

Offloading for Application Optimization Using Mobile Cloud Computing

Chinu Singla and Sakshi Kaushal

Abstract Mobile applications are increasingly becoming ubiquitous which provides rich functionalities and services to mobile users. But despite having some technological advancements, smart mobile devices (SMDs) are still minimum potential computing devices enforced by battery life, storage capacity, and network bandwidth which obstructs the possible execution of computational intensive applications. Thus, mobile cloud computing is employed to optimize the computationally intensive applications which use computation offloading techniques for increasing SMD's capabilities. The aim of this paper is to emphasize the specific issues related to mobile cloud computing, offloading, and thus concluding the paper by analyzing the challenges that have not been met perfectly and charts a roadmap towards this direction.

Keywords Mobile cloud computing · Application offloading · Distributed systems · Smart mobile devices

1 Introduction

The latest development in mobile computing has altered user preferences for computing. As users expect to operate computationally intensive applications on their smart mobile devices (smartphone's, personal computers, tablets) but smart mobile devices have certain limitations such as battery life, storage capacity, CPU potential, etc, due to which the user faces so many problems while running large applications. Examples of such applications comprise image processors [1, 2],

Chinu Singla (✉) · Sakshi Kaushal
Computer Science and Engineering, UIET, Panjab University,
Chandigarh, India
e-mail: cheenusingla10@gmail.com

Sakshi Kaushal
e-mail: sakshi@pu.ac.in

video processing [3], and online gaming. Such applications need battery lifetime, more computing power on resource constrained SMDs [4]. To overcome these limitations, the concept of mobile cloud computing has been introduced. Besides, its computational offloading is a good solution to enhance the performance of the mobile application.

Cloud computing (CC) directs to manipulating, configuring and accessing the application via the Internet. It provides online infrastructure, data storage, and applications. Cloud computing provides various services and resources to SMDs such as software as a service (SaaS), platform as a service (PaaS), infrastructure as a service (IaaS), testing as a service (TaaS), etc., through different service provider applications in an on-demand fashion and users can access these resources through centralized cloud server. With the support of cloud computing and the eruption of mobile applications, mobile cloud computing (MCC) is introduced, which is defined as the combination of cloud computing into the mobile environment. MCC provides ubiquitous and computationally intensive mobile applications by attaining the benefits of cloud data centers. It brings new facilities and services to mobile users by taking the benefits of CC and thus reduces the resource limitations of SMDs. For increasing computing potentials of SMDs and to eliminate resource limitations, MCC utilizes application offloading techniques [5, 6]. In application offloading, computationally intensive applications are offloaded by migrating intensive applications to remote server nodes. The terms “surrogate computing” or “cyber foraging” are also used to define computation offloading. Computation offloading methods improves the performance of mobile systems by saving energy and by reducing execution cost. Offloading can be performed either in the static environment (constant network bandwidth, server loads, and connection status) or in the dynamic environment (by changing network bandwidth, server loads, and connection status). For offloading intensive applications partitioning of the applications is done and it may be static or dynamic. In static application partitioning, various components are partitioned either at compile time or at runtime in a static manner, i.e., at development time, whereas in dynamic application partitioning, the applications are separated in a dynamic manner at runtime. Static partitioning is valid if and only if all the performance parameters are predicted in advance but this approach minimizes the overall performance because the servers speed may vary. So currently offloading techniques employ dynamic partitioning techniques to improve the performance of SMDs. But the major challenging issue in dynamic partitioning is the additional consumption on mobile devices during the establishment and management of distributed applications at runtime. This paper examines different frameworks for computational offloading at runtime by discussing the critical aspects, approaches used and related issues to it.

The remaining of the paper is as follows: Sect. 2 explains the concept of mobile cloud computing and computation offloading. Section 3 analyzes the different application partitioning frameworks. Section 4 concludes the paper by discussing the future work.

2 Background

This section demonstrates the basic concepts of MCC and also describes the concept and existing work related to computation offloading in MCC.

2.1 Mobile Cloud Computing

It is the current practical distributed computing model where both data processing and data storage occurs outside of mobile device. It is a computing paradigm which provides various outspread services to smart mobile devices. It shortens the development and execution cost of the mobile devices and enable users to access a variety of cloud services in an on-demand basis. Its main objective is to increase the computing potential of SMDs. Smartphone is a compressed mobile computing device which combines computing power of handheld equipments such as PDAs and capabilities of common cellular mobile phones. MCC provides various augmentation methods such as storage augmentation, screen augmentation, energy augmentation, and application processing augmentation to alleviate resource limitations in SMDs [7]. Three major components of MCC model are SMDs, computational cloud, and Internet wireless technology. Various wireless network protocols such as Wi-Fi, LTE, 3G, etc, can be used by SMDs so that one can easily access the services of computational clouds. Key issues occurs in mobile cloud computing are end user issues, privacy and security issues, application and services level issues, data management, context awareness issues, and operational issues. These issues highlight a certain set of challenges in MCC as depicted in Fig. 1.

MCC uses various cloud services which provide cloud processing services and online storage for extending the processing capabilities of SMDs [6]. The following section explains the concept of computation offloading which is a solution to reduce the limitations of MCC and enhances mobile systems potentiality by transferring heavy computation to rich resourceful datacenters.

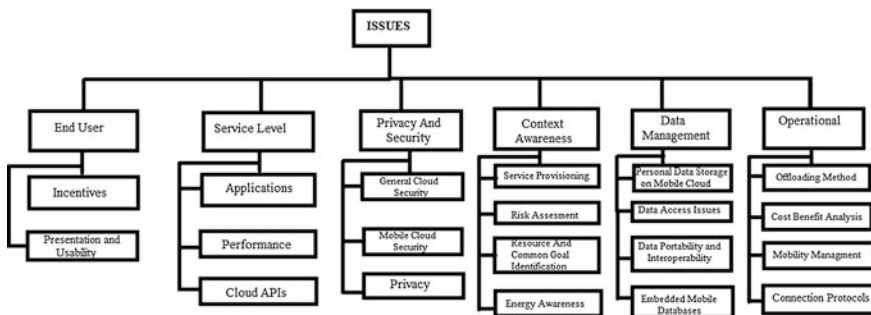


Fig. 1 Issues in mobile cloud computing

2.2 Computation Offloading

MCC utilizes the computing power of more resourceful cloud servers by offloading the computational intensive applications to cloud datacenters [8–10]. Nowadays, many computational offloading-based mechanisms have been introduced for outsourcing large applications partially or fully to remote servers [6, 11–13]. The latest computation offloading frameworks pinpoints the configuration of distributed processing of applications at runtime. Application partitioning is the most important step of offloading process and it can mainly be classified as static and dynamic.

2.2.1 Static and Dynamic Partitioning

When the offloading decision is static, it means partitioning of program is done during its development. Static partitioning involves low overhead and here the intensive application components are partitioned statically either at runtime or at compile time. This method is efficient only if all the factors are accurately known in advance. Prediction algorithms include history-based prediction [14, 15], fuzzy control [16], and probabilistic prediction [17]. It involves onetime partitioning of the application, i.e., during its design time. Here the distribution of workload occurs only once between remote servers and SMDs.

When the offloading decision is dynamic, it adapts to different runtime conditions. In dynamic offloading partitioning is done during execution of the program. This approach involves high overhead as compared to static partitioning approach because here program deals with the runtime conditions (network bandwidth, server speed). Distributed application processing frameworks is entrenched at runtime in an ad hoc manner where elastic applications are partitioned dynamically at runtime [18]. The establishment of distributed application frameworks at runtime is an energy consuming and a resource intensive method [13]. This partitioning method employs two steps, which includes application profiling and solving [11]. The mechanism where computational intensive components are identified is called application profiling, whereas application solving is the method where intensive components are separated for offloading process.

2.2.2 Algorithm for Application Partitioning and Offloading Process

Offloading Algorithm Begin:

- 1: Elastic mobile applications execute on SMDs and utilize the application profiling mechanism.
- 2: Then it evaluates availability and requirement of resources.
- 3: If there are insufficient resources then application solving mechanism is initialized to separate the computationally intensive applications at runtime. Go to step 5.

- 4: Else application executes on local SMDs. Go to step 6.
- 5: After partitioning, application is migrated to cloud server node for remote processing.
- 6: Upon successful execution of remote components of the application, the result is returned to the main application and running on SMDs.
- 7: Exit.

2.2.3 Various Entities Affecting Offloading Process

There are various parameters that affect the process of offloading as discussed below.

- (a) Smartphone efficiency: If a smartphone has powerful processor than that of a cloud server, then taking an offloading decision is a complex process.
- (b) Nature of application: If an application requires inbuilt resources, then offloading method is not beneficial.
- (c) Cloud provider: The cloud provider must have the availability of powerful resources otherwise; there will not be a benefit to offload the application to the cloud.

3 Analysis of Application Partitioning Frameworks

Current offloading frameworks offload computationally intensive components at various granularity levels such as at module, class, object, bundle, thread, and method levels. At the module level, whole module is migrated for offloading process [12, 13]. In the class level partitioning, application is partitioned into classes [9, 10]. At the object level, an object is partitioned for application offloading. At the thread level, a thread is offloaded for remote processing mechanism of the applications [19]. In the same manner, bundle level granularity represents groups of classes that are partitioned [20, 21]. Migration support indicates the support level required for migrating the intensive applications to remote clouds. Various migration patterns are VM-based migration, entire application-based migration, and application partitioning-based application offloading. The main objective for application offloading of the application is to improve performance, reducing memory constraints, saving energy, and updating the application dynamically. MISCO algorithm [22] employs static partitioning which partitions the application into two functions: map and reduce. Mirror server framework [23] employed VM-based migration; the designed VM template is called mirror and the server which is responsible for its management and development is called mirror server. The critical aspects of mirror server are that it is not basically sketched for data processing due to that limited services which can be achieved from it. Elastic clone cloud [24] uses thread-based granularity at application level. A major

disadvantage of this framework is that it requires a heavy traffic synchronization mechanism. In this architecture, VM-based migration has been done because it requires more resources and an additional support from operating system. In [18] a middleware framework has been proposed for the establishment of distributed applications dynamically. Its main critical aspects are more resource consumption due to dynamic partitioning of application.

Mobile assistance using infrastructure (MAUI) [25] is a dynamic partitioning-based framework which concentrates on saving energy techniques for SMDs. It uses application proxy method at application level. The main aspects of this framework are that it requires more efforts for the development of individual method of the application. This architecture is based on method level granularity so difficulty in maintaining consistency. Elastic application model [6] is a middleware framework which implements distribution processing elastic applications at application layer. Cuervo et al. [25] and Zhang et al. [6] employed method level granularity and requires more overhead because such frameworks involves application profiling and partitioning mechanisms at runtime [26]. Investigate VM deployment and management in simulation environment can be investigated using cloudsim. Such models need feasible profiling application method which is resource starvation and a time-consuming method [27]. Considered active service migration (ASM) approach which reduces energy consumption cost (ECC) and turnaround time by deploying coarse level granularity which present the lightweight nature of proposed framework. The limitations of such frameworks are that it involves additional overhead of application offloading and more complications occur in the deployment and management of distributed applications at runtime. Shiraz et al. [28] included SaaS with IaaS for eliminating ECC during component migration at runtime such as binary code migration cost and active data state migration cost. The critical aspects of this proposed framework is that it lacks in maintaining consistency between local SMDs and cloud server node.

Since there have not been sufficient work done in order to resolve all issues; such as availability of resources, homogeneous and consistent distributed platform, security and privacy issues so there is a scope to develop a lightweight model which overcomes these issues and optimizes and enhances the overall performance of applications in MCC.

4 Conclusion and Future Scope

This paper explains the procedure of mobile cloud computing, computation offloading and analyzes current distributed application processing frameworks. In this paper, we have explored the challenges and issues to optimal and lightweight distributed applications frameworks for MCC. Further, a lightweight optimized framework is needed to be devised which will enhance the performance of application and reduces developmental efforts by overcoming the challenges related to distributed processing of applications in mobile cloud computing.

Our future work will therefore focus on proposing lightweight methods to carry out the offloaded computation tasks and to design an optimal algorithm to enhance the performance of mobile systems by investigating the system performance in a real environment.

References

1. M. Kristensen and N. Bouvin, "Developing cyber foraging applications for portable devices," in *Portable Information Devices, 2008 and the 2008 7th IEEE Conference on Polymers and Adhesives in Microelectronics and Photonics. PORTABLE-POLYTRONIC 2008. 2nd IEEE International Interdisciplinary Conference on*. IEEE, 2008, pp. 1–6.
2. J. Porras, O. Riva, and M. Kristensen, "Dynamic resource management and cyber foraging," *Middleware for Network Eccentric and Mobile Applications*, vol. 1, p. 349, 2009.
3. B. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proc. 8th Workshop on Hot Topics in Operating Systems (HotOS), Monte Verita, Switzerland, 2009*.
4. M. Sharifi, S. Kafaie, and O. Kashefi, "A survey and taxonomy of cyber foraging of mobile devices," *IEEE Commun. Surveys Tuts.*, 2011.
5. G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proc. 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*. ACM, 2010, p. 6.
6. X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 270–284, 2011.
7. S. Abolfazli, Z. Sanaei, and A. Gani, "Mobile cloud computing: A review on smartphone augmentation approaches," in *Proc. 1st International Conference on Computing, Information Systems and Communications, 2012*.
8. Sanaei Z, Abolfazli S, Gani A, Buyya R. Heterogeneity in mobile cloud computing: Taxonomy and open challenges. *IEEE Communications Surveys and Tutorials* 2014; 16 (1):369–92.
9. Shiraz M, Ahmed E, Gani A, Han Q. Investigation on Runtime Partitioning of Elastic Mobile Applications for Mobile Cloud Computing. *Journal of Supercomputing* 2014; 67(1):84–103.
10. Abolfazli S, Sanaei Z, Ahmed E, Gani A, Buyya R. Cloud-based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Issues. *IEEE Communications Surveys and Tutorials* 2014; 16(1):337–68.
11. Cuervo E, Balasubramanian A, Cho DK, Wolman A, Saroiu S, Chandra R, Bahlx P. "MAUI: Making Smartphones Last Longer with Code Offload" *MobiSys'10*. June 15–18, 2010.
12. Hung, S. H., Shih, C. S., Shieh, J. P., Lee, C. P., & Huang, Y. H. (2012). Executing mobile applications on the cloud: Framework and issues. *Computers & Mathematics with Applications*, 63(2), 573–587.
13. Shiraz, M., Gani, A., Khokhar, R. H., & Buyya, R. (2013). A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing. *Communications Surveys & Tutorials, IEEE*, 15(3), 1294–1313.
14. Gurun S, Krintz C, Wolski R (2004) NWSLite: a lightweight prediction utility for mobile devices. In: International conference on mobile systems, applications, and services, pp. 2–11.
15. Huerta-Canepa G, Lee D (2008) An adaptable application offloading scheme based on application behavior. In: International conference on advanced information networking and applications—workshops, pp. 387–392.

16. Gu X, Nahrstedt K, Messer A, Greenberg I, Milojevic D (2003) Adaptive offloading inference for delivering applications in pervasive computing environments. In: IEEE international conference on pervasive computing and communications, pp. 107–114.
17. Rong P, Pedram M (2003) Extending the lifetime of a network of battery-powered mobile devices by remote processing: a markovian decision-based approach. In: Conference on design automation, pp. 906–911.
18. A. Messer, I. Greenberg, P. Bernadat, D. Milojevic, D. Chen, T. Giul and X. Gu, “Towards a distributed platform for resource-constrained devices,” in *Distributed Computing Systems, 2002. Proceedings of the 22nd International Conference on*. IEEE, 2002, pp. 43–51.
19. Goyal S, Carter J. A Light weight Secure Cyber Foraging Infrastructure for Resource—Constrained Devices, WMCSA 2004 6th IEEE Workshop. IEEE Publisher; 2–3 December 2004.
20. Giurciu I, Riva O, Juric D, Krivulev I, Alonso G. Calling the cloud: enabling mobile phones as interfaces to cloud applications. In: Proceedings of the ACM/IFIP/ USENIX 10th international conference on middleware (Middleware’09), Urbana, IL, USA. Springer-Verlag; 2009. pp. 83–102.
21. Giurciu I, Riva O, Alonso G. Dynamic software deployment from clouds to mobile devices. In: Proceedings of the 13th international middleware conference (Middleware’12), Montreal, QC, Canada. Springer-Verlag; 2012. pp. 394–414.
22. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikainen, and V. Tuulos, “Misco: a map reduce framework for mobile systems,” in *Proc. 3rd International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 2010, p. 32.
23. Zhao, Z. Xu, C. Chi, S. Zhu, and G. Cao, “Mirroring smartphones for good: A feasibility study,” *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pp. 26–38, 2012.
24. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clone cloud: elastic execution between mobile device and cloud,” in *Proc. sixth conference on Computer systems*, 2011, pp. 301–314.
25. E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “Maui: making smartphones last longer with code offload,” in *Proc. 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 49–62.
26. Shiraz M, Abolfazli S, Sanaei Z, Gani A (2012) A study on virtual machine deployment for application outsourcing in mobile cloud computing. *J Supercomput* 63(3):946–964.
27. Shiraz, M., Gani, A.: A lightweight active service migration framework for computational offloading in mobile cloud computing. *J. Supercomput.* **68**(2), 978–995 (2014). doi:[10.1016/j.jnca.2014.04.009](https://doi.org/10.1016/j.jnca.2014.04.009).
28. Shiraz, M., Gani, A., Shamim, A., Khan, S., & Ahmad, R. W. (2015). Energy Efficient Computational Offloading Framework for Mobile Cloud Computing. *Journal of Grid Computing*, *13*(1), 1–18.