to perform same task[7]. To save the resources like energy and processing power, the collocated mobile devices can collaboratively work as a local cloud and split the task into smaller subtasks to be assigned to all devices for execution[8]. The output can be aggregated and shared. The proposed approach helps avoiding the use of infrastructure-based cloud providers while keeping the main benefits of offloading on offer.

Fernando et al in [9] on the other hand present an approach to use local resources (smartphones, PDA, even computers) for collaboratively forming a local cloud and achieve a common goal. Their approach is to overcome the resource sparseness, energy utilization and network problems[10] faced in traditional mobile cloud computing. Workload sharing is dynamic, proactive and depends on cost model for offering benefits to all participants. A Resource Handler, a Job Handler and a Cost Handler primarily constitute the proposed architecture. The resource handler discovers the collocated resources, and then cost handler calculates the costs to find the best distribution of jobs that has most benefits and then the job handler distributes[11] the sub-tasks, run the jobs and collect results back on sender. Finally cost handler handles micropayments among the participating devices.

A distributed collaboration approach is proposed as SpACCE concept in [12], providing the available calculation capacity of participating PCs as the server. A SpACCE is a sophisticated ad hoc cloud computing environment that can be implemented according to the needs that occur at a particular 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 making a SpACCE, where the response time of the shared application can be enhanced.

A SpACCE facilitates the available calculation capacity of a PC as the server for collaboration to other PCs that have no application and/or not enough calculation capacity to be the server when needed. Each PC in this network can act as server or client according to its remaining calculation capacity for the applications in progress. The level of server's available calculation capacity becomes the deciding factor for the migration of server to take place. For ad hoc distributed collaboration, the migration of a server is executed with no management mechanism for application sharing.

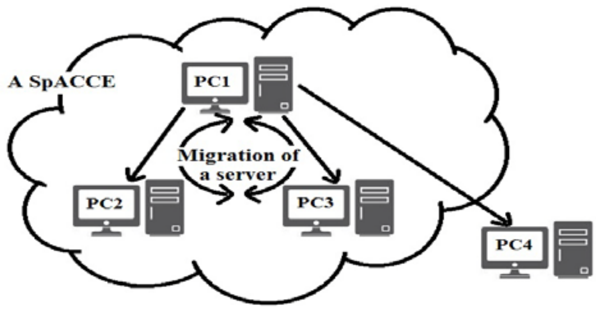

Figure 3. SpACCE architecture

In fig. 3, one of PC3, PC2 or PC1 can act as server and PC4 can become a client only. Apparently, PC1 becomes the server which is serving the rest namely PC2, PC3 and PC4. When PC1 observes a some latency or lag in the application, the server can be migrated to either of the PC2 or PC3.

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

Name of approach	**Job DistributionTime**	**Performance matrix**	**Constraints**	**Applications used**	**Advantages**	**Disadvantages**
Virtual Cloud Computing Framework	Static	Energy consumed	1. Sharing can be done with relatively static devices. 2. Framework doesnot contain a cost model.	OCR (Optical Character Recognition) Software	1. Lightweight architecture. 2. Ad hoc 3. Better energy utilization.	No criteria for fault tolerance. A basic framework.
Ad hoc and Opportunistic Job Sharing	Opportunistic/ Dynamic	Cost and execution capabilities of device	Devices should remain in close proximity.	Speech recoginition and synthesis	1. Win-win situation for all participants. 2. Ad hoc 2. Includes all types of local resources.	fault tolerance is not addressed.
SpACCE	Dynamic	Calculating capacity of participating PCs	1. Available calculating capability of servers must be more than 50%. 2. Some established network infrastructure is needed.	CollaboTray[14]	1. Aavailable calculating capacity decides setrver migration. 2. Acceptable response time is maintained even without any high-spec PC.	Required network infrastructure.

b. Migrating execution from mobile devices to resource rich platform: Integration of mobile devices to work collaboratively in a networked environment poses as a great option for a common task. But in certain cases, work cannot be distributed to a number or nodes but has to be offloaded to some resource rich platform. For that offloading 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. This mechanism maintains a stack for storing execution state and migrates only the latest execution state, i.e., the top of the runtime stack. So in this approach no matter what size the process image is, SOD migrates only the required part to the cloud. States of mobile devices in a portable manner are captured using asynchronous exception and stored using Twin Method Hierarchy approach so as to reduce the overhead. However offloading to a distant cloud causes latency overhead.

A cloudlet [19] architecture proposed by M.Satyanarayan, states a two tier approach to reduce the latencies. Proposed architecture states that instead of relying on a distant resource provider, we might be able to get rid of mobile device's resource poverty by using a nearby resource-rich cloudlet. Cloudlets are decentralized and widely spread Internet infrastructure components offering compute cycles and storage resources to be used by nearby mobile computing devices. A cloudlet can be accessed using Wi-Fi which reduces energy consumption as well as has better bandwidth as compared to other internet services. Hyrax, proposed by E. Marinelli in [20] is also quite a similar concept. This architecture comprises of mobile devices acting as nodes in order to create a mobile cloud computing platform. In order to enhance the performance of Hyrax which is an extended version of Hadoop[21], the mobile devices act as slaves and master is deployed on a PC (resource rich platform as compared to smartphones). Distributed data processing is facilitated using Hadoop's MapReduce implementation, which breaks down jobs submitted by the user into independent “tasks” and distributes these tasks among the slave nodes. In the following diagram, first the applications get divided and then are given to mobile devices using interfaces.

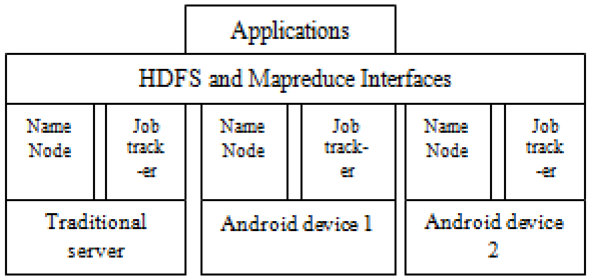

Fig 4. Hyrax architecture

Table 2. Comparison of approaches related to migration

Name of Approach	**Resource-rich platfrom**	**Migrated data**	**Internet Services**	**Proposed concept**	**Advantages**	**Disadvantages**
SOD_AE	Cloud Provider	Only instantaneous state stored in stack	3G	Proposed to migrate less data with only by sending state stored at given time	1).Lesser data migration. 2).Java 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	Try to offload to a nearby cloudlet. rather than relying on distant cloud.	1).Offloading to cloudlet reduces delay and energy consumed.	1).Response time increases if Cloudlet denies the service. 2).No criteria for defining whether to offload or not.
HYRAX	Resource-rich nearby Personal Computer	Subtasks to each participating device.	Networked	Break work in a components and distribute among slaves(mobile devices) via Master(Personal computer) node in the network.	1).Better for data processing. 2).Work is under a centralized controller.	1).Not good for computation intensive. 2).No cloud provider for computational tasks.

c. ***Augmented Execution:*** Recently, researchers have explored an era, in which applications are partly offloaded to the cloud and rest of the application is executed at mobile device, gives better results [23],[24],[22],[27]. B. Chun in [22] has stated an approach CloneCloud, with aims to offload execution blocks from mobile device to the cloud dynamically for enhancing the execution performance of a mobile device. The approach describes that clones of applications are made at the cloud side after each initiation of a service, which are almost the mirror images of the smartphone. In contrast with Smartphones, clones are deployed in a platform that is resource rich which do not have constraints like battery as well. Major advantage of the CloneCloud is said to be the performance improvement. Chun has taken applications such as Virus scanning, image search and behaviour profiling applications for performance evaluation as these are computation intensive applications. Also, there are some considerations as the application control can either be at entry level or at exit level. And, native methods cannot be offloaded.

Another related approach is being proposed by L.Yang in [25], which performs the offloading decision based on the resources available at mobile device at a particular time. This approach is based on elasticity of an application, which enables component offloading to cloud and vice versa at any particular time. Yang has advocated that the accuracy of many mobile data stream applications such as face/gesture recognition can be measured by its throughput. Such applications can be divided into a number of independent components such that each component can be offloaded to cloud fro execution as well as mobile device without blocking 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 to execute is the deciding component. Another 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 are native and cannot be offloaded, and multiple services that encapsulate separate application functionality (usually resource-demanding components) which are offload able. 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 is not offloadable to cloud and if corresponded by 1 then that can be offloaded. Some conditions are there to check for components whether offloading is beneficial or not according to binary string. The performance is evaluated using N-Queens problem and Face recognition and shows upto 95% better results in terms of energy than executing at mobile device only

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.

IV. OPEN ISSUES

- ***a.*** ***Task division:*** Over the period, it is observed that breaking the mobile applications into multiple sub-tasks / modules and delivering some of them to be executed on cloud resources, can be an efficient approach for the resource-poor mobile devices. However, there is still some scope of improvement in terms of an optimal, effective strategy or algorithm on how to breakdown these tasks or modules and which module should be processed by cloud and which one by mobile devices.
- ***b.*** ***Quality of Service (QoS):*** When a mobile user needs to access the services or resources offered by cloud, then he/she needs to request the servers residing in a cloud. In such scenario, the mobile users may face some issues like congestion due to low bandwidth of wireless networks, frequent network disconnection and the signal attenuation caused by the users' mobility. Performance parameters of network 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.
- ***c.*** ***Data delivery:*** It is observed that because of the resource constraints the memory, processing power, battery lifetime and screen size are vital point of concern in case of mobile devices like smartphones and PDAs. Applications for such devices should be good at resource optimization and light enough for achieving a level of performance that is deemed practically usable. The application programmers should also consider the strain put on these resources in the execution time, and there are often some tradeoffs to be made such as where to execute processes and store information, whether it is local resources of mobile or remotely on some more powerful device.
- ***d.*** ***Low Bandwidth:*** As it is found that a number of researches have proposed the optimal and efficient way of allocating bandwidth. Or we can say, the bandwidth limitation is still a big concern due to the dramatically increasing number of mobile and cloud users. And, in order to improve the bandwidth allocation, the emerging technologies such as 4G network should be used to overcome the problem and bring a revolution in bandwidth improvement.
- ***e.*** ***Architectural issues:*** A standard architecture for heterogeneous Mobile Cloud Computing environment is a critical need for unleashing the power of mobile computing in the direction of unrestricted and ubiquitous computing.
- ***f.*** ***Context-awareness issues:*** Context-awareness and socially-aware computing are indisputable traits of contemporary handheld computing devices. Designing resource-efficient environment-aware applications is an essential need for achieving the vision of mobile computing among heterogeneous converged networks and computing devices.
- ***g.*** ***Live VM migration issues:*** Running the resource-intensive mobile applications via VM migration-based offloading involves encapsulation of application in VM instance and migration to the cloud. It is a challenging task due to additional overhead of deployment and management of VM on mobile devices.
- ***h.*** ***Energy-efficient transmission:*** Mobile Cloud Computing requires regular interaction between cloud platform and mobile devices, due to the stochastic nature of wireless networks, the transmission protocol should be carefully designed.

V. CONCLUSION

As mobile applications have become the part of our daily routine nowadays, mobile cloud computing has evolved as an important branch of cloud computing.

This paper presents an in-depth survey of recent research trends in mobile cloud computing. It also discusses some open issues faced in the same along with the research done around them. The detailed survey of the main approaches of mobile cloud computing shows that collaboration among mobile devices, migration of task execution from mobile devices to resource rich platforms and partitioning of applications for offloading them to the cloud have resulted in pros for one terms and cons in other terms..

Also there is no practical realization of such type of applications till date. Every approach is put forward on conceptual basis. Also, there is no proper criteria for deciding migration, collaboration and offloading. It keeps changing from application to application. As there are no standard metrics in Mobile Cloud Computing for measuring the performance, some serious works regarding the standardization also needs to be carried out.

VI. 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, San Francisco, California, USA: June 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, Fukuoka: March 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), vol. 2, 2008.

[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, Greece, 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.