A Literature Study on Various Factors Affecting Computational Offloading Performance

PRAVEEN RAJ
M.Tech-Student, Computer Science and Engineering
R V College of Engineering, Bangalore, Karnataka, India

Abstract: **Mobile systems, such as smart phones, tablets, have become the primary resources of computation for many users. Many surveys have shown that longer battery lifetime as the most important feature of such systems. Consumers spend more time on media through mobile applications. Many mobile applications are too computation intensive to perform on a mobile system such as games, image processing and many more. The hardware limitations of mobile devices for higher performance and/or energy savings can be addressed by offloading computationally intensive task to external resource. There are many issues associated with computation offloading such as network bandwidth, intermittent connectivity, the transmission delays, the distance of remote computing resources from primary computing resource. This paper presents a literature study on research work done till date with respect to computation offloading strategies developed to overcome these challenges.**

Keywords : **Mobile Systems; Smart Phones; Offloading; Mobile Applications;**

I. INTRODUCTION

According to various statistical reports, the combined sales of smartphones and tablets has already surpassed sales of Desktops. IDC Prediction report[1] says Mobile app downloads already hit 150 billion and statistics shows total number of mobile applications available in leading app store has crossed 3.6 million. There are many energy hungry applications, for example, face recognition, speech and object recognition, natural language processing, audio/video encoders and decoders, computer vision and graphics, and mobile augmented reality. Mobile devices, including smart phones, tablets and netbooks, are also constrained with small battery size, CPU, and storage. At the same time, current battery technology is also a bottleneck in mobile devices and this result in a growing gap between the demand of energy and the capacity of the resources of mobile devices. Survey[2] shows short battery life was the most disliked feature for many mobile devices.

Therefore Offloading computational intensive task to the external resource can solve the challenges of saving power consumption at mobile device and improve user experience. With the rapid adoption of cloud computing concepts and the explosive growth in the usage of mobile devices, Mobile Cloud Computing (MCC), the availability of cloud computing services in a mobile ecosystem, has been introduced as a potential technology for mobile services.

II. LITERATURE SURVEY

Offloading computational intensive task to external resource could solve the challenges of increasing battery lifetime of mobile system and improve application performance. So far it has been identified three possible ways of offloading computational intensive task to the external resource. The three possible ways are offloading task to the cloud, cloudlet and neighbouring mobile device itself(other than initiator). The mobile device that intiates task offloading may bec called as Offloader and the external resource that executes the task may be called as offloadee. Following are various factors that affect the task offloading mechanism

i) Communication Technology: Various communication technologies such as 3G, 4G, WiFi and Bluetooth used in Mobile devices plays an important role in computation offloading to cloud, cloudlet and neighbouring mobile devices interms of energy savings and application performance, while it is found that bluetooth technology consumes less energy but at the increased cost of application performance, while 3G and WiFi helps improve application performance but at the cost of increased energy consumption while offloading computation and data.

ii) Round Trip time or latency: More the distance between offloading device and external computing device more is RTT, Therefore offloading to the external depends on the RTT, Offloading to neighbouring external device takes less latency, RTT is the performance bottleneck while offloading computation to the cloud.

iii) Speed: how fast computation can be offloaded to external device, Different communication technologies have their own speed. Increase in speed helps in improving application performance.

iv) Communication Stability: Communication stability is a major concern when offloading computation to neighbouring mobile devices.

v) Energy conservation: In case of offloading computation to neighbouring mobile devices, it is imporatant to balance remaining energies between the offloader and offloade devices.

vi) External Device capabilities: Cloud systems and cloudlets typically poses high end infrastructure compared to Smart mobile devices.

Vii) Cost: Cloud computing costs more as it is pay on use concept, offloading to cloudlet and neighbouring mobile devices cost very less.

Goals of Computational Offloading Strategies

The goals of any computational offloading strategy is to reduce power consumption and execution time at local mobile device.

1) Saving energy at local device

Offloading computational intensive tasks to remote device or cloud saves the energy consumption at local mobile devices, thereby increases the battery lifetime of the device. [3] and [4] formulate energy saving for computation offloading to the cloud as follows:

$$E_m - (E_i + E_{tr}) - (1)$$

where

• E_m - energy consumed by a mobile device when computation happen on the device
• E_i - energy consumed by a mobile device in idle state to wait for computation result when computations happen on other computational resources
• E_{tr} - energy consumed by a mobile device when sending and receiving data between the device and the other computational resource

If $E_m - (E_i + E_{tr})$ is positive, then it seems to better offload the task to external resource-intensive

If $E_m - (E_i + E_{tr})$ is negative or zero, then it seems to better not to offload the task to external resource

2) Improve Application Performance by reducing task completion time

The task completion time can be termed as Service time, from [5] Service time can be formulated as

Service time = Transmission time + Computing time
Transmission time is formulated as
Transmission time = DLoad/Bw + Latency

DLoadis Data transmission size, Latency is the time delay of transmission from mobile client to external resource and Bwis bandwidth of the link targeting moblile client.

Computation time at external resource is given as

Computation time = WLoad/ Cr + γ

Where, $WLoad_k$ is Computation work load, Cr_j is the computaiton capacity of external resource and γ is a slack variable to formulate the variability

Computation offloading for any task seems to be better when service time, the time consumed when task is offloaded for remote execution is less than execution of task at local mobile device.

Based on acheiving above mentioned goals of computation offloading strategies, various methodologies have been proposed, some are based on improving application performance, some are based on reducing energy consumtion at local offloading device and remaining are based on acheiving a trade-off between the two goals.

Based on two goals of computation offloading strategies Table-1, Table-2 and Table-3 breifly describes various methodologies proposed, challanges it addresses

Table-1 Methodologies Based on Improving Application Perfromance

Paper	Architecture/ Algorithms	Offloading strategy	Challenges Addresed
[6]	Proposes Serendipity System and offloade selection algorithms	Offoading task to neighbouring mobile devices	Task allocation among intermittently connected device
[7]	Uses AHP and fuzzy TOPSIS method to determine suitable cloud for task offloading	Selection of optimal cloud path	Determines the relative importance of criteria for selection of optimal cloud
[11]	Collaborative Mobile-to-Mobile opportunistic offloading architecture	Offoading task to neighbouring mobile devices	Addresses the challenges of a dividing a task into subtasks and offloading replicated subtasks to neighbouring mobile devices
[5]	Proposes ACO-based service offloading algorithm	Offloading task to cloud in an optimized network path	Optimize the network path using Ant colony optimization technique

Table-2 Methodologies Based on Reducing Energy Consumption

Paper	Architecture/ Algorithms	Offloading strategy	Challenges Addresed
[10]	Proposes Adaptive Offloading Algorithm	Offoading task to cloud/external PC	For any given task, determines the type of external resource for computational offloading

Table-3 Methodologies Based on acheiving trade-off between reducing energy consumption and improving application performance

Paper	Architecture/ Algorithms	Offloading strategy	Challenges Addresed
[8]	Proposes Power Balancing Algorithm	Offoading task to neighbouring mobile devices	Maximize the lifetime of a highly collaborative mobile device cloud
[9]	Towards Computational Offloading in Mobile Device Clouds	Offoading task to neighbouring mobile devices	Runs social based offloadee selection algorithms to select external device for computation
[12]	Femto-Cloud Architecture	Offoading task to neighbouring mobile devices	Addresses the challanges of allocating tasks to neighbouring mobile devices
[13]	Droid Cloud Architecture	Offoading task to neighbouring mobile devices	Offloading task dynamically at runtime to neighbouring mobile devices and giving choice to developer, to which part of code to be offloaded and also made parallelizable

Description of Proposed Computation Offloading Strategies

The External resources that performs the offloaded computation can be categorised into Cloud, Cloudlet and Neighbouring Mobile Device. Some of the strategies have been briefly described below.

A. Offloading Computation to cloud/cloudlet

The capabilities of cloud resources are exceptionally high, the bandwidth and latency associated with network and the cost of offloading are the challenges that need to be addressed while offloading computational intensive task to the cloud. Several strategies have been proposed in order to address these challenges. Some of them have been briefly described below

i) Cloud-Path Selection Algorithm

Choosing the optimal cloud to save execution time consumed by offloading to cloud is normally easier when considering only one factor. However, there are many criteria such as bandwidth, speed, security, price, and availability that need to be considered when making final decisions. A multiple criteria decision analysis approach based on the analytic hierarchy process (AHP) and the technique for order preference by similarity to ideal solution (TOPSIS) in a fuzzy environment is proposed to decide which cloud is the most suitable one for offloading[7]. The AHP is used to determine the weights of the criteria for cloud-path selection, while fuzzy TOPSIS is to obtain the final ranking of alternative clouds. The numerical analysis is performed to evaluate the model.

ii) ACO-based Service Offloading

Ant colony optimization (ACO) initially proposed by Marco Dorigo in 1992, aims to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. ACO - Based algorithm tries to give knowledge of optimized path to each smart-mobile devcies, requesting for remote cloud resource for task execution[5]. Each client device, which sends the request for task execution, can be treated as the colony and the services located on available remote servers are treated as the food. Ants start their searching from the colony. Proposed strategy assumes that ants will not go back to the entities it has visited. So, the offloading problem is translated as searching for the food for ants in that colony.

iii) Adaptive Offloading Algorithm

It may not be always necessary to offload all computations to a remote cloud server from a mobile device mainly due to the high communication delay that may be generated. Instead, the offloading target can be a nearby PC via a lower communication cost mechanism, such as Bluetooth, depending on the computation demands[10]. The impact of various factors, such as computation workload, file size, and wireless communication protocols, and have investigated insights on power consumption through offloading to a nearby PC or a remote cloud server. An adaptive algorithm is developed based on experimental results to automatically select a computation resource which has capacity to execute computationally intensive applications and thus to save energy consumption of mobile devices.

B. Mobile-to-Mobile Computation Offloading

Offloading comuptations to mobile device opens the door for avoiding RTT and network latency challanges faced while offloading tasks to distant clouds. MDCs are becoming a reality with the increase in average mobile devices per user or household [13] [14]. Overall, offloading to an Mobile device Cloud registers up to 80% and 90%

savings in time or energy respectively as opposed to offloading to the cloud and 20% and 35% savings in time or energy respectively by offloading to an MDC as opposed to a cloudlet[18]. Mobile devices are normally have intermittent connectivity between them. Various srategies have been developed to address this challenge.

i) Social Based Algorithms

Several Social based algorithms have been proposed[14][15][10] based on 4 Types of social connections: friendship-based, common interest-based and contact history based . These are determined by leveraging social interaction information. User mobile devices usually store and update one or more social databases such as user friendship graph(e,g facebook), These Algorithms use such databases to select the best offloadee candidate among the set of neighboring device at time t. The goal of such algorithm is to select stable connection with neighboring nodes in order to ensure that the connection duration between the two encountered nodes remains during (i) the migration of the task (i.e., exchange of data), (ii) task execution in the remote node, and (iii) sending back the results. The evaluation of such algorithms is based on success rate of offloaded task. Success rate of each task defined as the fraction of completed tasks within any time t. A 100% success implies that all tasks and subtasks have been migrated, executed, and results were sent back successfully.

ii) Significance of Predicting user presence time

Presence time of device one with respect to other device is the duration of two devices are in proximty with one another. Accurate prediction of presence time, helps the device that performs offloading task to find more stable device among a set of intermittently connected devices. Predicting the device presence time significantly decreases the task completion time and decrease the network load[12][6] .

iii) Importance of offloading replicated tasks

Without replication, the connection between the two nodes can potentially fail (e.g., , intermittent connectivity, node failure, network failure). Failure cause a huge delay and therefore considerable degradation in offloading performance. Offloading replicated tasks to other mobile devices in mobile device cloud, significantly improves the application performance[11], by decreasing the task completion time. This mechanism also faces limitations as overall energy consumed is more when offloading replicated tasks than offloading tasks without replication

Limitations of computational offloading mechanism

As the main objective of computation offloading in mobile devices is to reduce mobile device energy consumption and improve application performance. A mobile user performing computation in the cloud depends on the wireless network and cloud service. The capabilities of cloud is exceptionally high when compared to smart mobile devices, But offloading computation to the cloud or cloudlet may not be the optimal solution as it takes large Round trip time to offload the computations to distant clouds, further network latency adds up still more delay[9][10]. Dependence on the wireless network implies that cloud computing may not even be possible, let alone energy efficient, when connectivity is limited. Thus affects the reliability of Mobile Cloud Computing. Offloading computational tasks to cloudlets does not provide promising solution when compared with offloading to mobile device cloud, where task initiator and task executer are both mobile devices[9]. Even though ACO based algorithms[9] and cloud path selection algorithms [7] provides solution for finding optimized path in network and optimal cloud path selections based on criteria such as bandwidth, speed, security, availability and price, respectively, these algorithms need to be run more frequently due to changing network topology.

Serendipity system [6] Addresses the challenges of mobile to mobile task offloading when the network is intermittently connected, but it does not tackle the energy consumption challenges. Femto-Cloud[12] deconstructs cloudlet into a centralized controller but at the same time the controller becomes the single point of failure. Collaborative mobile device cloud [11] comes with an innovative idea of offloading replicated tasks to other mobile devices, it does not take into consideration of device remaining energies at offloadee site. Balancing energy among mobile devices and providing incentives to smart mobile device users for contribute their devices to mobile device cloud environment pose great challenge. Algorithms proposed[8] to balance energy among ensemble of mobile devices assumes that all task run simultaneously and there are no dependencies between any two tasks. In cloud computing, offloading of data to the cloud has implications for privacy and security. Because the data is stored and managed in the cloud, security and privacy settings depend on the IT management the cloud provides. Cloud service providers typically work with many third-party vendors, and there is no guarantee as to how these vendors safeguard data. Obviously, Some type of data cannot be stored in the cloud considering the privacy and security issues. One possible solution is to encrypt the data before offloading. But encryption alone cannot solve the problem. A technique called Steganography is used to hide the data from the cloud vendor[3].

III. CONCLUSION

This paper has presented a literature study on research associated with computation offloading for mobile systems. Various factors that affect computation offloading performance have been described. Different types of architectures and algorithms used to partition and offload tasks have been surveyed in order to improve performance or save energy. Some of the research areas associated with computation offloading strategies have been briefly described. This paper motivates the use of mobile devices in computational offloading in order to save time and energy.

IV. ACKNOWLEDGEMENTS

The author thanks Dr.S.Sridhar, Professor and Director, Cognitive & Central Computing, R.V.College of Engineering, Bangalore, India for communicating this article to this Journal for publication after modifications. The author would like to thank Principal and HoD/CSE of RVCE for the support provided.

V. REFERENCES

[1] Frank Gens IDC Predictions 2015 Team, IDC Predictions 2015: Accelerating Innovation — and Growth — on the 3rd Platform

[2] Brad Reed on Nov 26, 2013, http://bgr.com/2013/11/26/smartphone-survey-worst-features/

[3] K. Kumar., Y. H. Lu, “Cloud computing for mobile users: Can offloading Computation Save Energy?”, Computer, vol.43, no.4, April 2010, pp.51–56.

[4] K. Kumar, J. Liu, Y. H. Lu, B. Bhargava, “A Survey of Computation Offloading for Mobile Systems”, Mobile Networks and Applications, vol.18, Feb. 2013, pp.129–140

[5] Weidong Bao,Haoran Ji , Xiaomin Zhu, Ji Wang Wenhua Xiao and Jianhong Wu, “ACO-BASED SOLUTION FOR COMPUTATION OFFLOADING IN MOBILE CLOUD COMPUTING”, Big Data and Information Analytics c American Institute of Mathematical SciencesVolume 1, Number 1, pp 1-13,January 2016

[6] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura. Serendipity: Enabling remote computing among intermittently connected mobile devices. In ACM MobiHoc, 2012.

[7] Huaming Wu, Qiushi Wang and Katinka Wolter, “Methods of Cloud-path Selection for Offloading in Mobile Cloud Computing Systems” 2012 IEEE 4th International Conference on Cloud Computing Technology and Science, pp 443-448, 2012

[8] Abderrahmen Mtibaa, Afnan Fahim, Khaled A. Harras, Mostafa H. Ammar, “Towards Resource Sharing in Mobile Device Clouds: Power Balancing Across Mobile Devices” MCC '2013 Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing,Pages 51-56, ACM New York, NY, USA, 2013.

[9] Abderrahmen Mtibaa, Khaled A Harras, Afnan Fahim,”Towards Computational Offloading in Mobile Device Clouds”, 2013 IEEE International Conference on Cloud Computing Technology and Science pp 331-338, 2013.

[10] Kelie Zhan, Chung-Horng Lung, Srivastava, P.”A Comparative Evaluation of Computation Offloading for Mobile Applications” 2014 IEEE International Conference on Internet of Things (iThings 2014), Green Computing and Communications (GreenCom 2014), and Cyber-Physical-Social Computing (CPSCom 2014), pp 518-525, 2014.

[11] Abderrahmen Mtibaa, Mohammad Abu Snober, Antonio Carelli, Roberto Beraldi, Hussein Alnuweiri. “Collaborative Mobile-To-Mobile Computation Offloading”. 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2014), pp 460-465,2014

[12] Karim Habak, Mostafa Ammar, Khaled A. Harras , Ellen Zegura. “FemtoClouds: Leveraging Mobile Devices to Provide Cloud Service at the Edge”, 2015 IEEE 8th International Conference on Cloud Computing. Pp 9-16, 2015

[13] CDB. (2011, January) Household penetration rates for technology across the digital divide. [Online]. Available: http://communities-dominate.blogs.com/brands/2011/01/household-penetration-rates-for-technology-across-the-digital-divide.html

[14] Bloomberg. (2012, August) Average household has 5 connected devices, while some have 15-plus. [Online]. Available: http://goo.gl/ob6Tkh

