International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 6 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT195635

An Adequate Computation Offloading in Mobile Cloud Computing

CH. Ellaji, C. Pradeepthi, P. JayaSri
Department of Computer Science and Engineerin, SoET, SPMVV, Tirupathi, Andhra Pradesh, India

ABSTRACT

The development and upgrades that mobile gadgets have encountered, they are as yet considered as restricted registering gadgets. Presently, clients become the more requesting and hope to execute computational serious applications on their mobile gadgets. Accordingly, Mobile Cloud Computing (MCC) coordinates versatile figuring and Cloud Computing (CC) so as to broaden capacities of mobile devices utilizing offloading procedures. Computational offloading handles restrictions of Smart Mobile Devices (SMDs, for example, constrained battery lifetime, constrained preparing capabilities, and constrained Storage capacity by offloading the execution and outstanding burden to other rich frameworks with better execution and assets. Here, the current offloading systems, Compuatational offloading strategies, and evaluate them alongside their principle basic issues. In addition, it investigates distinctive significant parameters dependent on which the systems are actualized, for example, offloading technique and level of dividing. At last, it condenses the issues in offloading systems in the MCC space that requires further research.

Keywords : Mobile Cloud Computing(MCC), Cloud Computing(CC), Offloading Computation, partitioning algorithm

I. INTRODUCTION

The primary objective of CC is to enable IT departments to concentrate on their services and projects rather than simply dealing with their server farms and keeping them working [2,18,20]. CC is another idea that intends to give computational resources as services in a fast way, on request, and paying according to use. The CC worldview is introduced in three cloud conveyance models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) as appeared in Fig. 1. According to Gartner [3], CC will have in 2016 a Global Compounded Annual Growth Rate (CAGR) of IaaS: 41%, PaaS: 26.6% and SaaS: 17.4%.

As of late, client inclinations for processing have changed on account of the most recent improvements and upgrades in portable figuring advancements. A few reports and studies have displayed the significance of MCC and its effect on versatile customers and undertakings. For example, and according to an ongoing report by ABI Research, in excess of 240 million business will use cloud services through mobile gadgets by2015and this will drive the income of the MCC to $5.2 billion [11].

Additionally, the utilization of PDAs has expanded quickly in different areas, including endeavor, the executives of information frameworks, gaming, e-learning, stimulation, gaming, and human services. In spite of the fact that the expectations that cell phones will command the future registering gadgets, Mobile gadgets alongside their applications are as yet confined by certain impediments, for example, the battery life, processor potential, and the memory limit of the SMDs [31].

These days, present day mobile gadgets have adequate assets, for example, quick processors, huge memory, and sharp screens. Not with standing, it is as yet insufficient to help with registering escalated errands, for example, characteristic language preparing, picture acknowledgment, and basic leadership. Mobile gadgets give less computational power contrasting with server PCs or normal work areas and calculation escalated undertakings put substantial loads on battery control.

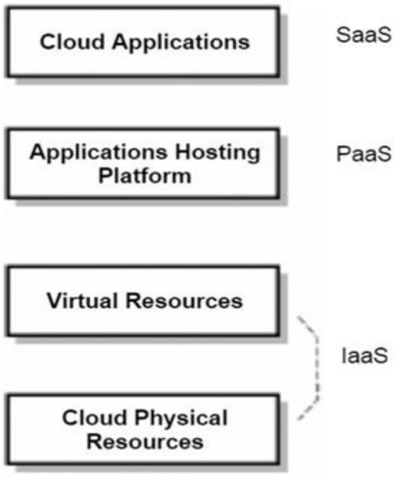


Fig 1: Cloud Computing layers

The following two points highlight our main points in this paper.

1. Classifying current computation offloading frameworks. Analyzing them by identifying their approaches and crucial issues.
2. Presenting the related open issues inv computation offloading for MCC and challenges that require more investigation and elaboration.

This paper is organized as follows: Section 2 explains the essential background concepts and terminology, including CC, and the MCC concepts, Section 3 presents the Compuatation Offloading, Section 4 explains review of offloading frameworks Section 5 highlights the MCOP algorithm . Finally, Section 6 gives a summary and points to future work.

II. CONCEPTS&FRAMEWORK

A. Cloud Computing:

CC is another method for giving computing resources and services. It alludes to an on-demand infrastructure that enables clients to access computing resources whenever from any place [25].CC offers to clients and business three fundamental focus points:

(1) Enormous computing resources available on demand,

(2) Payment for use as needed and on a short-term basis (storage by the day and release them as needed), and

(3) It is Simplified IT management & maintenance capabilities [1].

CC gives clients various applications as administrations by means of the Internet. As instances of open CC we can list Windows Azure and Amazon Web Services (AWS). Windows Azure is an open and flexible cloud stage which gives a few administrations to create, convey and run web applications and administrations in cloud information centers[33].AWS, which Is considered for instance of a registering device, gives clients two models: framework as an assistance and programming as a help. These administrations enable the client to utilize virtualized resources in cloud datacenters[23]. Computational mists actualize an assortment of administration models so as to utilize them in various computing visions[4].

B. Mobile Cloud Computing:

MCC can be viewed as a scaffold that fills the hole between the constrained registering assets of SMDs and preparing necessities of escalated applications on SMDs.

The Mobile Cloud Computing Forum defines MCC as follows[11]: ''Mobile Cloud Computing at its simplest form refers to an infrastructure where both the data storage and the data processing happen outside of the mobile device. Mobile cloud applications move the figuring force and information stockpiling endlessly from mobile gadgets and into the cloud carrying applications and portable processing to advanced mobile phone clients as well as an a lot more extensive scope of versatile supporters". MCC has pulled in the consideration of specialists as a beneficial and helpful business arrangement that limits the advancement and execution expenses of portable applications, enabling versatile clients to get most recent innovation advantageously on an on-demand basis.

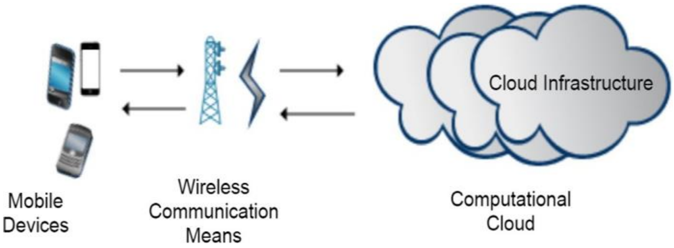


Fig 2 : General view of MCC

Details the general view of MCC which consists of three main parts: the mobile device, wireless communication means, and a cloud infrastructure which contains data centers, which can provide storage services, processing, and security mechanisms for both the cloud environment and mobile devices.

III. PROPOSED WORK

A. Computation Offloading:

Computation offloading is the assignment of sending computation serious application components to a remote server. As of late, various computation offloading systems have been proposed with a few methodologies for applications on mobile gadgets These applications are parceled at various granularity levels and the segments are sent (offloaded) to remote servers for remote execution so as to expand and upgrade the SMD's abilities. In any case, the computation offloading systems are as yet confronting a few difficulties.

In the rest of the piece of this area, our goal was to give a synopsis about the MCC offloading research by examining the accompanying:

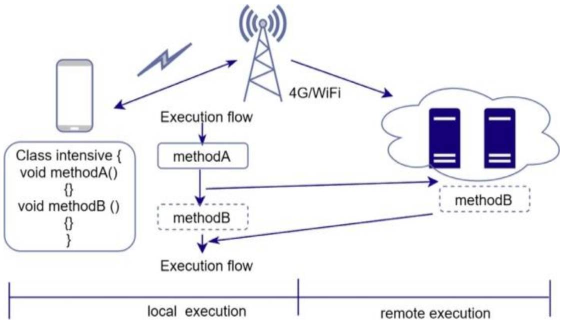


Fig 3 : Offloading Process Overview

1. Usage scenarios for offloading in MCC.
2. Techniques being applied in offloading.
3. A classification of proposed offloading frameworks.

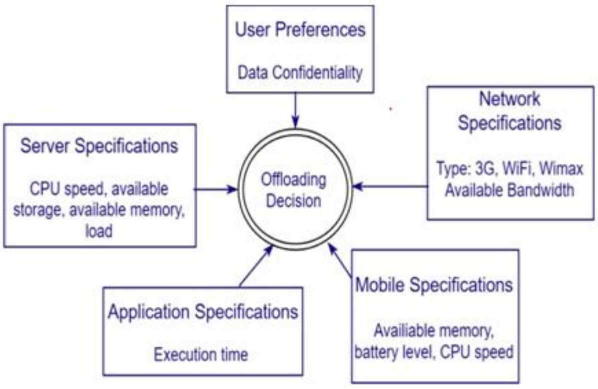


Fig 4 : Aspects affecting the offloading decision

B. Exploration

The latest investigates have concentrated on the virtual machine movement or portable code as offloading technique. Without Hyrax, Virtual Cloud, Cuckoo different structures don't utilize client server architecture. Despite the fact that virtual machine migration and mobile code utilize more seasoned structure Hadoop and Ib is which was intended for dispersed and framework registering, the techniques help VM relocation over regular client server architecture. There as on is Hadoop and Ib is have a greater number of points of interest than RPC. In spite of the fact that client server communication has very much upheld APIs, it isn't so strong if there should be an occurrence of the circulated versatile system. It requires pre-introduced benefits on gadgets and separated tasks are not upheld. Thinking about the heterogeneity, on-going continuous connection and adhoc nature of mobile computing scientists underline the VM migration and mobile code frame works

Cloud let, Clone Cloud, MAUI, MobiCloud use VM movement and it diminishes a great deal of weight from the designer or E-write application code. Be that as it may, full virtualization isn't appropriate for fine-grained code parceling, in spite of the fact that revamping the full or some portion of the application isn't normal. MAUI utilizes a mix of both virtualization and code parceling. Again forward E-mobility of gadgets some of the time the VM migration might be demonstrated heavyweight along these lines Scavenger utilizes the versatile specialist to offload to surrogate gadgets in a powerful domain.

IV. REVIEW OF SOME OFFLOADING FRAMEWORKS

Having inspected diverse existing computation offloading structures alongside their fundamental qualities, a general view about these systems and classifies them dependent on the accompanying characteristics:

Preparation: Any necessary preparations before offloading. Partitioning supported or not.

Decision: Dynamic or static.

Offloading Mechanism: Mechanism used to offload intensive computations.

Granularity Level: Granularity Level (i.e. class, method, thread).

Annotation: Automation of partitioning process (Automatic or manual).

Contribution: Solved problems?

A. Security and privacy in mobile cloud applications

Security of data transmission is a significant worry in cloud based application handling. Security and protection are two crucial ideas that should be kept up during the offloading process. These ideas can be tended to from various angles:(1)Mobile device, (2)cloud data centers , and (3) During data transmission over the system. Other than every one of the innovations, there is an incredible increment in the assortment of refined assaults on mobile gadgets which are the principle focuses for attackers. With respect to security in the cloud data centers, dangers are essentially identified with the transmission of data between the various hubs or nodes over the network. Consequently, high levels of security are normal by both the versatile customers and the cloud suppliers. In the current frameworks [10,12], binary transfer of the application code at runtime is constantly subjected to security threats. In spite of the accessible arrangements, solid measures and a secure environment are required for the three substances of MCC model.

In [39], the authors center around streamlining errands and computations, and they investigate secure offloading of material Linear programming (LP) computations. In this paper, authors construct their work dependent on the disintegration of the LP computation offloading into open LP solvers running on the cloud and private LP parameters possessed by the client. To accomplish an efficient and approve results, the authors center around the major duality hypothesis of LP computation and think of the basic conditions that must satisfied by correct results .

Bugiel et al. present in [40] a design for secure outsourcing of data and arbitrary computations to an untrusted cloud commodity . The architecture proposed in their methodology comprises of two mists (twins): a Trusted cloud and a commodity cloud.

The computations are isolated so that the Trusted cloud is mainly used for critical situations ,whereas solicitations to the offloaded at are prepared in parallel by the fast commodity cloud on encrypted data.

Be that as it may, partitioning tasks and dealing with them by various mists lead to various difficulties.

For example, the deployment and support of this architecture of cloud will require clear modifications in the infrastructure.

The security threat is progressing in a fast way beyond what we can stay aware of it. Security techniques need to upgrade and advance continually to meet new changes and new offered services. Along these lines, it is never again conceivable to define a security framework that would illuminate all the security dangers without a moment's delay.

B. Fault-tolerance and Continuous Connectivity

In MCC, portability is one of the most significant qualities of SMDs. This is on the grounds that opportunity of development and autonomy of communication during the utilization of mobile cloud services , are urgent criteria for clients' fulfillment. Be that as it may, there are a few imperatives that forestall the accomplishment of consistent availability and continuous access to cloud services while moving.

As mobile users move, data trade rates and network data transfer capacity may fluctuate. Additionally, clients may lose their connection while sending or getting data ; there-fore, offloading approaches ought to be given appropriate fault tolerant techniques so as to resend the lost components, limit the response time, and diminish the energy consumption of mobile gadgets. It ought to be noticed that the assurance of a fruitful execution of offloaded applications is exceptionally pivotal for mobileusers.

C. Automatic Mechanism

The accessible computation offloading frameworks still should be mechanized. This will enable the offloading to procedure to be per shaped in a consistent manner while finding the encompassed condition [5,9,14]. The accomplishment of such automation isn't a simple assignment as it needs the execution of a convention committed to finding and finding services relying upon the present setting and its constraints.

D. Partition Offloading and External Data Input

At runtime, it is trying to choose which application components should be offloaded and to find the reasonable server for that. Algorithms answering this problem need resource intensive effort, , which can influence the execution time of the offloaded partitions of the application [13].

Albeit existing application partitioning algorithm permit a versatile execution of the application between the mobile gadgets and the cloud servers, regardless they don't give any arrangement on the most proficient method to use and benefit from the flexible assets in the mists. This is specifically required so as to make the

Mobile cloud computing for computation offloading applications versatile when large number of mobile users should be served and when the application requires input information that are put away in other remote servers. It reiterates the fundamental difficulties to current offloading frameworks and open research issues in MCC. The difficulties demonstrate the issues in the computation offloading frameworks in MCC that still require more elaboration and careful examination, while the open issues indicate uncertain issues in current offloading frameworks.

E. Partitioning Algorithm for offloading Computation

In this segment, we present the min-cost offloading dividing (MCOP) algorithm for WCGs of arbitrary topology. The MCOP algorithm accepts a WCG as input to which an application's operations/calculations are represented to as the nodes and the communication between them as the edges. Every node has two costs: first the cost of playing out the activity locally (e.g., on the mobile device) and second the cost of performing it somewhere else (e.g., in the cloud). The weight of the edges is the communication cost to the offloaded computation. We accept that the communication cost between undertakings in a similar area is irrelevant. The result contains data about the cost and reports which operations ought to be performed locally and which ought to be offloaded.

V. MCOP ALGORITHM

A. Unoffloadable Vertices Merging

An unoffloadable vertex is the one that has extraordinary highlights making it incapable to be relocated outside of the mobile phones and consequently it is found distinctly in the unoffloadable partition. Aside from this, we can pick any assignment to be executed locally as per our inclinations or different reasons. At that point all vertices that won't be relocated to the cloud are converged into one that is chosen as the source vertex. By 'merging', we imply that these hubs are combine into one, whose weight is the total of the loads of every blended hub. Give G a chance to speak to the first diagram after all the unoffloadable vertices are merged.

B. Coarse Partitioning

The objective of this progression is to coarsen G to the coarsest graph G|V|. To coarsen intends to merge two nodes and diminish the node count by one. Hence, the algorithm has |V|−1 stages. In each stage i(for 1 ≤ i ≤| V|−1), the cut value, for example the partitioning cost in a graph G_i =(V_i, E_i) is determined. G_i+1 emerges from G_i by blending "suitable nodes", where G_i = G. The partitioning results are the base cut among every one of the cuts in an individual stage i and the relating bunch records for neighborhood and cloud execution. Moreover, in each stage i of the coarse partitioning despite everything we have five steps:

(a) Start with A={a}, where a is usually an unoffloadable node in G_i.

(b) Iteratively add the vertex to A that is the most tightly connected to A.

(c) Let s, t be the last two vertices (in order) added to A.

(d) The graph cut of the phase i is (V_i\ {t},{t}).

(e) G_i+1 arises from Gi by merging vertices s and t.

C. Algorithmic process

The algorithmic process is illustrated as the Min Cut function ,and in each stage i, it calls the Min Cut Phase work as portrayed. Since some tasks have to be executed locally, we need to merge the min to one node.

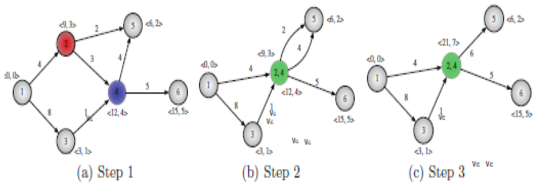
The merging function is utilized to merge two vertices into one new vertex, which is executed On the off chance that nodes s,t∈ V (s ≠ t), at that point they can be merged as follows:

1. Nodes s and t are chosen.

2. Nodes s and t are replaced by a new node $x_{s,t}$. All edges that were previously incident to s or t are now incident to $x_{s,t}$(except the edge between nodes s and t when they are connected).

3. Multiple edges are resolved by adding edge weights. The weights of the node $x_{s,t}$ are resolved by adding the weights of s and t.

For example, we can merge nodes 2 and 4 as shown in Figure



(a) Step 1 (b) Step 2 (c) Step 3

Fig 5 : An example of merging two nodes

Finally the algorithm makes easy to select next vertex to be added to the local set

VI.CONCLUSION

This paper explores the various issues in current offloading frameworks and features challenges that still block these systems in MCC. Besides, the paper shows the various methodologies that are utilized by the structures to accomplish offloading. A portion of these methodologies utilize static offloading while others utilize dynamic offloading. Despite the fact that there exist an assortment of approaches, every one of them focus on a similar target which is the improvement of the mobile phone gadget abilities by sparing vitality, diminishing reaction time, or limiting the execution cost.

Here presents offloading structures are still facing a few difficulties and difficulties. For example, absence of standard structures. This deficiency prompts more difficulties while creating and dealing with a proposed structure. At long last, it is imperative to think of a lightweight worldview or model that will defeat the difficulties and scaled down endeavors while creating, conveying, and dealing with an offloading structure.

We accept that investigating different other options, for example, introducing a middleware based engineering utilizing an advancing offloading calculation, could help better the accessible edge works and give more efficient and increasingly flexible answers for the MCC clients.

VII. REFERENCES

[1]. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al, A view of cloud computing, Commun. ACM 53 (4) (2010) 50–58.

[2]. R. Barga, D. Gannon, D. Reed, The client and the cloud: democratizing research computing, IEEE Internet Comput. 15 (1) (2011) 72.

[3]. B. Butler, Gartner: Cloud Putting Crimp in Traditional Software, Hardware Sales, Networkworld, 2012.

[4]. R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generat. Comput. Syst. 25 (6) (2009) 599–616.

[5]. W.-C. Chuang, B. Sang, S. Yoo, R. Gu, M. Kulkarni, C. Killian, Eventwave: programming model and runtime support for tightly-coupled elastic cloud applications, in: Proceedings of the 4th annual Symposium on Cloud Computing, ACM, 2013, p. 21.

[6]. B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, A. Patti, CloneCloud: elastic execution between mobile device and cloud, in: Proceedings of the Sixth Conference on Computer Systems, ACM, 2011, pp. 301–314.

[7]. B.-G. Chun, P. Maniatis, Augmented smartphone applications through clone cloud execution, HotOS 9 (2009) 8–11.

[8]. E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R. Chandra, P. Bahl, Maui: making smartphones last longer with code offload, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, ACM, 2010, pp. 49–62.

[9]. Y. Cui, X. Ma, H. Wang, I. Stojmenovic, J. Liu, A survey of energy efficient wireless transmission and modeling in mobile cloud computing, Mobile Networks and Applications 18 (1) (2013) 148–155.

[10]. J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.

[11]. H.T. Dinh, C. Lee, D. Niyato, P. Wang, A survey of mobile cloud computing: architecture, applications, and approaches, Wireless Commun. Mobile Comput. 13 (18) (2013) 1587–1611.

[12]. A. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikainen, V.H. Tuulos, Misco: a mapreduce framework for mobile systems, in: Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments, ACM, 2010, p. 32.

[13]. I. Giurgiu, O. Riva, G. Alonso, Dynamic software deployment from clouds to mobile devices, in: Middleware 2012, ACM, 2012, pp. 394–414.

[14]. M.S. Gordon, D.A. Jamshidi, S. Mahlke, Z.M. Mao, X. Chen, Comet: Code offload by migrating execution transparently, in: Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12), 2012, pp. 93–106.

[15]. K. Ha, G. Lewis, S. Simanta, M. Satyanarayanan, Cloud Offload in Hostile Environments, Technical Report, DTIC Document, 2011.

[16]. S. Hakak, S.A. Latif, G. Amin, A review on mobile cloud computing and issues in it, Int. J. Comput. Appl. 75 (11) (2013).

[17]. J. Hauswald, T. Manville, Q. Zheng, R. Dreslinski, C. Chakrabarti, T. Mudge, A hybrid approach to offloading mobile image classification, in: International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE, IEEE, 2014, pp. 8375–8379.

[18]. A. Huth, J. Cebula, The Basics of Cloud Computing, United States Computer, 2011.

[19]. R. Kemp, N. Palmer, T. Kielmann, H. Bal, Cuckoo: a computation offloading framework for smartphones, in: Mobile Computing, Applications, and Services, Springer, 2010, pp. 59–79.

[20]. K. Kumar, Y.-H. Lu, Cloud computing for mobile users: can offloading computation save energy?, Computer 43 (4) (2010) 51–56

[21]. S. Kundu, J. Mukherjee, A.K. Majumdar, B. Majumdar, S.S. Ray, Algorithms and heuristics for efficient medical information display in PDA, Comput. Biol. Med. 37 (9) (2007) 1272–1282.

[22]. G.A. Lewis, S. Echeverr'ıa, S. Simanta, B. Bradshaw, J. Root, Cloudlet-based cyber-foraging for mobile systems in resource constrained edge environments, in: Companion Proceedings of the 36th International Conference on Software Engineering, ACM, 2014, pp. 412–415.

[23]. S. Mathew, Overview of Amazon Web Services, Amazon Whitepapers, 2014.

[24]. J. Matthews, M. Chang, Z. Feng, R. Srinivas, M. Gerla, Powersense: power aware dengue diagnosis on mobile phones, in: Proceedings of the First ACM Workshop on Mobile Systems, Applications, and Services for Healthcare, ACM, 2011, p. 6.

[25]. P. Mell, T. Grance, The Nist Definition of Cloud Computing, 2011.

[26]. A. Mtibaa, A. Fahim, K.A. Harras, M.H. Ammar, Towards resource sharing in mobile device clouds: power balancing across mobile devices, ACM SIGCOMM Comput. Commun. Rev. 43 (4) (2013) 51–56.

[27]. Y. Nimmagadda, K. Kumar, Y.-H. Lu, C.G. Lee, Real-time moving object recognition and tracking using computation offloading, in: International Conference on Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ, IEEE, 2010, pp. 2449– 2455.

[28]. H. Qian, D. Andresen, Jade: reducing energy consumption of android app, Int. J. Network. Distrib. Comput (IJNDC) 3 (3) (2015) 150–158 (Atlantis Press).

[29]. M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for vm-based cloudlets in mobile computing, IEEE Pervasive Comput. 8 (4) (2009) 14–23.

[30]. M. Shiraz, E. Ahmed, A. Gani, Q. Han, Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing, J. Supercomput. 67 (1) (2014) 84–103.

[31]. M. Shiraz, A. Gani, R.H. Khokhar, R. Buyya, A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing, IEEE Commun. Surv. Tutorials 15 (3) (2013) 1294–1313.

[32]. M. Smit, M. Shtern, B. Simmons, M. Litoiu, Partitioning applications for hybrid and federated clouds, in: Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research, IBM Corp., 2012, pp. 27–41.

[33]. M. Tulloch, Introducing Windows Azure for IT Professionals, Microsoft Press, 2013.

[34]. C. Wang, Z. Li, Parametric analysis for adaptive computation offloading, ACM SIGPLAN Notices, vol. 39, ACM, 2004, pp. 119–130.

[35]. S. Wang, S. Dey, Rendering adaptation to address communication and computation constraints in cloud mobile gaming, in: Global Telecommunications Conference (GLOBECOM2010), 2010 IEEE, IEEE, 2010, pp. 1–6.

[36]. F. Xia, F. Ding, J. Li, X. Kong, L.T. Yang, J. Ma, Phone2cloud: exploiting computation offloading for energy saving on smartphones in mobile cloud computing, Inform. Syst. Front. 16 (1) (2014) 95–111.

[37]. K. Yang, S. Ou, H.-H. Chen, On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications, Commun. Mag. IEEE 46 (1) (2008) 56–63.

[38]. B. Zhao, Z. Xu, C. Chi, S. Zhu, G. Cao, Mirroring smartphones for good: a feasibility study, in: Mobile and Ubiquitous Systems: Computing, Networking, and Services, Springer, 2010, pp. 26– 38.

[39]. C. Wang, K. Ren, J. Wang, Secure and practical outsourcing of linear programming in cloud computing, in: INFOCOM, 2011 Proceedings IEEE, IEEE, 2011, pp. 820–828.

[40]. https://www.marketsandmarkets.com/Market-Reports /mobile- applications-228.html

[41]. https://www.businesswire.com/news/home/20110609005403/en/ Worldwide-Smartphone-Market-Expected-Grow-55-2011.

[42]. Conti M, Chong S, Fdida S, Jia W, Karl H, et al. (2011) Research challenges towards the Future Internet. Comput commun 34: 2115-2134.

[43]. Gavalas D, Economou D (2010) Development platforms for mobile applications: status and trends. IEEE Software 28: 77-86.

Cite this article as :

CH. Ellaji, C. Pradeepthi, P. JayaSri, "An Adequate Computation Offloading in Mobile Cloud Computing ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 6, pp. 208-216, November-December 2019. Available at doi : https://doi.org/10.32628/CSEIT195635
Journal URL : http://ijsrcseit.com/CSEIT195635