International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp. 9607-9615

Resource Augmentation for Mobile Devices in Mobile Cloud Computing Environments: A Review

[1]Jitender Kumar and [2]Amita Malik

[1, 2]Computer Science & Engineering Department, DCRUST Murthal, Sonepat, Haryana, India

Abstract-

Mobile devices with sophisticated hardware and software have emerged as the primary computing platforms for the day to day activities of the individuals. However, computation on these handheld devices is still inhibited by the limited battery power and limited resources. Mobile Cloud Computing (MCC) has been coined to augment the capability of such handheld devices. MCC empowers the application processing capabilities of the mobile devices by offloading computational intensive applications to resource rich distributed application processing systems. But the key challenge lies in the seamless distribution of the applications on the distributed application processing cloud platforms. Researchers have explored varied resources for the augmentation of mobile device capabilities. These resources range from the use of nearby idle computing resources to the public/private cloud resources. This paper investigates the existing research for the augmentation of mobile devices in MCC and presents the review of comprehensive existing work under two fundamentals perspectives: Provisioning of distributed application processing systems and Computation offloading. Furthermore, this paper also points out the associated issues and challenges from both perspectives.

***Index Terms*—Mobile Cloud Computing, Distributed systems, Computation offloading.**

INTRODUCTION

Advances in the hardware and software technology for the handheld mobile devices have shifted the computing preferences of users from the traditional standalone personal computers and laptops to the smartphones and tablets. Recent years have also witnessed a subsequent growth in the mobile apps for the day to day activities of individuals [1]. These apps can provide simple to sophisticated services to the users. The handheld devices although provide better portability and availability to the users but suffer from inherent limitations such as limited battery life and limited computing resources [2]. Applications like real time natural language processing, speech recognition, face recognition and augmented reality require very high computing power and storage which alternately proves to be too expensive for mobile devices in terms of compute resources and memory [3].

The possible solutions to augment the capabilities of mobile devices are: either to improve the hardware of mobile devices or to device some software level solution. Although hardware technology of mobile devices is advancing but the battery technology is not advancing at the same pace [4]. So, the alternative solution is to use the software level solution like Mobile Cloud Computing (MCC). MCC augments the application processing capabilities of mobile devices by offloading computational intensive application/parts to the resource rich distributed application processing systems [5]. It empowers the capabilities of mobile devices by conveying the services of cloud resources to the vicinity of mobile devices [6].

Although, the capabilities of mobile devices can also be enhanced by the leveraging the computational resources from the locally available stationary devices, but these devices can only be accessed when the users are present in the proximity of these devices [7]. However, cloud computing resources can be accessed from anywhere through on-demand access [8]. It can enable access to the shared cloud resources with least management efforts with pay per use way [9]. So, the use of cloud resources has been advocated as the best possible solution for the augmentation of mobile devices. The cloud resources can be leased from the public cloud service vendors like Amazon [10] or Google [11] or by some private vendors. Some of the popular cloud services are Amazon EC2, Google app engine, Google gmail etc.

Current mobile resource augmentation architectures employ various mechanisms for the execution of computational intensive tasks on cloud resources. Some strategies focused on the establishment of distributed application processing systems [12, 13] while others have focused on execution migration techniques [14]. Out of this vast amount of researches, some basic questions need to be answered. What are the various mechanisms through which the distributed application processing systems can be established? What are the significant characteristics of these distributed application processing systems? What are the mechanisms to delegate the code execution to these distributed application processing systems? A number of surveys have also been proposed for the mobile cloud computing [2, 5, 15]. However, different from these surveys this paper considers the establishment of distributed application processing systems and computation offloading jointly. The aim of this review paper is to investigate the technical background of current available solutions for the augmentation of mobile device capabilities and identify the challenges which need to be addressed for the future success of mobile cloud computing.

The paper is being organized as follows: Section 2 discusses the background of mobile cloud computing, Section 3 presents the distributed application processing systems provisioning techniques in mobile cloud computing, section 4 covers the enabling techniques for computation offloading, section 5 covers open challenges and section 6 covers concluding remarks.

BACKGROUND OF MOBILE CLOUD COMPUTING

The computing capabilities of the mobile devices have improved to a significant extent during the recent years. Now the mobile devices are no longer considered as just the voice communication devices, instead these devices have changed the computing world. These devices have now become the integral part for the day to day activities of the users. Although the ever evolving technology is dealing the intrinsic limitations of the mobile devices, but still these devices are not able to fulfill all the computational requirements of the users [16,17]. Some of the compute intensive applications like natural language processing, real time speech recognition, augmented reality etc. prove to be too expensive for mobile devices in terms of computational resources and battery life. Mobile cloud computing has been introduced to address these concerns. MCC is being proclaimed as the next generation solution for the mobile devices [3].

MCC is a paradigm for the enhancing of capabilities mobile devices via ubiquitous wireless access to the cloud computing resources [5]. Whereas, cloud computing is mode for enabling users to access shared pool of computing resources through network [8]. It is provided through four types of deployment models namely: public cloud, private cloud, community cloud and hybrid cloud [8]. The three key services delivered by cloud computing are: 1) infrastructure as a service, 2) platform as a service, and 3) software as a service. The deployment and service stack of cloud computing is shown in Fig. 1. MCC can leverage the benefits of software and infrastructure cloud services through computation offloading. However, the existing research has focused on different aspects of MCC for the augmentation of mobile device capabilities.

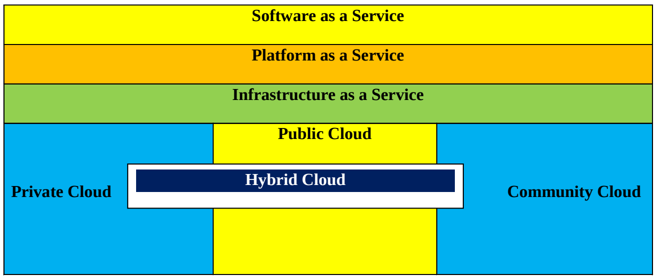

Figure. 1. Cloud computing deployment modes and service models.

Motivation:

Several inherent limitations of mobile devices are still inhibiting the feasibility of computational intensive application on these handheld devices. The vision of performing complex tasks like natural language processing, real time speech recognition can be realized on these handheld devices by leveraging the cloud resources [18, 19]. The various limitations that motivated the introduction of MCC are: Processing capabilities, limited storage, and limited battery of mobile devices.

Limited Processing capabilities: A lot of researchers [19-21] have identified that the mobile devices are still poor performers as compared to their desktop counterparts. Increasing in the clock speed of mobile device's CPU would drain more battery resources. These devices also lack internal RAM for storing the applications stack space.

Limited storage: The local storage of mobile devices is still significantly low as compared to the stationary personal computers or laptops. This limitation alternately forces the users to frequently delete their personal files and frequent installation and un-installation of various applications.

Limited Battery: Although the hardware of mobile devices is advancing but the battery technology is not advancing at the same pace. The number of transistors on an integrated circuit gets doubled in a period of approximately two years [4]. However, battery capacity is advancing at a rate of about 5 to 10% a year [22]. Moreover, increasing battery power may cause serious concerns to the user's safety [23]. The explosion of such heavy batteries can result in severe consequences to the human life.

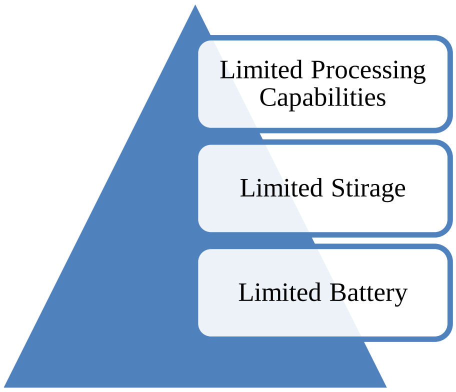

Figure. 2. Intrinsic challenges leading to the motivation of resource augmentation for mobile devices.

The above discussions led the researchers to think about some alternative solutions like mobile cloud computing. The next section have focused on the investigation of approaches for the augmentation of mobile devices and presents different approaches for providing distributed applications processing resources to the mobile devices for computation offloading.

Distributed Application provisioning systems in MCC

MCC resources provisioning and acquisition techniques have been characterized in diverse ways in the existing research. Every research focuses on different aspects of cloud resource provisioning techniques. Existing distributed application provisioning technique research can be broadly classified in three categories: 1) centralized cloud service datacenters, 2) distribution of cloud service datacenters in the geographical region called cloudlets, and 3) ad-hoc mobile clouds.

Centralized Cloud Service Datacenters

Centralized cloud services are very popular among the commercial cloud service providers like Amazon [10], Google [11] and open-source cloud service tools like Eucalyptus [24], openStack [25]. Fig. 4 shows the basic architecture of centralized cloud datacenters approach for distributed application processing. Centralized datacenters provide better scalability and availability to the infrequent requests from the mobile users. Furthermore, the centralized datacenters approach does not require the management and distribution of datacenters in particular geographical areas. The mobile device users can access centralized datacenter resources through 3G/4G cellular networks or Wi-Fi access points [5]. However, such systems suffer from the long term WAN latencies [27]. These long latencies affect the user experience as well as energy consumption on the mobile device.1

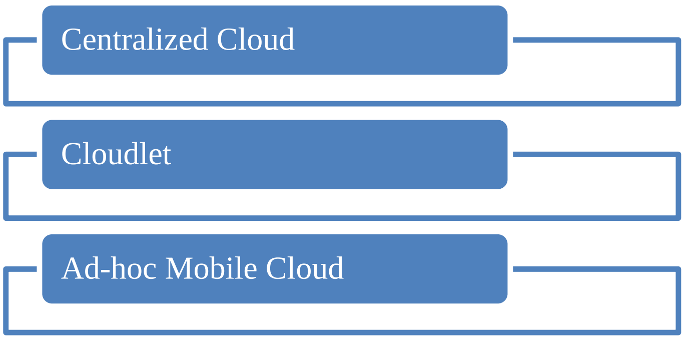

Figure. 3. Types of Distributed application resource provisioning systems.

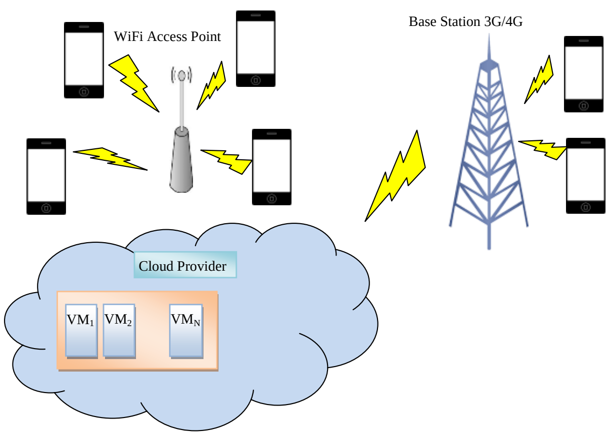

Figure. 4. Architecture of Centralized datacenter for distributed application processing.

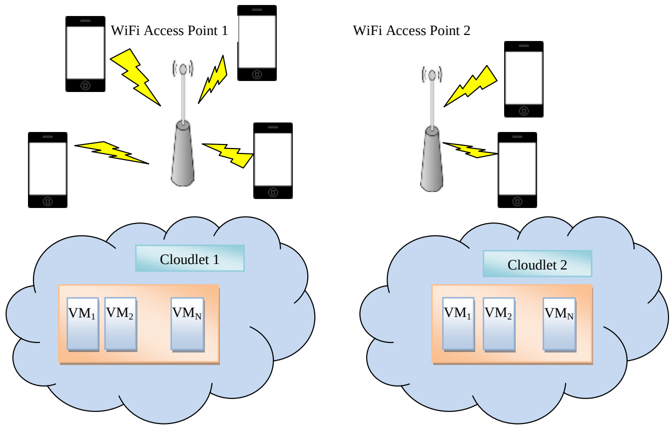

Figure. 5. Architecture of Cloudlets for distributed application processing.

Cloudlets

Cloudlets based approach advocates to organize datacenters in the form of cells like cellular networks. The Cloudlets can be accessed via the Wi-Fi networks. Fig. 5 shows the basic architecture of cloudlets based approach for the distributed application processing systems. The biggest advantage of cloudlets is that it reduces the long term WAN latencies. Satyanarayanan et al. [27] proposed the cloudlets concept. The proposed framework utilizes virtual machine migration technique to offload the execution of local application onto the cloud servers. The entire local application is encapsulated inside a virtual machine and the execution is carried out on the cloudlet servers. Jaraweh et al. [28] emphasized to organize datacenters only near to the mobile device users. However, during mobility author's suggested to use the public cloud services. This study advocated to utilize centralized cloud service for tracking the user's mobility and continuation the service outside the cloudlets region. Yu et al. [12] advocated to use resource sharing through cooperation among the geo-distributed cloud service providers. The authors further argued that such an approach can avoid the centralized cloud service access. Although cloudlets based approach seems to be better than the centralized approach, but it requires certain challenges to be addressed. First of all, the range Wi-Fi networks is limited. Second, it requires to resolve the issues of inter-cloudlet mobility and handoff management. Third, the Cloudlets have limited resources, so, during the peak workloads these systems may not be able to satisfy the users demand effectively i.e. these systems may suffer from the availability issues. Fourth, the inter-cloudlet migration may be hindered by the pricing, security, and policy heterogeneity issues.

Ad-hoc Mobile Cloud

Ad-hoc mobile clouds can be constituted by synthesizing the computing resources from the nearby devices present in the proximity. The objective of such distributed application processing system is to exploit the idle computing power of nearby mobile device such as tablets and smartphones present in a crowded area. Fig. 6 shows the basic architecture of ad-hoc mobile clouds. The remote servers in ad-hoc mobile cloud may be stationary or moving in synchrony with the mobile device. The guidelines for synthesizing the resources from the nearby idle devices have been proposed by a number of researchers. The concept proposed by Canepa et al. [29] argues that the nearby devices can be exploited without internet usage. The study has suggested to utilize the abundant computing power of nearby devices in the remote areas. Liu et al. [13] proposed another cooperation based architecture for MCC. This study focused on resource discovery in the ad-hoc clouds. Panneerselvam et al. [30] proposed a multi-tier architecture for synthesizing the resources from the nearby devices. The proposed architecture advocates to use different level compute clusters for the different application requirements. Like cloudlets, the ad-hoc clouds can also be accessed via Wi-Fi. However, such systems suffer from additional overheads because the service creation, assurance and security are at generosity of the constituting devices.

The comparisons of the different distributed application provisioning systems research are summarized in Table I. It compares the key features of different architectures in aspect of datacenter organization, objectives, scalability. Scalability refers to the process of acceding to the incoming requests optimally. Furthermore, Table I also highlights the pros and cons of each of the architecture.

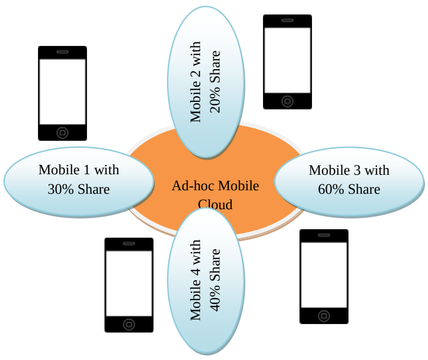

Figure. 6. Ad-hoc Mobile Cloud formed by collaboration of nearby mobile computing devices.

ENABLING TECHNIQUES FOR COMPUTATION OFFLOADING

Computation offloading techniques refers to the process of dispatching the execution of mobile application to the remote cloud servers and getting the results back. Existing computation offloading technique research can be broadly classified in three broad categories: 1) virtual machine migration based, 2) mobile-agent based, and 3) client-server protocols based.

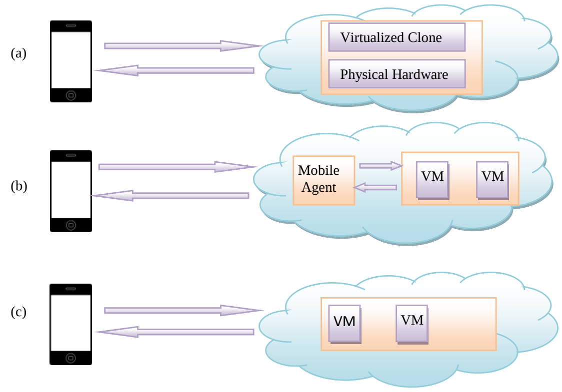

Figure. 7. Computation offloading techniques: a) VM migration based, b) mobile-agent based, c) client-server protocols based.

Virtual Machine (VM) Migration based computation offloading

VM migration based approaches target to avoid the device heterogeneity issues. In this technique, the mobile application running state is packed inside a virtual machine and this virtual machine then migrates to cloud servers for execution. Finally, the application execution results are updated to the mobile device [21]. Fig. 7 (a) shows the steps followed in the virtual machine migration mechanism. Cuervo et al. [31] proposed a .NET reflection based technique for computation offloading. This work requires the application to be re-designed and developers need to mark each component of the application as remote or local executable. The proposed framework formulates the application partitioning process as the dynamic 0/1integer linear optimization problem. The application partitioning process considers the network and workload characteristics to partition the application. CloneCloud framework discussed in [32], is also based on virtual machine migration mechanism. This work tried to avoid the application re-designing overheads. CloneCloud runtime system automatically migrates the runtime threads from the mobile device to the device clones present in the datacenter. However, the CloneCloud system also formulates the application partitioning process as the 0/1 integer linear optimization problem. ThinkAir [33] framework also advocates virtual machine migration mechanism to offload the computation. However, different from CloneCloud the ThinkAir framework advocates for keeping more than one device clone in the cloud for parallel processing of the application. Another architecture based on virtual machine migration is CloudAP [34]. The proposed architecture advocates for migrating the whole system virtual machine. Although the author's claim it to be beneficial, but such architecture can be feasible only if the cloud datacenters are near to the mobile device to avoid the WAN latencies. Although, the virtual machine migration mechanism overcomes the device heterogeneity issues, but it requires higher amount of data to be transferred between mobile device and device clones present in the cloud system. The intermittent connectivity significantly affects the virtual machine migration based computation offloading.

TABLE I
CHARACTERISTICS OF VARIOUS DISTRIBUTED APPLICATION PROCESSING SYSTEMS IN MCC

Researcher	Year	Datacenter Organization	Objective	Scalability	Advantages	Drawbacks
Satyanarayan[27]	2009	Cloudlets	Energy Saving	Medium	Reduces the long WAN latencies	Mobile device remains as a thin client as the entire application execution is carried on cloudlets. Mobility and handoff management issues.
Canepa [29]	2010	Ad-hoc Mobile cloud	Time Saving	Low	Reduces internet data costs and WAN latencies	Security and service assurance are not guaranteed.
Cuervo [31]	2010	Centralized	Energy Saving	High	Offloads only methods by the virtual machine migration mechanism, so lesser data transmission.	Application re-designing overheads for annotations of cloud executable components.
Chun [32]	2011	Centralized	Energy Saving	High	Does not requires changes in the existing application	Requires exclusive virtual machine as the device clone in the cloud
Kosta [33]	2011	Centralized	Energy Saving	High	Parallel processing of the application results in more execution speedup	Requires multiple computational resources on the cloud and alternately may prove to be costlier as the cloud services are paid services.
Hung [38]	2011	Centralized	Time Saving	High	Supports disconnected operation so works in intermittent connectivity for synchronizing the mobile device and cloud resources	Intrinsic security issues of mobile agents may prove to be costlier.
Kovachev [41]	2012	Centralized	Energy Saving	High	Considers the dynamic environmental conditions application partitioning	Vendor-locking issues as it requires specialized communication library
Ferber [14]	2012	Centralized	Energy/Time Saving	High	Adaptively utilizes the cloud resources	Requires 10% extra virtual machine to be running all the time and may be affected heavily by the temporary workload spikes.
Zhang [34]	2013	Cloudlet	Time saving	Medium	Reduces the service down time by placing the clones near to the mobile device	Requires whole system virtual machine migration which alternately raises the security issues.
Angin [37]	2013	Centralized	Energy/Time Saving	High	Supports disconnected operations for method level migration, so lesser data transmission.	Intrinsic security issues of mobile agents may prove to be costlier, also requires applications to be redesigned.
Liu [13]	2014	Ad-hoc Mobile Cloud	Energy Saving	Low	Helps in discovering nearing ad-hoc cloud resources	Heterogeneity in cloud platforms may affect the service during the mobility in ad-hoc clouds.
Shiraz [39]	2015	Centralized	Energy Saving	High	Requires only input and output data to be transmitted, so lighter weight	Requires to address the issues of simultaneous access for the same virtual machine.
Panneerselvam [30]	2016	Ad-hoc Mobile Cloud	Energy saving	Low	Divides the ad-hoc clouds in multiple clusters and allocates these clusters according to the application requirements.	Requires to address the issues of simultaneous access for reusability.
Zhang [26]	2017	Centralized	Energy saving	High	Considers the dynamic environmental conditions at fine granular level.	Does not provides any solution for cloud resource management for accessing the service.

Mobile-Agent based computation offloading

Mobile-agent based computation scheme offloading offers disconnected services to the MCC users and so, helps in overcoming the intermittent connectivity issues [35]. In this technique, the mobile application itself manages the distribution and execution of mobile user's task on the cloud resources [36]. Fig. 7 (b) shows the mechanism of mobile-agent based computation offloading. Angin et al. [37] exploited the concept of agent based computation offloading to reduce energy consumption on the mobile device. The proposed framework advocates for migrating the application components at runtime to cloud servers. Such offloading mechanism avoids the mobile system to remain intact with the cloud virtual machines. Similarly, Hung et al. [38] utilized the mobile agent's concept to synchronize the states of the mobile device and cloud virtual machine. Although mobile agent supports disconnected operations and transfer lesser data in comparison to virtual machine migration techniques, but mobile agent based systems impose serious security threats and so, its further use has been discontinued by the industry. Furthermore, mobile-agent based computation offloading requires developer support for handling the code migration and execution of respective code on cloud resources.

Client-server protocols based computation offloading

Client-server based computation offloading relieves the programmers from bothering about the communication details between clients and servers and does not require any code migration. These protocols just rely on transferring the input and output data between clients and servers. The client-server protocols based computation offloading systems employ either web-based protocols like SOAP [39], REST [7, 40] or traditional client-server protocols RMI or CORBA. Fig. 7 (c) shows the steps followed in the migration mechanism for client-server based computation offloading. Web-based protocols provide platform independent services, but require higher amount of data to be transferred between client and servers [14] as compared to traditional client-server systems. Kovachev et al. [41] proposed a middleware library for client-server communication. The middleware relieves the programmers from designing the communication libraries. The proposed concept formulates the application partitioning as the 0/1 linear optimization problem. The authors showed that upto 95% mobile device energy can be saved for N-queens and face recognition applications. Similarly, Shiraz et al. [39] exploited the SOAP protocol for the migration of execution from the mobile device to the cloud servers. Although, these computation offloading systems are lighter-weight, but require developer-support and changes in the existing applications. Also such offloading systems may suffer from vendor-locking as different cloud service vendors may offer offloading services through different client-server protocols.

The characteristics of different computation offloading systems are summarized in Table II. It highlights the key features of different computation offloading systems in aspect of offloading unit, migration mechanism, application profiling tool, application redesigning, and the sample contender application for MCC. Offloading unit denotes the segment of the mobile application whose execution can be delegated to the remote resources. Migration mechanism is the process of delegating the code execution to cloud resources. Application profiling tool are the tools which can be used for measuring the device hardware characteristics.

PEN CHALLENGES

This section presents the challenges which need further attention for the success of mobile cloud computing.

- ***Standard architecture**:* As mobile cloud computing is still in its primary stage and no standard reference model has been laid down. So, every individual researcher defines the underlying protocols and offloading strategy in different ways. So, there is an urgent need for laying down the standards for developing the applications according to mobile cloud computing.
- ***Cost Model:*** **Different** cloud vendors incur different service costs from the users. However, these costs are available for fixed time quanta's viz. Google cloud service [11] are available for 10 minute time quanta's whereas Amazon [10] cloud service are available for fixed per hour time quanta. However, MCC applications may utilize cloud services for just few seconds. So, this issue must be resolved.
- ***User Preferences:*** Mobile cloud services vendors must consider the user's preference for the service provisioning e.g. some time users may want prompt service and for other time users may want to save prices. So, cloud service providers must offer flexible service to the users.

TABLE II

CHARACTERISTICS OF VARIOUS COMPUTATION OFFLOADING MECHANISMS IN MCC

Researcher	Offloading Unit	Migration Mechanism	Application Profiling Tool	Application Redesign (Yes/No)	Example Application
Satyanarayan[27]	Application	Virtual machine	Not specified	No	Language Translation, Face recognition
Canepa [29]	Application	Client-server	Not specified	No	OCR reader
Cuervo [31]	Method	Client-server	Hardware Power meter	Yes	Chess Game, Face recognition app
Chun [32]	Thread	Virtual machine	Not specified	No	Virus Scanning, Image search
Kosta [33]	Method	Virtual machine	Powertutor	No	N-Queens, Virus Scanning
Hung [38]	Application	Mobile Agent	Not specified	No	Sudoku, google translate, Youtube
Kovachev [41]	Method	Client-server	Powertutor	Yes	N-Queens problem, Face detect application
Ferber [14]	Method	Client-server	Not specified	Yes	Sunflow
Zhang [34]	All Data	Virtual machine	Not specified	No	Abiword, Openoffice, Firefox
Angin [37]	Component	Mobile Agent	Powertutor	Yes	Sudoku, N-Queens problem
Liu [13]	Method	Virtual machine	Not specified	No	Cloud gaminig
Shiraz [39]	Component	Client-server	Not specified	Yes	Bubble Sort, matrix chain multiplication
Panneerselvam [30]	Application	Client-server	Not specified	Yes	Synthetic application workload
Zhang [26]	Application	Not specified	Not specified	Yes	Video application

- ***Service quality assurance**:* The service quality assurance must be provided to users in all circumstances. For instance, in case of peak workloads, the service provider may not be able to procure enough resources. So, there must be some regulations for cloud services.
- ***Profiling tools:*** There is an acute of standard tools for verifying whether an application or its parts need to be offloaded and whether it resulted in any type of benefit proclaimed by the MCC service providers. Some tools are like powertutor [42], powermeter [43], and powerUsage [44] are available, but these tools are not standardized by any authority.
- ***Security:*** **Security** is at the stake in all type of remote communication. It can cause serious problems to the individual mobile device users and cloud service providers in terms of

economic loss and socio life distortion. In addition, complex security measures would impose extra overheads on resource constraint mobile devices. So, in MCC appropriate security measures needs to be undertaken keeping in mind the resource scarceness of mobile devices.

- ***Intermittent Connectivity:*** As MCC services are accessed through the unpredictable and uncertain wireless medium, so, the service disruption may cause un-necessary reconnection overheads. These overheads would consume more resources in terms of time and energy. The virtual machine migration mechanism requires comparatively higher data transmission between clients and servers, so it suffers more significantly as compared to client-server protocol based systems. Although, mobile agents provides disconnected services, but mobile agents usage have been discontinued by the industry. So, there is a need to deal with the intermittent connectivity problem.

- ***Mobility and handoff management****:* The key characteristics of smart mobile devices are portability and availability in all circumstances, but the mobility may require virtual machine migration. These virtual machines may migrate through heterogeneous cloud platforms. So, there is a need to handle the migrations under the heterogeneous distributed backbone platforms and common consensus between different service providers. **Conclusion**

Mobile cloud computing has emerged as a software level solution to augment the computing power of resource scarce mobile devices. It works by uplifting the compute intensive parts of application to the powerful cloud servers. It can not only make the puniest mobile device as a computing giant, but can also realize the dream of computing on the fly. The seamless distribution of application on cloud resources can pave way for the wide applicability of the mobile devices in various fields like real time speech recognition, face recognition etc. This paper presented the underlying technological research for MCC under the two fundamental perspectives: provisioning of distributed application processing systems for MCC and computation offloading. It has also highlighted the associated characteristics for each of the perspectives. Although, research in the mobile cloud computing has made a lot of progress, but several open challenges are still hampering its future growth. The open issues like cost model, service quality assurance, security and standardized architecture needs to be dealt with prime considerations for the further success of mobile cloud computing.

REFERENCES

[1] R. Kemp, N. Palmer, T. Keilmann, F. Seinstra, S. Drost, J. Maassen, and H. Bal, "eyeDentify: Multimedia Cyber Foraging from a Smartphone," Proceedings of of 11th IEEE International Symposium on Multimedia, DOI 10.1109/ISM.2009.21 pp. 392-399.

[2] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile Cloud Computing: A Survey," Future Generation Computing Systems, 2012, DOI: 10.1016/j.future.2012.05.023, pp. 85-106.

[3] P. Bahl, R. Y. Han, L. E. Li and M. Satyanarayanan, "Advancing the State of Mobile Cloud Computing," MCS'12, June 25, 2012, Low Wood Bay, Lake District, UK.

[4] Wikipedia, "Moore's Law – Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Moore%27s_law (accessed: 23.10.2016)

[5] F. Liu, P. Shu, H. Jin, L. Ding, and J. Yu, "Gearing Resource-Poor Mobile Devices with Powerful Clouds: Architectures, Challenges, and Applications" IEEE Wireless Communications, 2013.

[6] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, "A context Sensitive Offloading Scheme for Mobile Cloud Computing Service," Proceedings of 8th IEEE Conference on Cloud Computing, DOI 10.1109/CLOUD.2015.119, pp. 869-876, 2015.

[7] J. Kumar, A. Malik, S. K. Dhurandher and P. Nicopolitidis, "Demand-Based Computation Offloading Framework for Mobile Devices," IEEE Systems Journal, DOI: 10.1109/JSYST.2017.2706178, 2017.

[8] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (Draft),"http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.

[9] M. Armburst, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", EECS Department, University of California, Berkeley, tech. Rep. UCB/EECS-2009-28, 2009.

[10] Amazon ec2 http://aws.amazon.com/ec2/.

[11] google app engine for PaaS https://cloud.google.com/appengine/

[12] R. Yu, J. Ding, S. Maharjan, S. Gjessing, Y. Zhang, and D.H.K. Sung, "Decentralized and Optimal Resource Cooperation in Geo-distributed Mobile Cloud Computing," IEEE Transactions on Emerging Topics in Computing, DOI 10.1109/TETC.2015.2479093, 2015.

[13] W. Liu, T. Nishio, R. Shinkuma, T. Takahashi, "Adaptive Resource Discovery in Mobile Cloud Computing," Computer Communication, http://dx.doi.org/10.1016/j.comcom.2014.02.006, 2014.

[14] M. Ferber, T. Rauber, M. H. C. Torres and T. Holvoet, "Resource Allocation for Cloud-Assisted Mobile Applications," Proceedings of 5th IEEE Conference on Cloud Computing, 2012, DOI 10.1109/CLOUD.2012.75, pp. 400-407.

[15] K. Kumar, J. Liu, Y.H. Lu, and B. Bhargava, "A Survey of Computation Offloading for Mobile Systems," Mobile Network Applications, DOI 10.1007/s11036-012-0368-0, 2012.

[16] J. Kumar and A. Malik, "MAYA: An Approach for Energy and Cost Optimization for Mobile Cloud Computing Environments," Big Data Analytics, Advances in Intelligent Systems and Computing 654, pp. 575-581, 2018.

[17] N. Kaushik, J. Kumar, "A Computation Offloading Framework to Optimize Energy Utilization in Mobile Cloud Computing Environment," International Journal of Computer Applications & Information Technology, Vol. 5, pp. 61-69, 2014.

[18] M. Altamimi, R. Palit, K. Naik, and A. Nayak, "Energy-as-a Service (EaaS): On the Efficacy of Multimedia Cloud Computing to Save Samrtphone Energy," IEEE Fifth International Conference on Cloud Computing, DOI 10.1109/CLOUD.2012.72, pp. 764-771, 2012.

[19] A. P. Miettinen and J. K. Nurminen, "Energy Efficiency of Mobile Clients in Cloud Computing," in proceedingsof 2nd USENIX Conference, Hot Topics Cloud Computing, 2010.

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

[21] B.G. Chun and P. Maniatis, "Augmented Samrtphone Applications through Clone Cloud Execution", Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS), 2009.

[22] Y. Neuvo, "Cellular phones as emdded systems," Digest of Technical Papers, IEEE ISSCC, Vol. 47, No. 1, pp. 32-37, 2004.

[23] D. Ben, B. Ma, L. Liu, Z. Xia, W. Zhang, and F. Liu, "Unusual burns with combined injuries caused by mobile phone explosion: watch out for the "mini-bomb!"," J. Burn Care and research, Vol. 30, No. 6, p. 1048, 2009.

[24] Eucalyptus Cloud. http://www.eucalyptus.com/.

[25] OpenStack Cloud Software. Open Source Software for building private and public clouds. http://www.openstack.org.

[26] L. Zhang, D. Fu, J. Liu, E. C. H. Ngai, and W. Zhu, "On Energy-Efficient Offloading in Mobile Cloud for Real-Time Video Applications," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 27, No. 1, pp. 170-181, 2017.

[27] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. "The Case for VM-based Cloudlets in Mobile Computing," IEEE Pervasive Computing, 2009, pp.14-23.

[28] Y. Jararweh, L.A. Tawalbeh, F. Ababneh and F. Dosari, "Resource Efficient Mobile Computing using Cloudlet Infrastructure", Proceedings of the 9th International Conference on Mobile Ad-hoc and Sensor Networks 2013, DOI 10.1109/MSN 2013.75, pp. 373-377.

[29] G. H. Canepa, and D. 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.

[30] J. Panneerselvam, J. Hardy, L. Liu, B. Yuan, and A. N. Antonopoulos, "Mobilouds: An Energy Efficient MCC Collaborative Framework with Extended Mobile Participation for Next Generation Networks," IEEAccess, DOI 10.1109/ACCESS.2016.2602321, Vol. 4, pp. 9129-9144, 2016.

[31] E. Cuervo, A. Balasubramanian, D.Cho, A. Wolman, S.Saroiu, R. Chandra, and P.Bahl, "MAUI: making smartphones last longer with code offload," In Proceedings of 8th International Conference on Mobile Systems, Applications, and Services, MobiSys'10, ACM, New York, NY, USA,2010 pp.49-62.

[32] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. "CloneCloud: Elastic execution between mobile devices and Cloud," In Proceedings of the Sixth Conference Computer Systems, EuroSys, 2011, pp. 301-314.

[33] S. Kosta, A. Aucinas, P. Hui, R. Mortier and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in cloud for mobile code offloading," INFOCOM, 2012 Proceedings IEEE 10.1109/INFCOM.2012.6195845 pp. 945-953.

[34] Y. Zhang, R. Yang, T. Wo, C. Hu, J. Kang, and L. Cui, "CloudAP: Improving the QoS of Mobile Applications with Efficient VM Migration," In proceedings of the IEEE Conference on High Performance Computing and Communications, 2013, DOI: 10.1109/HPCC.and.EUC.2013.195, pp. 1374-1381.

[35] D. Chess, C. Harrison, and A. Kershenbaum, "Mobile agents: Are they a good idea?" in Mobile Object Systems Towards the Programmable Internet, LNCS, J. Vitek and C. Tschudin, Eds. Springer-Verlag, 1997, vol. 1222.

[36] M. Kristensen, "Scavenger: Transparent Development of efficient cyber foraging applications," In Proceedings of the IEEE International Conference on Pervasive Computing and Communication, (PerCom), DOI: 10.1109/PERCOM.2010.5466972.

[37] P. Angin and B. Bhargava, "An Agent-based Optimization Framework for Mobile-Cloud Computing," Journal of Wireless Mobile networks, Ubiquitous Computing and dependable Applications (JoWUA), Vol. -4,Number 2, pp. 1-17, 2013.

[38] S.H. Hung, C.S. Shih, J.P Shieh, C.P Lee, and Y. H. Huang, "Executing mobile applications on the cloud: Framework and issues," Computers and Mathematics with Applications 63(2012) DOI: 10/1016/j.camwa.2011.10.044, pp. 573-587.

[39] M. Shiraz, A. Gani, A. Shamim, S. Khan, and R.W. Ahmad, "Energy Efficient Computational Offloading Framework for Mobile Cloud Computing," J Grid Computing, DOI 10.1007/s10723-014-9323-6, 2015.

[40] J. H. Christensen, “Using RESTful Web-Services and Cloud Computing to Create Next Generation Mobile Applications,” Proceedings of the 24th Conference on Object-Oriented Programming Systems Language and Applications (OOPSLA’09) ACM 2009, pp. 627-634.

[41] D. Kovachev, T. Yu and R. Klamma, “Adaptive Computation Offloading from Mobile Devices into the Cloud,” In Proceedings of 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, DOI: 10.1109/ISPA.2012.115, pp. 784-791, 2012.

[42] M. Gordon, L. Zhang, B. Tiwana, R. Dick, Z.M. Mao, and L. Yang, “PowerTutor: A power monitor for android-based mobile platforms,” http://ziyang.eecs.umich.edu/projects//powertutor/, 2013.

[43] Monsoon Solutions Inc. Monsoon Power Monitor. http://www.msoon.com/.

[44] F. Ding, F. Xia, W. Zhang, X. Zhao, and C. Ma, "Monitoring energy consumption of smartphones. In proceedings of 2011 International conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, (iThings/CPSCom), pp. 610-613. IEEE, 2011.