Article title: Elegant Computational Intensive Task Offloading Scenario for Android
Authors: MAHESH KALUTI[1]
Affiliations: Computer Science & Engineering, P E S College of Engineering Mandya, VTU[1]
Orcid ids: 0000-0002-9919-7119[1]
Contact e-mail: mahesh.rcet@gmail.com
License information: This work has been published open access under Creative Commons Attribution License http://creativecommons.org/licenses/by/4.0/, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Conditions, terms of use and publishing policy can be found at https://www.scienceopen.com/.
Preprint statement: This article is a preprint and has not been peer-reviewed, under consideration and submitted to ScienceOpen Preprints for open peer review.
DOI: 10.14293/S2199-1006.1.SOR-.PPJHAVB.v1
Preprint first posted online: 04 November 2021
Keywords: Technical Glitch, offloading techniques, intensive tasks
Elegant Computational Intensive Task Offloading Scenario for Android

Dr Mahesh Kaluti

1 Associate Professor, Department of CS&E, P. E. S College of Engineering Mandya
mahesh.rkcet@gmail.com

Abstract. Despite the technical changes and enormous day by day upgradation in the field of mobile computing the smart devices as well as IOT devices had experienced tremendous technical glitch, which narrow’s the life span and survivability of small scale processing devices. Today, end users are becoming more demanding and are expecting to run computational intensive tasks on their Smart phone devices and IOT devices. Therefore, virtual cloud computing (VCC) integrates local device computing and Cloud Computing (CC) in order to extend computational capabilities of smart phone devices and IOT devices using cloud offloading techniques. Computation Offloading tackles limitations of Smart phone devices and IOT devices such as limited battery duration, limited computational capabilities, and limited storage capacity by offloading the execution and workload to cloud which has better systems with better computation and storage capabilities.

This paper aims to present the techniques to offload computational intensive tasks to cloud framework and analyses them along with traditional local execution techniques and their issues. Furthermore, it explores other important parameters based on which the applications are implemented such as offloading technique and partitioning of tasks.

Keywords: Technical Glitch, offloading techniques, intensive tasks

1 Introduction

1.1 Task offloading

Nowadays, user preferences for computing have been impacted by the latest research and advancements in various computing technologies. Major reports and studies have highlighted the importance of MCC and its impact on smart phone devices and IOT devices. The total Internet of Things (IoT) market was worth around 389 billion U.S. dollars in 2020, and is expected to rise to more than one trillion U.S. dollars in 2030. Moreover, the usage of smart phones and IOT Devices has increased drastically in various fields including, wearable, traffic monitoring, fleet management, agriculture, hospitality, smart grid and energy saving, water supply and health care. Although the
predictions that Smart phone devices and IOT Devices will be dominating the future tech industry, these devices along with their applications are still restricted by some limitations such as computational capacity, battery life, and the storage capacity. Now days the devices used in the field of mobile communication are upgraded with many techniques along with sufficient memory, however, it is still not sufficient to help with computing intensive tasks such as image recognition, natural language processing (NLP), 3D- simulation and decision-making. Smart phone devices and IOT devices provide less computational power comparing to personal computer (PC) or cloud servers and computation-intensive tasks put heavy loads on battery life, processor and storage. Currently, there are various techniques Cloud Computing (CC) that aim at improving the computational capabilities of resource-constrained Smart phone client devices and IOT client devices by providing these clients access to cloud infrastructures, software, and computing services. There are various frameworks that allow to process computational intensive tasks remotely on cloud servers. And recent applications specially high end applications and processing of these applications made through smart devices will reduce the burden on these up to certain extent and this will reduce the hurdle for manufacturer to offload the work to cloud environment and the consistency offered by the cloud environment will also ensures the reliability of the system.

![Task Offloading](image)

**Task Offloading**

1.2 **Computation offloading** is the transfer of computational intensive tasks to a separate processor, such as cloud servers. Offloading to a cloud server can be used to accelerate applications including; image processing and mathematical calculations, simulation, and natural language processing (NLP). Offloading computational intensive tasks to an external platform over a network can provide computational edge and helps to overcome hardware limitations of a device, such as limited computational power, storage, and energy.
The main objective of the project is to present a framework that automatically offloads computational intensive tasks of the clients such as Smart phone devices and IOT devices to virtual machine in the cloud.

2. Proposed System
Computation Offloading is the process of sending computational intensive tasks to a cloud server. But the challenge is offloading and the threshold value behind that, at what extent the system need to make decision regarding the condition where the application need to meet all criteria during the run time environment of applications and before offloading the designer need decide the factors However, the computation offloading techniques are still facing many challenges. The potential of mobile offloading mainly depends on the mobile network technologies like cellular and Wi-Fi. They determine the viability of mobile offloading. Today, the Wi-Fi technology is during a position to supply high bandwidth connections. However, the data transmission using the cellular network requires a considerable amount of energy from the mobile device as against a Wi-Fi network. The figure illustrates the environment that supports computation offloading. During this overview, the mobile device decides to dump method B to a cloud server or a powerful machine. The cloud platform here provides the virtual computational resources to run the offloaded tasks. The powerful machine is a server or cluster in an exceedingly computing centre, or a computing grid, or a virtual server within the cloud.

![System Architectural View](image-url)
2.1 Algorithms Used

Operation Modes: The framework will operate on the IoT device using the tasks written by the developer. The developer can run the framework in one of the following two schedulers:

- **Manual**: The developer can manually configure which tasks to run on the IoT device. All remaining tasks will be offloaded to the cloud.
- **Automatic**: The developer specifies the expected throughput of data processing. The framework will dynamically decide which tasks to run on the IoT device and which tasks to offload to the cloud in order to meet the specified throughput.

**Automatic Scheduler:**

In automatic Scheduler the developer can specify the expected throughput of Data processing. Given this expected throughput, the framework will dynamically decide which tasks to offload to the cloud to meet the throughput requirement.

Let Q be the developer-specified throughput requirement. Essentially, the Algorithm offloads tasks at the end of the IOT device task list until the last Task meets the required throughput Q. Initially, the framework will run all Tasks [1, 2, ..., n] on the IOT device. The algorithm will then run the following steps:

1. For a short period of time, the algorithm will measure the current throughput q of the last task T on the IoT device.
2. If the measured throughput is less than the desired throughput (q < Q), then the last task T must be offloaded to the cloud. The algorithm will notify the framework to route all messages for task T to Cloud Pub/Sub, allowing Google Cloud to run that task.
3. Otherwise, the IoT device has met the throughput requirement Q. No more tasks must be offloaded to the cloud, so the algorithm can terminate. For a short period of time, the algorithm will measure the current throughput q of the last task T on the IOT device.

**Test Results**
Conclusion

The proposed model analyses the existing research of computational offloading and gives a simple comparison of computational offloading framework in MCC. The issues and challenges on computational offloading in MCC are presented for future research.

References

1. K. Liu J, Lu Y H, et al (2013). A Survey of Computation Offloading for Mobile Systems[J]. Mobile Networks & Applications, 18(1):129-140.
2. Shiraz M, Gani A, Shamim A, Khan S, Ahmad RW (2015) Energy Efficient Computational Offloading Framework for Mobile Cloud Computing. Journal of Grid Computing 13 (1):1-18.
3. Shiraz M, Gani A, Ahmad RW, Adeel ASS, Karim A, Rahman ZA (2014) A lightweight distributed framework for computational offloading in mobile cloud computing. Plos One 9(8).
4. Folino G, Pisani FS (2014) Automatic offloading of mobile applications into the cloud by means of genetic programming. Applied Soft Computing 25:253265.
5. Arora M, Kalra M, Singh S ACOF: Autonomous computation offloading framework for Android using cloud. In: Information Management in the Knowledge Economy (IMKE), 2013 2nd International Conference on, 2013. pp 143-149.
6. Kovachev D, Yu T, Klamma R Adaptive Computation Offloading from Mobile Devices into the Cloud. In: Parallel and Distributed Processing with Applications (ISPA), 2012 IEEE 10th International Symposium on, 2012. pp 784-791.
7. Baccarelli E, Cordeschi N, Mei A, Panella M, Shojafar M, Stefa J (2015) Energy-efficient Dynamic Traffic Offloading and Reconfiguration of Networked Datacenters for Big Data Stream Mobile Computing: Review, Challenges, and a Case Study. IEEE Network:1-21.
8. Zhang Q, Yang LT, Chen Z (2015) Privacy Preserving Deep Computation Model on Cloud for Big Data Feature Learning. Computers, IEEE Transactions on PP (99):1-1.