Volume 4, Issue 5, May 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Efficient Resource Utilization of Smartphone's Using Cloud Computing

Abhinandan Jain, Prof. Omprakash W. T.

Department of Computer Engineering,

DYPSOET Lohegaon, Pune –MH, affiliated to

University of Pune-India

Abstract— as Smartphones are becoming more complex and powerful to provide better functionalities, concerns are increasing regarding resource utilization of Smartphones. To efficiently utilize the Smartphone's resources we are using cloud computing there by increases the battery life and performance of the Smartphone's. Consider two computational processes, one by moving the security solution for Smartphones to cloud. Moving the resource-intensive security analyses to the cloud provides powerful protection without exhausting the Smartphones limited resources. Once a security compromise is detected within the emulated environment it instructs its lightweight agent running on the device to take the required actions, e.g. to remove an infected file or to close an attacker's network connection. Two by moving high battery consuming Location based services (LBS) to cloud. LBS include use of GPS, accelerometer, gyroscope, Wi-Fi and Bluetooth etc. By sending the latitude and longitude of the Smartphone's to the cloud, perform the LBS in cloud and send the result to the Smartphone's. This reduces the battery consumption of the Smartphone's and also helps in increasing the performance of the Smartphone's.

Keywords— Include Discrete Cosine Transform (DCT), Global Positioning System (GPS), Location Based Services (LBS), Mobile Cloud Computing (MCC), Software Environment Energy Profile (SEEP).

I. INTRODUCTION

Mobile devices such as Smartphones and tablets have become the device of choice for many users as a result market is growing at unprecedented rates. As modern Smartphones are often used to store user-centric sensitive information like contacts, credentials or to perform financial transactions, they have become an appealing target for cyber criminals [1]. Since Smartphones use software architecture similar to PCs, they are vulnerable to the same classes of security risks. Unfortunately, Smartphones are constrained by their limited resources, battery life, memory and performance that prevent the integration of advanced security monitoring solutions that work with traditional PCs.

The above considerations naturally motivate the need for smarter security solutions, capable of running and performing effectively under an energy-aware perspective. In practice, this requires mechanisms to measure the power consumption. These mechanisms are also challenging to design and develop, because of their conflicting requirements. First, they must not introduce a significant power load themselves. Second, they should gather precise measurements. Third, they must run on the mobile device itself without substantial modifications. For example, hardware-based approaches (e.g., employing a sophisticated sensor attached to the device in a laboratory) are not desirable because they constrain the device to a fixed location, changing completely its usage model (although they guarantee high-precision measurement) [1].

In this paper, we consolidate the aforementioned perspective work by providing the implementation details and, we explain how it is useful. The design is not obtrusive in two ways: first, it does not require any software or hardware modifications on the device; second, it offloads expensive computation to remote servers, keeping only a lightweight client process on the Smartphone.

II. LITERATURE SURVEY

The MPower approach creates an adequate and precise knowledge base of the power "behavior" of several different devices and users, which allows us to create better device-centric power models that considers the main hardware components and how they contributed to the over- all power consumption. In this paper we consolidate our perspective work on MPower by providing the implementation details and evaluation on 278 users and about 22.5 million power-related data. Also, we explain how MPower is useful in those scenarios where low-power, unobtrusive, accurate power modeling is necessary (e.g., green security applications) [1].

Optimizing the energy efficiency of mobile applications can greatly increase user satisfaction. However, developers lack viable techniques for estimating the energy consumption of their applications. This paper proposes a new approach that is both lightweight in terms of its developer requirements and provides fine-grained estimates of energy consumption at the code level. It achieves this using a novel combination of program analysis and per-instruction energy modelling. In evaluation, our approach is able to estimate energy consumption to within 10% of the ground truth for a set of mobile applications from the Google Play store. Additionally, it provides useful and meaningful feedback to developers that help them to understand application energy consumption behaviour [2]. In the survey an author analyses and compares general solutions for energy efficiency on mobile devices at the software level on six major axes: from operating system solutions to energy savings via process migration to the cloud and protocol optimisations. The classifications are Energy

aware operating systems, Energy measurements and power models, Users interaction with applications and computing re- sources, Wireless interfaces and protocol optimisations, Sensors optimisations and Computation off-loading [3].

Andromaly: a framework for detecting malware on Android mobile devices. The proposed framework realizes a Host-based Malware Detection System that continuously monitors various features and events obtained from the mobile device and then applies Machine Learning anomaly detectors to classify the collected data as normal (benign) or abnormal (malicious). Since no malicious applications are yet available for Android, we developed four malicious applications, and evaluated Andromaly's ability to detect new malware based on samples of known malware. We evaluated several combinations of anomaly detection algorithms, feature selection method and the number of top features in order to find the combination that yields the best performance in detecting new malware on Android. Empirical results suggest that the proposed framework is effective in detecting malware on mobile devices in general and on Android in particular [4].

A Survey author discussed about some background notions on mobile technologies, both for wireless telecommunication and networking standards. Describes different types of mobile malware, along with some predictions on future threats, and outlines the differences among security solutions for Smartphone's and traditional Pc's and discusses current threats targeting Smartphone's: first, it analyzes the different methodologies to perform an attack in a mobile environment; then, it investigates how these methodologies can be exploited to reach different goals [5].

III. MOTIVATION

A key challenge in building effective Smartphone security solutions is the resource limitation of Smartphone's. Existing security solutions for Smartphone's consume many resources for their operation, such as memory, storage, CPU, and battery; this compromises their usability and encourages their users to avoid such solutions. Indeed, to be effective, a security solution needs to keep a comprehensive database of malware signatures, requiring a large storage from the phone device. Also, any suspicious behaviour needs to get correlated against a large list of stored signatures, consuming high processing and memory resources from the resource-limited Smartphone devices. Additionally to protect users against zero-day threats, many suggest combining different security solutions, which is not feasible for on-device deployment considering the Smartphone's limited resources. As an example of high resource utilization in Smartphone security solutions, SMobile VirusGuard, a popular antivirus solution for Android phones, takes 40 min and consumes 10% of the battery charge to scan a 200 MB folder on a typical Android phone. This, however, affects the effectiveness and accuracy of such solutions significantly in protecting the devices against security threats [7].

IV. IMPLEMENTATION

The proposed system high level architecture view consists of the client agent and the sever agent. The proposed system is shown in figure 1. The client agent is the android application installed in the android Smartphone's which will register to the server agent. The server agent consists of the services that are used by the client agents.

A. *Client Agent*

The client agent is the lightweight mobile application that performs five tasks:

- Compresses the files that are selected by the user
- Send the compressed files to the server
- Listens to the notifications sent by the server
- Fetch the geo location of the user either using GPS or network provider
- Calculates the energy consumed by the application.

B. *Server Agent*

The server agent is the software application running in the cloud that performs five tasks:

- Receives the compressed files from the client and de-compress them.
- Store the files in the database.
- Perform the security analysis on the files received by the client
- Notify the client if any of the file is virus infected
- Get the user location details and perform the respective computation and send the results to the client.

The user interface of the client application of the proposed system is shown in figure 2.

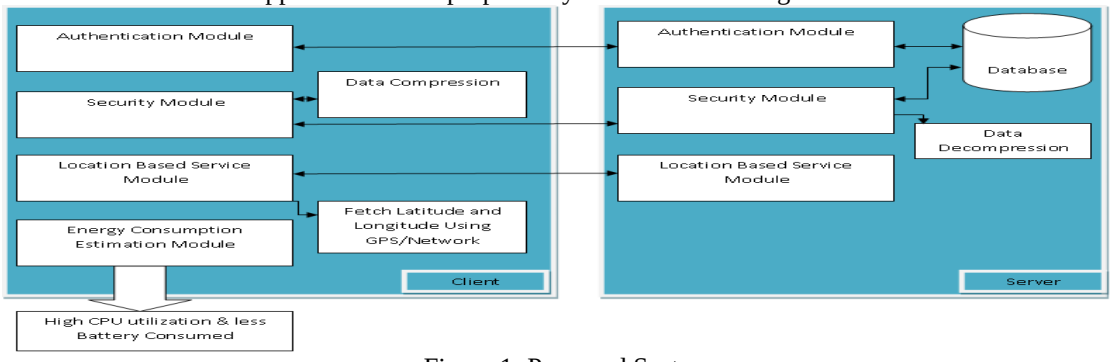

Figure 1: Proposed System.

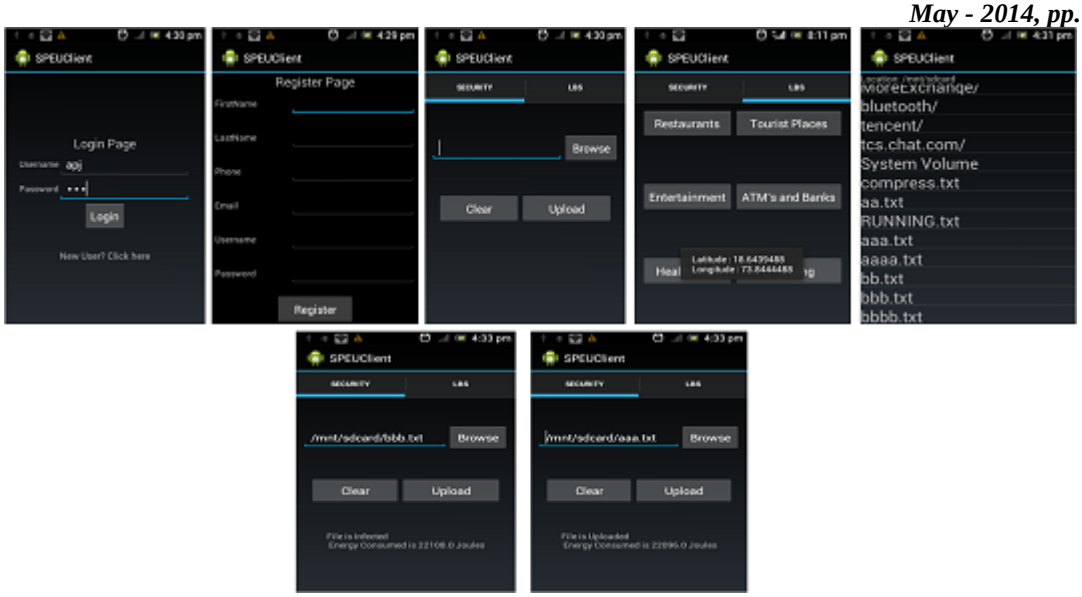

Figure 2: User Interface of the client application.

V. PROPOSED METHODOLOGY

Mathematical Model for the above proposed model is as defined below,

S = system, A = Authentication model, SS = Security model, DC = data compression model, DDC = Data De Compression,

EE = Estimate Energy, DB = Database, LBS = Location Based Services.

S = {A, SS, DC, DDC, EE, DB, LBS}

A = {user_Id, pwd} where user_Id = user name, pwd = password

SS = {f, v} where f = file & v = virus.

DC / DDC = {f, cf} where f = file & cf = compressed file (uses DCT algorithm)

LBS = {lat, long, addr, d} where lat = latitude, long = longitude, addr = address & d = data

EE = {SEEP} where SEEP = software environment energy profile.

SEEP is an algorithm used to find the Energy Consumption by an application and as follows,

Input: H: set of hardware components, l: source line, m: method, and P: path

Output: Energy estimate in Joules

SEEP Algorithm is as follows:

1. **Cost ← 0**
2. **<Θ,M,L> ← regenerate(P)**
3. **D ← propagateDT(Θ)**
4. **for all h ∈ H do**
5. **for all i ∈ Θ do**
6. **f_h ← powerstate(i, h)**
7. **if L(i) = l ∧M(i) = m then**
8. **cost(h)+=C(i, h, f_h, D(i))**
9. **return $\sum_{h \in H}$ cost(h)**

VI. DATA TABLES AND DISCUSSIONS

A. *Virus Data Sets For Security Analysis*

Data sets used for the security analysis will be stored in the server. Once the client agent sends the files for scanning, the server scans the file and checks if file is infected by the any of the viruses defined in the datasets and notifies the user accordingly.

1) *EICAR Standard Anti-Virus Test File:* EICAR test file is a computer file that was developed by the European Institute for Computer Antivirus Research (EICAR) and for Computer Antivirus Research Organisation (CARO), to test the response of computer antivirus programs.

2) *Shutdown virus:* This is the type of virus which on execution it automatically shutdown the system without users permission.

These types of viruses are the data sets used in proposed system for the security analysis. The virus format for the above mentioned viruses are shown in the Table 1.

Table 1: Virus database

SL. NO	Virus Name	Virus Format
1	EICAR standard anti virus	X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR -STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
2	Shutdown antivirus	@echo off shutdown -s -t "30″

VII. EXPERIMENTAL RESULTS

The expected outcome of the proposed work is the energy consumed by the application while performing the computational tasks like security analysis and location based services.

A. *Energy Consumed By Security Analysis*

Energy consumed is estimated from the time when user starts using security analysis module till the server agent sends the results like whether the file is infected or virus free. Table 2 shows the energy consumed the proposed system.

Table 2: Energy estimation for the security analysis module

Sl. No	File Size in Kilo Bytes(kb)	Energy Consumed in Joules(KJ)
1	25	20
2	50	41
3	75	73
4	100	97

B. *CPU Utilization For Security Analysis*

CPU utilization is the CPU time consumed by the application to complete the tasks.

1) *CPU Utilization to Access The File system of Smartphone:* The time utilized for selecting the files from the file system of the Smartphone's is shown in figure 3. In the figure we can see the time utilized for selecting the file is 3.7% (93 ms) of CPU time.

2) CPU Utilization For Sending The File to Server Agent: The time utilized for transferring the selected file from client to server agent using HTTP requests is shown in figure 4. In the figure we can see the time utilized for selecting the file is 23.3% (4170 ms) of CPU time.

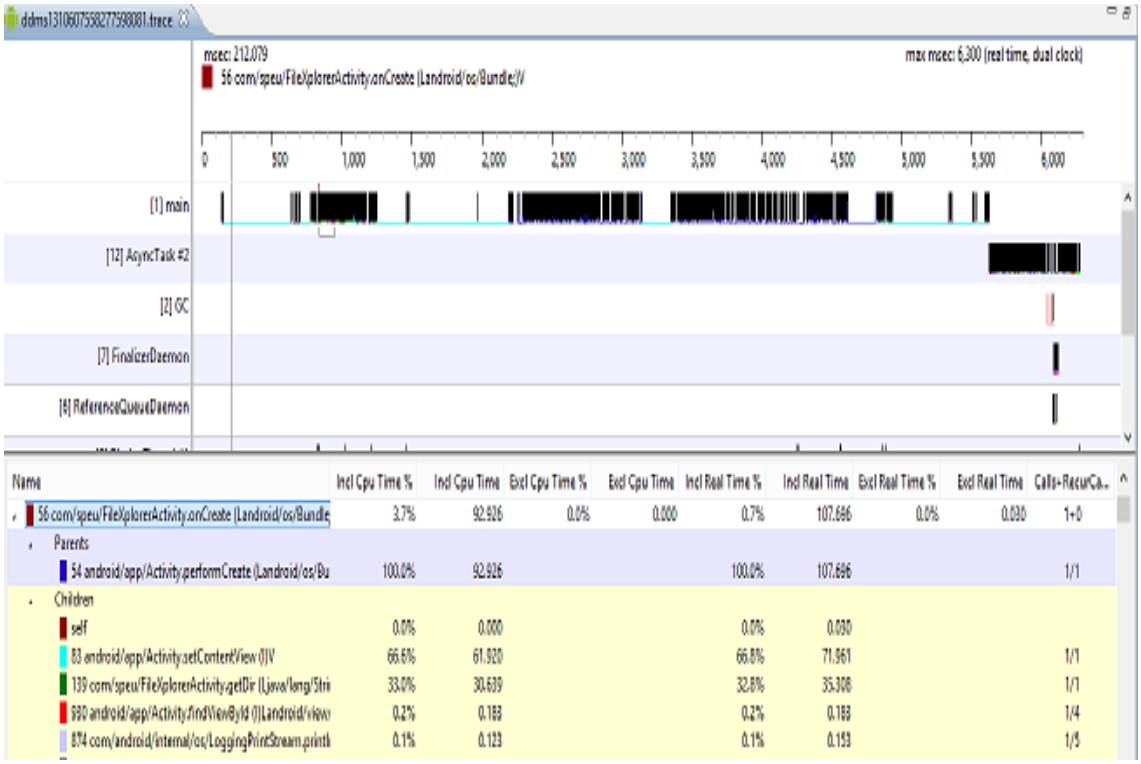

Figure 3: CPU time utilization to explore the file system of Smartphone's

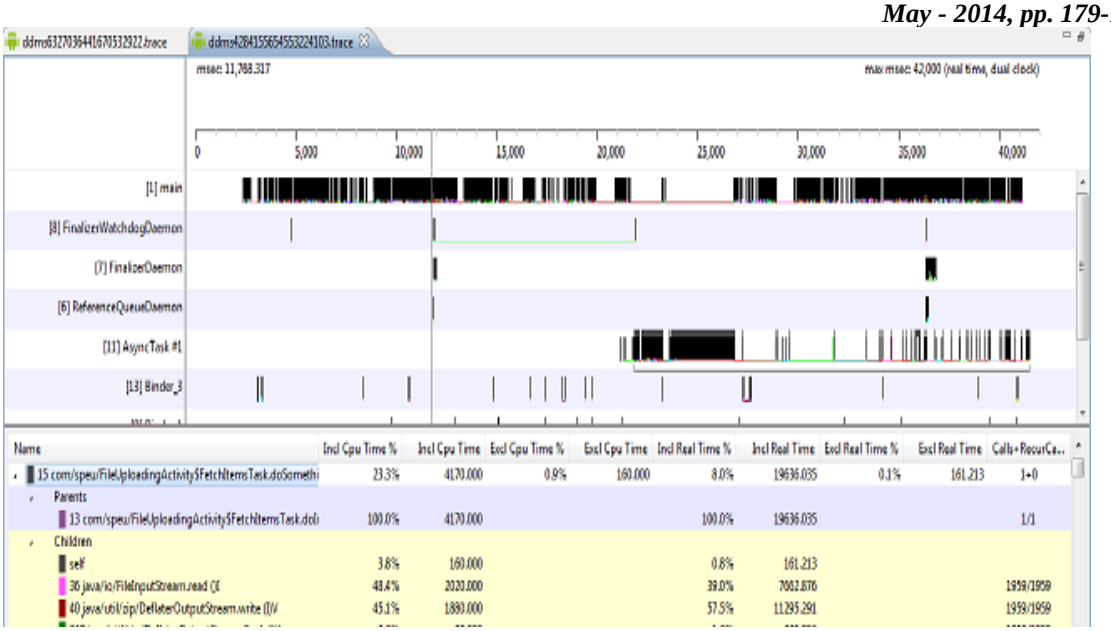

Figure 4: CPU time utilization for transferring the file.

VIII. CONCLUSION

In this paper, we proposed the cloud based services for the efficient resource utilization of the Smartphone's. First the proposed system provides the protection for the Smartphone's with the lightweight mobile application by computing the tasks in the cloud server. This helps in less use of the mobile resources for the security analysis in the smarphone's. Second we are performing the location based services in the cloud server and send the end results to mobile application, this helps in efficient utilization of the mobile resources. In the Future work we can target for the more computational tasks to offload it to cloud and develop lighter mobile client application. And also include the various malware attacks to the Smartphone's.

ACKNOWLEDGEMENT

With immense pleasure, I am presenting this Paper on Efficient Resource Utilization of Smartphone's Using Cloud Computing as part of the curriculum of M.E. Computer Engineering. Inspiration and guidance are invaluable in every aspect of life especially in the field of academics, which I have received from our respected Ms. Arti Mohanpurkar: Head of Computer Department and Mrs. Smita Chaudhari: PG coordinator and Guide: Prof. Omprakash W. T.,

I would also like to thank all my colleagues who have directly or indirectly guided and helped me in the preparation of this paper and also for giving me an unending support right from the stage this idea was conceived.

I also acknowledge the research work done by all the researchers in this field.

REFERENCES

[1] A. A. Nacci F. Trov o F. Maggi M. Ferroni A. Cazzola D. Sciuto M. D. Santambrogio: "Adaptive and Flexible Smartphone Power Modeling", Springer Science+Business Media New York 2013, Mobile Netw Appl (2013)

[2] Shuai Hao, Ding Li, William G. J. Halfond, Ramesh Govindan: \Estimating Mobile Application Energy Consumption using Program Analysis ", 2013 IEEE ICSE 2013, San Francisco, USA 90.

[3] Narseo Vallina-Rodriguez and Jon Crowcroft: \Energy Management Techniques in Modern Mobile Handsets", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 15, NO. 1, FIRST QUARTER 2013.

[4] Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, Yael Weiss: "Andromaly: a behavioral malware detection framework for android devices", Published online: 6 January 2011 Springer Science+Business Media, LLC 2011, J Intell Inf Syst (2012) 38:161190 DOI 10.1007/s10844-010-0148-x.

[5] Junzuo Lai, Robert H. Deng, Chaowen Guan, and Jian Weng: \Attribute Based Encryption With Veriable Outsourced Decryption", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8, NO.8, AUGUST 2013

[6] Mariantonietta La Polla, Fabio Martinelli, and Daniele Sgandurra: "A Survey on Security for Mobile Devices", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 15, NO. 1, FIRST QUARTER 2013.

[7] Saman Zonouz, Amir Houmansadr, Robin Berthier, Nikita Borisov, William Sanders: "Secloud: A cloud-based comprehensive and lightweight security solution for smartphones published in computers & security", security 37 (2013) 215 e227 ScienceDirect.

[8] Angelos Stavrou Jerey Voas, Tom Karygiannis, and Steve Quirolgico:"Building Security into Off-the-Shelf Smartphones", Published in IEEE Computer Society 0018-9162/12/$31.00 2012, IEEE.

[9] Hilarie Orman: "Did You Want Privacy With That? Personal Data Protection in Mobile Devices", MAY/JUNE 2013 1089-7801/13/$31.00 2013 IEEE Published by the IEEE Computer Society

[10] Fangming Liu, Peng Shu, Hai Jin, Linjie Ding, and Jie Yu, Huazhong University Of Science and Technology Di Niu, University Of Aleberta BO LI, The Hong Kong University Of Science and Technology: "Gearing Resource

Poor Mobile Devices With Powerful Clouds: Architecture, Challenges, And Applications", IEEE Wireless Communications June 2013.

[11] Qing Li and Greg Clark, "Mobile Security: A Look Ahead", January/February 2013 Copublished by the IEEE Computer and Reliability Societies.

[12] Yong Wang, Kevin Streff, and Sonell Raman: "Smartphone Security Challenges." Published by the IEEE Computer Society 0018-9162/12/$31.00 2012 IEEE.

[13] Ahmad Rahmati, Member, IEEE, and Lin Zhong, Member, IEEE: "Studying Smartphone Usage: Lessons from a Four-Month Field Study", IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 12, NO. 7, JULY 2013

[14] www.developer.android android developer site

