

# Computation Offloading: Overview, Frameworks and Challenges

Jaya Ashok Suradkar

M. E. Student, Dept. of Computer, Engineering  
Dr. D. Y. Patil Institute of Engineering and  
Technology, Pune, India.

R. D. Bharati

Professor, Department of Computer Engineering  
Dr. D. Y. Patil Institute of Engineering and  
Technology, Pune, India.

## ABSTRACT

In spite of major advances in a mobile cloud computing, applications are still very limited due to poor computation processing power, memory and limited battery life of a mobile device. Recently, offloading computation intensive task to the cloud has become a great solution to improve both battery life and performance of mobile devices. Offloading makes use of quality of services concept to remove mobile application limitations. This paper covers overview, frameworks and challenges of computation offloading.

## General Terms

Overview, process, frameworks and challenges.

## Keywords

Computation offloading; mobile cloud computing; mobile devices;

## 1. INTRODUCTION

In our daily life, mobile devices have become common entity. These mobile devices provide us with many more exciting applications which require large computing power, memory, network bandwidth and energy to run such applications. These devices' houses many sensors like accelerometer, proximity etc. used for different functions like multimedia, GPS navigation, real time games etc. which also adds energy consumptions constantly. As there is a great improvement in a battery technology, mobile devices still suffers from battery lifetime problem. Energy or Battery is the only resource in mobile devices that cannot be restored immediately and needs external resources to be renewed. Computation offloading is a way to improve performance and save energy. Although several efforts are facilitating and promoting computation offloading, still research and development in this domain remains a top priority. Such a challenges faced by mobile devices can be overcome by successfully shifting computation intensive part of the mobile application to the remote servers and then retrieve results from these servers to mobile devices. However, managing these servers is not easy task. In such scenario cloud computing provides the better solution. Cloud computing provide services based on pay-as-you-use principle like water, electricity, telephone etc. and provides a model for "enabling ubiquitous, 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", as NIST[1] describes. Mobile cloud computing is a combination of cloud computing, wireless communication technologies, mobile web, location based services and computing devices. There are many mobile cloud applications such as android applications, biometric application, social media applications,

mobile commerce application, mobile healthcare applications etc. are using software as a service model. These applications or services can offload.

## 2. RELATED WORK

### 2.1 Works on Energy Saving

Improving battery lifetime is one of the most difficult design objectives of mobile devices because they usually come with a limited battery capacity. Few of them adopted profiling partitioning method to identify offloaded parts of an application for saving energy of mobile devices [3], [4], [6], [7], [8]. Authors in [5] designed a similar method to determine whether Java methods and byte code to native code compilation should be executed at remote servers to save energy but their methods cannot be applied to dynamic workload as they have variance of data input at runtime. Some works develop energy models and they construct cost graph or make offloading decisions but parameters such as bandwidth variation dynamic workload and idle-mode energy consumption are not included which results in incorrect partitions or incorrect offloading decisions [9], [10], [11], [12], [13]. Recently, some of the authors studied the feasibility of both computation offloading and software backups but they have not discussed in depth that how to measure energy consumption of mobile devices [14]. In [15] authors develop an energy model which ensures higher accuracy than earlier works and also designed a software based method for energy estimation. Designed and implemented TDM (Ternary Decision Making) framework for computation offloading which is suitable for various decision parameter.

### 2.2 Works on Time Saving

Quick responsiveness of mobile applications is needed as they are usually real time and interact to user. To reduce execution time several research efforts have been done [2], [18], [19], [20]. Authors in [2] designed an offload engine that dynamically partitions an application when required resources reach the maximum capacity of the mobile devices. In [15], to increase portability authors developed Linux program at user space so that modifications of OS or drivers are not required and there is no need to depend on a specific software packages.

### 2.3 Works on Energy and Time Saving

Optimizing time and energy for mobile devices simultaneously are difficult design objectives and very few research efforts have adopted on energy and time saving simultaneously. Computation offloading frameworks such as ThinkAir, Cuckoo were developed by some authors to execute applications on cloud but energy and time estimation model were not given [16] [17]. In [15] authors proposed energy and

execution model which can achieve higher accuracy and better performance in time and energy saving. Also, designed and implemented a decision framework for computation offloading.

### **3. COMPUTATION OFFLOADING FRAMEWORK**

#### **3.1 Cuckoo**

Cuckoo is very simple programming model which is mainly focuses on android platform. Cuckoo integrates with the popular open source Android framework and the Eclipse development tool that allows for a single interface with a local and a remote implementation. For a Cuckoo framework decision parameters are decided at run time as it follows dynamic offloading [17].

#### **3.2 SmartDiet**

SmartDiet [21] is a tool kit, to assist mobile application developers in creating code which is suitable for energy efficient offloading, limitation identification and software structure analysis. It uses method level application execution and calculates energy consumption for allocation of communication. The idea for the software structure analysis tool is that it analyzes the program code to find solutions to save energy through restructuring.

#### **3.3 Coca**

COCA stands for Computation Offload to Clouds using AOP (Aspect Oriented Programming). COCA is a programming framework that allows mobile application developers to offload part of the computation to servers in the cloud easily. This framework works at the source level. COCA's aim is to offload partial computation from resource limited mobile devices to cloud servers. Framework provides a source language mechanism to generate remote objects and local mobile device program automatically, allowing different tradeoffs between communication overhead and computation improvement. COCA can achieve computation effectiveness and energy savings by offloading part of the computation to cloud for a chess game [22].

#### **3.4 A Game Theoretical Approach**

Game theory is an efficient framework to analyze the interactions among several mobile device users who act in their own interests and develop incentive rich computation offloading methods such that no user has the incentive to deviate unilaterally. This approach developed for achieving efficient computation offloading decision making among the mobile device users. Game theory is a beneficial tool for developing decentralized mechanisms with low complexity such that users can self-organize into a mutually satisfactory solution. This approach shows that game always admits Nash equilibrium for both heterogeneous and homogeneous wireless access [26].

#### **3.5 Thinkair**

ThinkAir is a structure for offloading mobile computation to the cloud. ThinkAir uses a concept of Smartphone virtualization in the cloud and results in a method level computation offloading. It focuses on the elasticity and scalability of the cloud and improves the power of mobile cloud computing by parallelizing method execution using multiple virtual machine (VM) images. Utilizing ThinkAir requires just few adjustments to an application's source code coupled with utilization of their ThinkAir tool chain. They also show that a parallelizable application can involve multiple VMs to execute in the cloud in a clean and on

demand manner such as to achieve effective reduction in execution time and energy consumption. Implementation of ThinkAir involves evaluation with a range of benchmarks starting from simple micro benchmarks to more complex application. They have ported Android to Xen allowing it to be run on business cloud base, and they keep on chipping away at enhancing developer help for parallelizable applications [23].

#### **3.6 MAUI**

MAUI is a framework that empowers fine grained energy aware offload of mobile code to the infrastructure. Past methodologies to these issues either depended on software programmer support to partition an application, or they were coarse grained need full VM migration. MAUI utilizes the profits of a managed code environment to offer the best of both worlds: it supports fine grained code offload to enhance energy savings with negligible load on the software programmer. MAUI chooses at run time which techniques should be remotely executed, determined by an optimization engine that attains to best vitality energy savings under the mobile phone's present connectivity constrains. In their assessment, they demonstrate that MAUI empowers: 1) a resource intensive application which is a face recognition application that consumes less energy, 2) game application which is latency sensitive that doubles refresh rate, and 3) a voice based language translation application that ignores the Smartphone environment by executing unsupported segments remotely [24].

#### **3.7 CloneCloud**

CloneCloud system automatically transforms mobile applications to benefit from the cloud. The system is a flexible application partition and execution runtime that allows unmodified mobile applications running in an application level virtual machine to cleanly off load part of their execution from mobile devices onto device clones operating in a computational cloud. CloneCloud is a combination of static analysis and dynamic profiling. It partition applications automatically at a fine granularity while reducing execution time and energy use for a target computation and communication environment. At runtime, the application partitioning is enhanced by migrating a thread from the mobile device at a chosen point to the clone in the cloud, executing there for the remaining part of the partition, and reintegrating the migrated thread back to the mobile device. CloneCloud can adapt application partitioning to various environments, and can help few applications achieve as much as a execution speed up and a decrease of energy consumption on the mobile device [25].

## **4. COMPUTATION OFFLOADING**

### **4.1 Forms of Computation offloading**

#### **4.1.1 Computation Offloading Process**

Figure 1 shows process of computation offloading

#### **4.1.2 Partial Offloading**

In this type of offloading large computation intensive tasks are performed on a virtual machines and rest of the tasks executed on mobile device Results obtained from virtual machines and mobile devices collectively provided to the respective user in a required format.

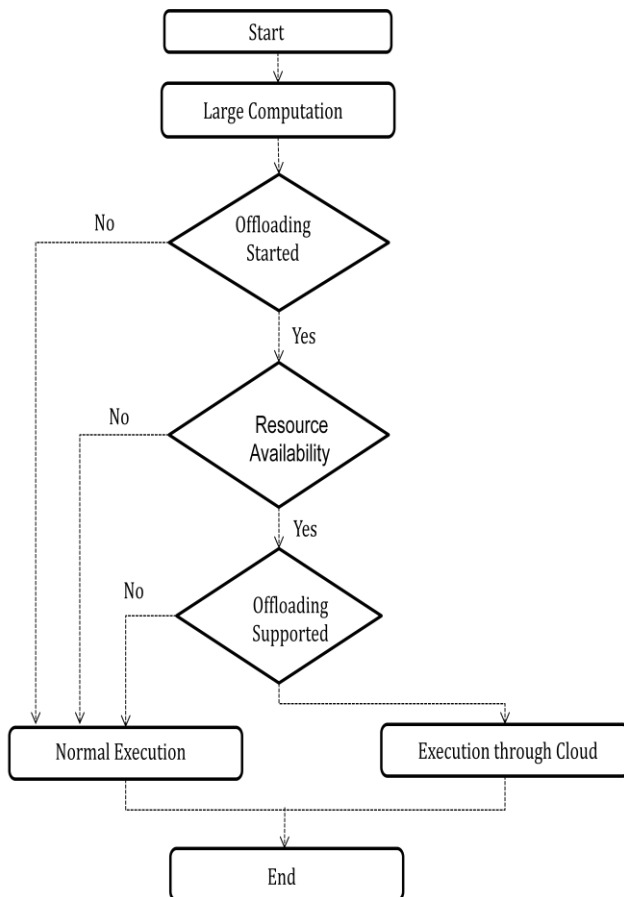


Fig 1: Computation Offloading Process

#### 4.1.3 Complete offloading

In case of Complete offloading whole application is kept on cloud and cloud act as an exact replica of the mobile device with the same functionality. The drawback of this type architecture is that the applications that fits into this architecture are very less, applications are available only the ones with very less or no user interaction.

#### 4.1.4 Task offloading

It is much different in that it involves the offloading of a specific task/process of an application or a series of tasks.

#### 4.1.5 Code Offloading

It involves offloading a particular part of an application's code for example an object, function or class to a remote server on the cloud.

### 4.2 Computation offloading Stages

For offloading the mobile application to the cloud, there are different stages involved in a computation offloading:

#### 4.2.1 Program Partitioning:

Before offloading the mobile application to the cloud, parts of the application that can offload need to be identified. This can be done by partitioning a program. There are different ways to partition the program as follows:

##### 4.2.1.1 Predefined Interface

Developer has a predefined interface in the system when the class implements this interface it is marked as off loadable

##### 4.2.1.2 VM Migration

The complete state of the VM would be migrated into remote server and there is alternative approach which is VM synthesis where the smaller state is transferred which causes more computation to be performed locally.

##### 4.2.1.3 Method Level

The program partitioning takes place at the method level. As the method is called for remote execution the state associated with this method is also transferred. The method is to be noted as off loadable which are analyzed during static analysis phase to make offloading decisions

##### 4.2.1.4 Thread Level

In case of thread level partitioning, less states are required to be transferred among client and server machine. Conventional thread migration techniques first execute a current thread which is suspended then capture Java stack, program counter, data state and send it to the target. Then thread is rescheduled for execution

#### 4.2.2 Static or Dynamic Decision

For static offloading, decision parameters are defined at development time while in case of dynamic offloading decision parameters are defined at run time of an application.

### 4.3 Challenges

Different challenges associated with computation offloading and mobile cloud computing. Some of them are listed below:

#### 4.3.1 Security

Security is one the important issues in mobile cloud computing as all the data of different users can be stored in the same cloud. Maintaining privacy of the users' data from various threats is a major.

#### 4.3.2 Reliability

To perform computation offloading to the cloud wireless network and cloud service play an important role. For effective computation offloading speed of the internet is extremely necessary to work efficiently. Limited options are available such as 3G and Wi-Fi. 3G has advantages due to its availability and speed but it consumes huge amount of energy. On the other hand Wi-Fi consumes less energy and has high bandwidth to enable computation offloading but it has less availability.

#### 4.3.3 Heterogeneity

WCDMA, WiMAX, GPRS, WLAN and CDMA2000 are several access technologies used by different mobile devices to access services offered by the cloud.

#### 4.3.4 Availability

Mobile users may not be able to connect to the cloud due to traffic congestion, lack of network coverage and various failures in the network

#### 4.3.5 Low Bandwidth

One of the biggest issue of mobile cloud computing and computation offloading is a bandwidth. The lack of wireless network access technologies with large speed and latency is a major issue.

## 5. CONCLUSION

When migrating applications to cloud based platform, careful design and planning is necessary to meet best performance with minimum cost. We surveyed an approach towards conserving energy on mobile devices which are resource

limited when offloading a computation also energy related challenges with respect to processing tasks on mobile devices. In order to understand the state of the art of computation offloading, we also gave an overview of the major technologies.

## 6. REFERENCES

- [1] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST special publication, 2011.
- [2] X.Gu, A. Messer, I.Greenberg, D.Milojicic, and K.Nahrstedt, "Adaptive offloading for pervasive computing," *IEEE Pervasive Comput.*, vol. 3, no. 3, pp. 66-73, Jul./Sep.2004.
- [3] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: A partition scheme," in *Proc. 2001 Int. CASES*, 2001, pp. 238–246.
- [4] 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 Int. Conf. MobiSys*, Jun. 2010, pp. 49–62.
- [5] G. Chen, B.-T. Kang, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, and R. Chandramouli, "Studying energy tradeoffs in offloading computation/ compilation in java-enabled mobile devices," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 9, pp. 795–809, Sep. 2004
- [6] S. Han, S. Zhang, and Y. Zhang, "Energy saving of mobile devices based on component migration and replication in pervasive computing," in *Proc. Ubiquitous Intell. Comput.*, Aug. 2006, pp. 637–647.
- [7] B. Seshasayee, R. Nathuji, and K. Schwan, "Energy-aware mobile service overlays: Cooperative dynamic power management in distributed mobile systems," in *Proc. 4th ICAC*, Jun. 2007, p. 6.
- [8] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *Proc. 6th ACM SIGOPS/EuroSys*, 2011, pp. 301–314.
- [9] K. Kumar and Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.
- [10] Y. Hong, K. Kumar, and Y. Lu, "Energy efficient content-based image retrieval for mobile systems," in *Proc. IEEE ISCAS*, May 2009, pp. 673–1676.
- [11] X. Zhao, P. Tao, S. Yang, and F. Kong, "Computation offloading for H.264 video encoder on mobile devices," in *Proc. IMACS*, Oct. 2006, pp. 1426–1430.
- [12] C. Cai, L. Wang, S. U. Khan, and J. Tao, "Energy-aware high performance computing: A taxonomy study," in *Proc. 17th IEEE ICPADS*, 2011, pp. 953–958.
- [13] S. U. Khan, "A goal programming approach for the joint optimization of energy consumption and response time in computational grids," in *Proc. 28th IEEE IPCCC*, Dec. 2009, pp. 410–417.
- [14] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? The bandwidth and energy costs of mobile cloud computing," in *Proc. IEEE INFOCOM*, 2013, pp. 1285–1293.
- [15] Ying-Dar Lin; Chu, E.T.-H.; Yuan-Cheng Lai; Ting-Jun Huang, "Time-and-Energy-Aware Computation Offloading in Handheld Devices to Coprocessors and Clouds," in *Systems Journal, IEEE* , vol.9, no.2, pp.393-405, June 2015.
- [16] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. 31st Annu. IEEE (INFOCOM) Int. Conf. Comput. Commun.*, 2012.
- [17] R. Kemp, N. Palmer, and T. Kielmann, "Cuckoo: A computation offloading framework for smartphones," presented at the *Proc. 2nd Int. Conf. Mobile Comput., Appl., Serv.*, Santa Clara, CA, USA, 2010.
- [18] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using bandwidth data to make computation offloading decisions," in *Proc. IEEE IPDPS*, Apr. 2008, pp. 1–8.
- [19] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso, "Calling the cloud: Enabling mobile phones as interfaces to cloud applications," in *Proc. 10th ACM/IFIP/USENIX Int. Conf. Middleware*, Dec. 2009, pp. 1–20.
- [20] K. Yang, S. Ou, and H. Chen, "On effective offloading services for resource-constrained mobile devices running heavier mobile Internet applications," *IEEE Commun. Mag.*, vol. 46, no. 1, pp. 56–63, Jan. 2008.
- [21] A.Saarinen, M.Siekkinen, Y.Xiao, J. Nurminen, M. Kempainen, P. Hui, "SmartDiet: Offloading Popular Apps to Save Energy ",Pub. in: *ACM SIGCOMM 2012* conference held at Helsinki, Finland, 2012,,ISBN: 978-1-4503-1419-0/12/08.
- [22] H. Chen, Y. Lin, and C. Cheng., " COCA: Computation Offload to clouds using AOP." Pub. in: In *12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, Ottawa, ON,Canada, ,May 2012, pp.: 466-473, ISBN:978-0-7695-4691-9.
- [23] Kosta, S., Aucinas, A., Hui, P., Mortier, R., Zhang, X.: ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In: *IEEE INFOCOM*, pp. 945–953 (2012)
- [24] Cuervoy, E., Balasubramanianz, A., Cho, D.: MAUI: Making smartphones Last Longer with Code Offload. In: *MobiSys 2010* (2010)
- [25] Chun, B., Ihm, S., Maniatis, P., Naik, M., Patti, A.: CloneCloud: Elastic Execution between Mobile Device and Cloud. In: *ACM Workshop EuroSys*, pp. 301–314 (2011).
- [26] Xu Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," in *Parallel and Distributed Systems, IEEE Transactions on* , vol.26, no.4, pp.974-983, April 1 2015. doi: 10.1109/TPDS.2014.2316834