www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume - 3 Issue - 8 August, 2014 Page No. 7738-7743

Conserve Energy by Optimally Executing Mobile Applications in Mobile device or Offloading to Cloud

Bhavatharini M G, Veena A, Thejas Chikkalingappa M G

[1] *M.Tech Student, Computer Science Dept, Channabasaveshwara Institute of Technology, Karnataka, India*
Email id ***mgbhavatharini@gmail.com***
[2] *Asst. Professors, Computer Science Dept, Channabasaveshwara Institute of Technology, Karnataka, India*
Email id ***veena.a@cit.tumkur.org***
[3] *M.Tech Computer Science Dept, Visvesvaraya Technological University (VTU), Karnataka, India*
Email id ***thejascmg@gmail.com***

Abstract

Mobile systems have limited resources, such as battery life, storage capacity, and processor performance. These restrictions overcome by computation offloading. Energy efficiency is a fundamental consideration for mobile devices. Mobile Cloud Computing (MCC) is a model for flexible growth of mobile device capabilities via universal wireless access to cloud storage and computing resources. Rather than conducting all computational and data operations locally, MCC takes advantage of the abundant resources in clouds to store and process data for mobile devices. Our objective is to conserve energy for the mobile device, by optimally executing mobile applications in the mobile device (i.e., mobile execution) or offloading to the cloud (i.e., cloud execution). Our framework for energy optimal execution of applications derived a condition when offloading is beneficial and amount of energy saved by optimal execution of application. Proposed system demonstrated significant gain in execution speed and battery life of mobile phones.

Key Words: **Mobile cloud computing, computation offloading, optimal execution, energy analysis, Offload Decider**.

---***---

1. INTRODUCTION

Together with an explosive growth of the mobile applications and emerging of cloud computing concept, mobile cloud computing (MCC) has been introduced to be a potential technology for mobile services. The tension between resource-hungry applications and resource-poor mobile devices is considered as one of the driving forces for the evolution of mobile platforms. Due to the limited physical size, mobile devices are inherently resource-constrained [1], equipped with a limited supply of resources in computation, bandwidth, energy and storage. In particular, the energy supply from the limited battery capacity [2] has been one of the most challenging design issues for mobile devices. The limited battery life has been found by market research as the biggest complaint for smart phones [3]. Therefore, resource limitations in the mobile devices should be considered for the design of the mobile applications [21].

MCC is a model for elastic augmentation of mobile device capabilities via ubiquitous wireless access to cloud storage and computing resources, with context-aware dynamic adaption to changes in the operating environment. Rather than conducting all computational and data operations locally, MCC takes advantage of the existing resources in cloud platforms to gather, store, and process data for mobile devices. Many popular mobile applications have actually employed cloud computing to provide enhanced services. Cloud computing and mobile applications have been the leading technology trends in recent years.

Energy efficiency has always been critical for mobile devices and the importance seems to be increasing. Use cases are developing towards always on-line connectivity, high speed wireless communication, high definition multimedia, and rich user interaction. Development of battery technology has not been able to match the power requirements of the increasing resource demand [13]. The amount of energy that can be stored in a battery is limited and is growing only 5% annually [4]. Bigger batteries resulting into larger devices are not an attractive option. Also thermal considerations limit the power budged of the small devices without active cooling to about three watts [5]. Energy efficiency improvements can also always be traded for other benefits like device size, cost and R&D efficiency. Indeed, large part of the hardware technology benefits have been traded for programmability in mobile phone designs [6].

We illustrate architecture of the cloud-assisted mobile application platform in Figure 1. Shows the system architecture Energy monitor module measures the amount of mobile energy consumed by an application to execute on mobile or on cloud and the mobile components that will check for energy consumption are CPU, LCD, Wi-Fi and GPS. The offload decider decides based on the optimal application execution policy. The applications run on cloud make use of virtualization of resources thus this saves the execution cost and also saves the mobile energy as the execution done outside the mobile device and only the result appear in mobile. Communication between cloud and mobile should be secured and can be achieved by using encryption and decryption [17].

2. RELATED WORK

To extend the battery life of mobile devices computation offloading is necessary show by experiments that significant power can be saved through remote processing for several realistic tasks (up to 50% of battery life) decision making algorithm that learns and adapts its decision based on previous CPU time measurements [7][19][20]. Moreover, some literatures have studied the energy issues of cloud computing. [3] Presents an energy model to analyze whether to offload applications to the cloud, mainly considering computation energy in the mobile device the communication energy for offloading [18]. Demonstrates that workload, data communication patterns and technologies used (i.e., WLAN and 3G) are the main factors that highly affect the energy consumption of mobile applications in cloud computing. This analysis is roughly based on statistical measurements and investigations [14] [15] [16].

Computation offloading has been the topic of a number of studies. However, only a subset of those studies focus on the effect offloading has on the energy consumption of the mobile device. In most cases the focus is on response time and other resource consumption. Large part of the research uses modeling and simulation, like [8], which is an early investigation of offloading work from mobile to a fixed host concluding that under certain conditions 20% energy savings would be possible.

Compiler technology has been studied in, e.g., [10], where a program is partitioned to client and server parts. The client parts are run on a mobile device and the server part is offloaded. The main metrics evaluated are execution speed and energy consumption. Even though the measurements show that significant energy savings are possible, the outcome is shown to be sensitive to program inputs.

Middleware based approach has been studied in, e.g., [6]. The described framework performs resource accounting and uses execution time, energy usage and application fidelity as criteria for deciding between local, remote and hybrid execution.

Compared to these previous efforts, this paper provides the framework of optimal execution of application on mobile execution and cloud execution in order to conserve energy consumption and time.

3. SYSTEM MODELING AND PROBLEM FORMULATION

In this section, we present a model for application execution on the cloud-assisted mobile application platform. First, we define an application model. Following that, we introduce an Energy monitor module for measures the energy consumption for application execution, and one more model is there to take decision for energy-optimal application.

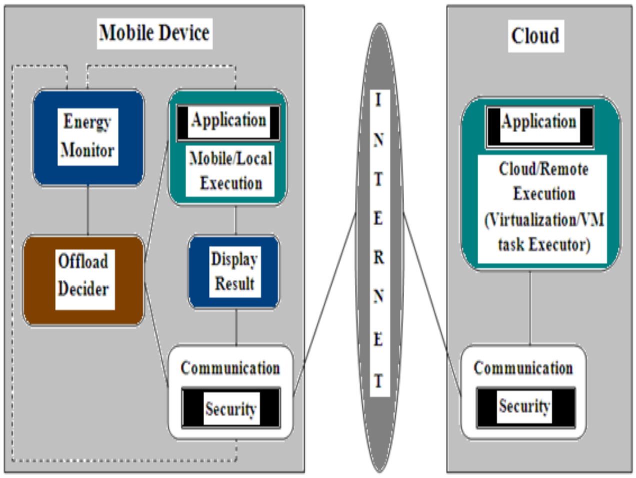

Figure 1. A cloud-assisted mobile application platform: the mobile device is cloned by a energy monitor, offload decider, security which helps in optimal execution of application.

3.1 Application Module

A model that completely depicts all the aspects of an application is complex. Most suitable application for offloading and to get maximum benefit from offloading is an application which is computationally complex and the application input is as low as possible. Both the input data size and the application execution time have the impact on the energy consumption of mobile device. Normally with more data input and (or) high execution time the energy consumption is higher [9].

Image clustering is one of the suitable applications for offloading as it is computationally complex and input is smaller size. Figure 2. Shows execution mode of an application in the proposed system, the image clustering application executes optimally in two alternative modes: the mobile execution and the cloud execution based on the optimal application execution policy which is explained in offload decider module.

Working procedure of Image Clustering with the k-means Algorithm Simple and easy way to classify a given image through a certain number of clusters (assume k clusters) fixed a priori. Discrete cosine transform (DCT) helps to separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality).

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centers.

2. Assign each object to the group that has the closest centers.
3. When all objects have been assigned recalculate the positions of the K centers.
4. Repeat Steps 2 and 3 until the centers no longer move.

Algorithm aims at minimizing an objective function in this case a squared error function. The objective function

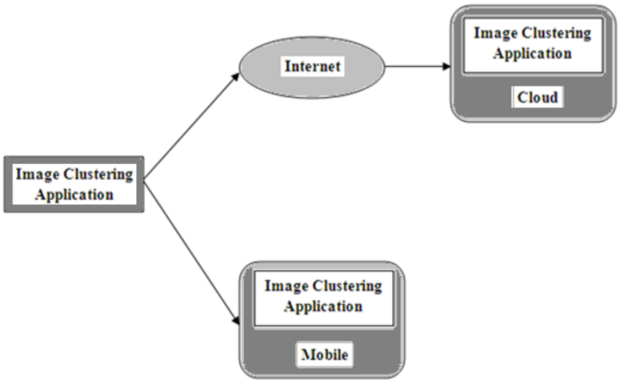

Figure 2. Application Executes in Two Modes

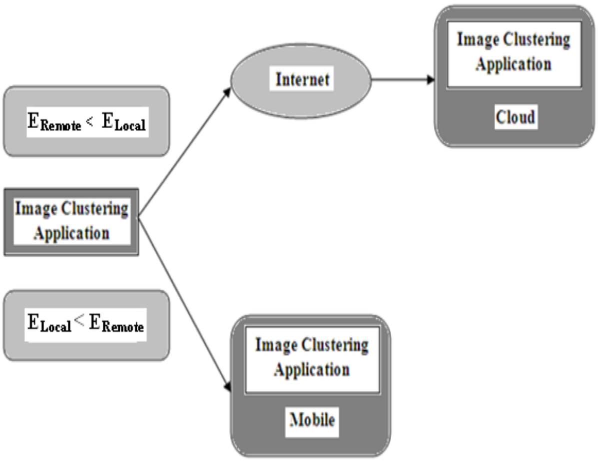

Figure 3. Optimal Execution of Application

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)} - c_j\right\|^2 \quad \text{............ Eq. (1)}$$

Where $\left\|x_i^{(j)} - c_j\right\|^2$ in Eq. (1) is a chosen distance measure between a data point and the cluster centre is an indicator of the distance of the n data points from their respective cluster centre.

3.2 Energy Monitor Module

Energy Monitor app runs on Android devices to measure the energy consumption of the device. Energy Monitor app makes use of open source code of PowerTutor [11] Android app. Energy Monitor app measures the energy consumption by components including CPU, WIFI, 3G and Display. Each component has different states and the power consumption rate at each state differ. Energy Monitor app includes a set of Parameters for each state describing how fast power is consumed at the state.

Energy Consumption for Each Mobile Application or process is considered as a separate user with its own UID. Under /proc/uid_stat/<UID>/ directory lots of information are available about the app or process including data transmitted, memory usage etc. Energy Monitor app map process id to UID then based on the statistics found under the <UID> folder to decide the states for each component. Based on statistics for each component for each UID or app, Energy Monitor app obtained the power consumption for each UID or app [12].

3.3 Offload Decider Module

The decision for energy-optimal application execution is to choose where to execute the application with an objective to minimize the total energy consumed on the mobile device. Based on optimal execution policy offload decider module decides the execution of application.

The two energy values that need to consider for deriving optimal execution policy are below:

$\mathbf{E_{Local}}$ –Performing computation on mobile.

$\mathbf{E_{Remote}}$ –Performing computation on cloud. It is sum of energy for transferring the computational data and $\mathbf{E_{Wait}}$ - the energy consumption while waiting for the result of the offloaded computation E_{Local} and E_{Remote} are mobile energies.

If D is the amount of data to be transferred in bytes and C is the computational requirement for the workload in CPU cycles then

$$E_{Remote} = (D/D_{eff}) + E_{Wait} \text{ ...... Eq. (2)}$$

$$E_{Local} = C/C_{eff} \text{ .... Eq. (3)}$$

Where D_{eff} and C_{eff} are device specific data transfer and mobile computation efficiencies respectively. The D_{eff} parameter is a measure for the amount of data that can be transferred with given energy (in bytes per joule) whereas the C_{eff} parameter is a measure for the amount of computation that can be performed by mobile with given energy (in cycles per joule). After sending the data the mobile needs to poll the network interface while waiting for the result of the offloaded computation this energy consumption by mobile device is E_{Wait}. E_{Remote} is the sum of transfer energy and the waiting energy E_{Wait} as given in the Eq. (2).

3.3.1 Optimal Execution Policy

Optimal execution of an application is as shown in the Figure 3. Offloading to cloud is beneficial when an application execution satisfies Eq. (4) where the sum of energy to transfer the computational data and waiting energy is less than the energy to perform computation locally on mobile. And the mobile execution is beneficial when an application execution satisfies Eq. (5). So based on the energy which is less the offload decider decides where to execute. Based on previous execution history will decide future execution.

$$E_{Remote} < E_{Local} \text{ .... Eq. (4)}$$

$$E_{Local} < E_{Remote} \text{ .... Eq. (5)}$$

To make offloading save energy heavy computation and light communication should be considered [7]. In Eq.(2) assumes that data must be transmitted from the mobile system to the cloud. Fortunately this may not be true in many cases. For example the data (such as photographs and videos) may also reside in clouds with high-speed networks (such as Facebook.com and YouTube.com). Instead of transmitting the data from the mobile system to the cloud the mobile system needs to provide links to the cloud and the cloud may download the data directly from the hosting sites. In this case offloading improves performance of mobile and save large amount of energy.

4. PERFORMANCE EVALUATION

To validate the performance of the proposed system, run the image clustering application for different number of images and on different configuration mobile devices. Mobile A slightly high processing power than the Mobile B. Table 1 give the phone specifications. Figure 4. And Figure 5.shows a comparison of amount of mobile energy consumed by clustering application that run on cloud and mobile for Different number of inputs. Figure 6. and Figure 7. shows a comparison of execution time of clustering application that run on cloud and mobile for different number of inputs.

Table 1 Phone Specifications

Mobile	Name and OS	CPU	RAM
Mobile A	Micromax Canvas 2 and Android 4.0	1.2 GHz Dual Core Cortex-A9	512 MB
Mobile B	Samsung Galaxy S and Android 2.3	1 GHz Cortex-A8	512 MB

Figure 4,5,6,7 show the comparison between the energy consumption and execution time when an image clustering application is executed on mobile and cloud. From the following figures clearly see the difference between various energy values and execution times. For smaller input the execution on mobile might be better but as go on increasing the input size i.e., number of images cloud execution proves to be the better option.

From the above figures conclude that perform high CPU bound tasks on mobile devices with the help of cloud computing. Comparison between the energy consumption and execution time shown in the above graphs clearly states that with the help of developing network technology perform all the CPU intensive jobs on mobile devices and meanwhile Mobile CPU is free to perform other important tasks which are required to be performed. Moreover it can also save battery as most of work is done by cloud and not by the device.

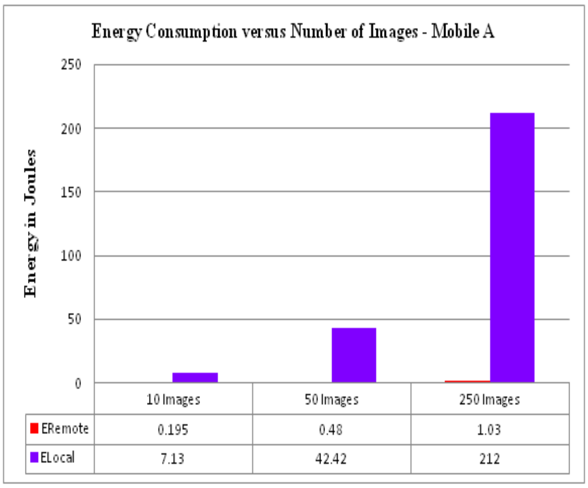

Figure 4. Graph of energy consumed versus number of images - mobile A

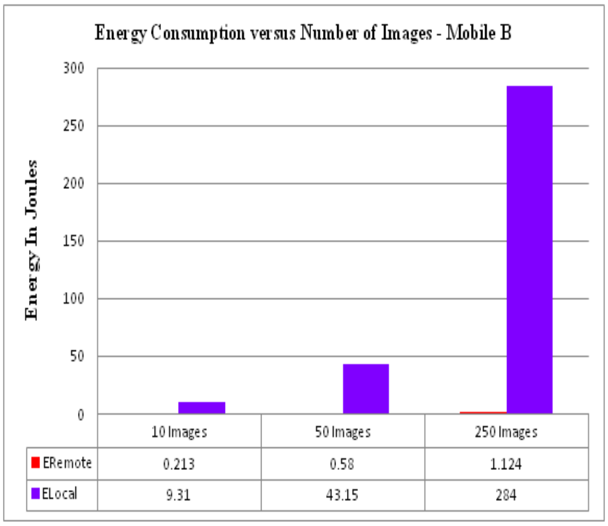

Figure 5. Graph of energy consumed versus number of images - mobile B

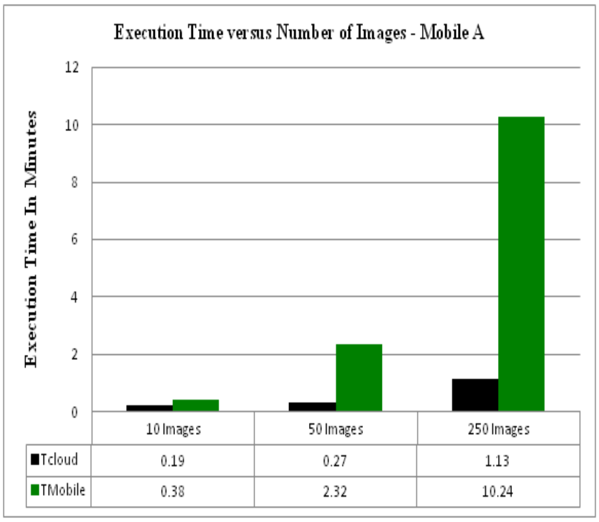

Figure 6. Graph of execution time versus number of images for mobile A.

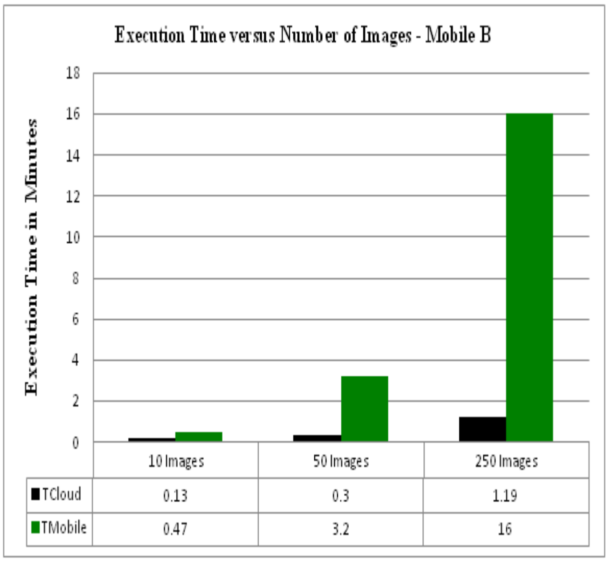

Figure 7. Graph of execution time versus number of images for mobile B

5. CONCLUSION

The convergence of cloud computing and mobile technology is one possible solution to the constraints of mobile devices such as limited computation capability and battery power. Proposed model transfer data and computation from mobile devices into the cloud using native mobile application based on execution policy. The recent execution history of application will take into consideration to take decision for future executions. The mobile devices can be used to save both time and energy when it comes to executing computationally heavy tasks by offloading it to cloud. And also showed the amount of energy saved and execution time is device specific.

This approach can significantly improve the energy saving and time, system efficiency and performances. In this system offloading operations are done manually. With this importance, this article has provided an architecture module description about application, energy monitor and offloads decider execution policy. Then performance evaluations have been discussed.

REFERENCES

[1] M. Satyanarayanan, “Fundamental challenges in mobile computing,” in Proc. 1996 ACM Symp. Principles of Distributed Computing, pp. 1–7.

[2] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in Proc. 2010 USENIX Conference on Hot Topics in Cloud Computing.

[3] K. Kumar and Y. H. Lu, “Cloud computing for mobile users: can offloading computation save energy?” IEEE Computer, vol. 43, no. 4,pp. 51–56, 2010.

[4] Robinson, S. Cellphone Energy Gap: Desperately Seeking Solutions. Tech. rep., 2009. Strategy Analytics.

[5] Neuvo, Y. Cellular phones as embedded systems. Digest of Technical Papers, IEEE Solid-State Circuits Conference (ISSCC) (2004), 32–37.

[6] Flinn, J., Narayanan, D., AND Satyanarayanan, M. Self- tuned remote execution for pervasive computing. In HOTOS ’01: Proceedings of the Eighth Workshop on Hot Topics in Operating Systems (Washington, DC, USA, 2001), IEEE Computer Society, p. 61.

[7] Othman, M., AND Hailes, S. Power conservation strategy for mobile computers using load sharing. SIGMOBILE Mob. Comput. Commun. Rev. 2, 1 (1998), 44–51

[8] Kumar K, Lu YH (2010) Cloud computing for mobile users:can offloading computation save energy? IEEE Comput 43(4):51–56

[9] W. H. Yuan and K. Nahrstedt, “Energy-efficient CPU scheduling for multimedia applications,” *ACM Trans. Computer Systems*, vol. 24, no. 3, pp. 292–331, 2006

[10] Wang, C., AND Li, Z. A computation offloading scheme on handheld devices. J. Parallel Distrib. Comput. 64, 6 (2004), 740–746.

[11] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. Mao and L. Yang,“Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones,” in Proc. Int. Conf. Hardware/Software Codesign and System Synthesis, Oct. 2010.

[12] Y. Chen, S. Q. Zhang, S. G. Xu, and G. Y. Li, “Fundamental trade-offs on green wireless networks,” *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 30–37, 2011.

[13] Cuervo E, Balasubramanian A, Dae-ki C, *et al.* MAUI: making smartphones last longer with code offload, In *Proceedings of the 8th International Conference on Mobile systems, applications, and services*, 2010; 49–62.

[14] M. Othman and S. Hailes, “Power conservation strategy for mobile computers using load sharing,” *Mobile Computing and Commun. Review*, vol. 2, pp. 44–51, 1998.

[15] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H. I. I. Yang, “The case for cyber foraging,” in *Proc. 2002 ACM Special Interest Group on Operating Systems European Workshop*, pp. 87–92.

[16] T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” in IEEE Communications Surveys & Tutorials, pp. 116 - 130, March 2009.

[17] X. W. 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.

[18] P. Rong and M. Pedram, “Extending the lifetime of a network of batterypowered mobile devices by remote processing: a Markovian decisionbased approach,” in *Proc. 2003 Annual Design Automation Conference*, pp. 906–911.

[19] C. Xian, Y. H. Lu, and Z. Y. Li, “Adaptive computation offloading for energy conservation on battery-powered systems,” in *Proc. 2007 International Conference on Parallel and Distributed Systems*, vol. 2, pp. 1–8.

[20] D. Huang, P. Wang, and D. Niyato, “A dynamic offloading algorithm for mobile computing,” *IEEE Trans. Wireless Commun.*, vol. 11, no. 6, pp. 1991–1995, 2012.

[21] Weiwen Zhang, Yonggang Wen, Kyle Guan, Dan Kilper, “Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel” IEEE Trans on wireless communications, vol. 12, no. 9, 2013.