

# CONTEXT-AWARE DECISION MAKING SYSTEM FOR MOBILE CLOUD OFFLOADING

Mustafa TANRIVERDİ<sup>1</sup> and M. Ali AKCAYOL<sup>2</sup>

<sup>1</sup>Institute of Information, Gazi University, Ankara, Turkey

<sup>2</sup>Department of Computer Engineering, Gazi University, Ankara, Turkey

## ABSTRACT

*In this study, a mobile cloud offloading system has been developed to decide that a process run on the cloud or on the mobile platform. A context-aware decision algorithm has been developed. The low performance and problem of battery consumption of mobile devices have been fundamental challenges on the mobile computing. To overcome this kind of challenges, recent advances towards mobile cloud computing propose a selective mobile-to-cloud offloading service by moving a mobile application from a slow mobile device to a fast server in the cloud during run time. Determine whether a process running on cloud or not is an important issue. Power consumption and time limits are vitally important for decision. In this study we used PowerTutor application which is a dynamic power measurement modelling tool. Another important factor is the process completion time. Calculate the power consumption is very difficult.*

## KEYWORDS

*Mobile Cloud Computing, Cloud Computing, Mobile Cloud Offloading, Decision Making System*

## 1.INTRODUCTION

Recently, smartphones have shown rapid development. Emailing and Web browsing via smartphones have gain popularity for users [1]

The number and variety of mobile applications has been rapidly increased in recent years. Smartphones have become more powerful and more popular thanks to these applications. In some cases smartphones may be inadequate while operating applications using rich resources. Mobile cloud computing is emerging as the best solution to overcome such challenges.

Cloud computing is not only related to personal computers, but also affects the mobile technology using cloud sources in mobile application [2]. Mobile cloud computing can be defined running of cloud computing services and applications on mobile devices [3]. According to the research from Juniper, the cloud computing based mobile software and application are expected to rise 88% annually from 2009 to 2014, and such growth may create US 9.5 billion dollars in 2014.

The main topic of mobile cloud computing solution is to migrate computationally-intensive methods to the cloud servers over the wireless networks in order to reduce the power consumption or the response time. This refers to as mobile cloud offloading [4] [5] [6]. Mobile cloud offloading allows the mobile applications to achieve better user-experience, and offers the cloud service providers more business opportunities [7]. As a result mobile cloud offloading offers huge advantages to application developers and firms.

This paper makes the following contributions;

- We integrated open source PowerTutor application on our Decision Making System for dynamic power measurement.
- An algorithm is developed to estimate the amount of power required for running processes taking into consideration AsyncTask issue.
- We design, implement a decision algorithm, which uses the previously collected data determine whether to offload each process to the cloud servers.
- The experimental results evaluated on the basis of platform, connection type and process.

The rest of the paper is organized as follows: Section 2 presents related works. Developed system and algorithm are given in Section 3. Details of implementation are given in Section 4. Section 5 shows the results. Finally, Section 6 presents conclusions.

## 2. RELATED WORKS

When the literature is examined it is determined that limited studies and similar studies in this field has increased dramatically in recent years. These studies may be classified as collecting data over the mobile environment, decision to transfer a process to the cloud and transferring a process to the cloud as divided parts. In these studies special algorithms were developed to decision making stages for mobile cloud offloading. In Addition the machine learning algorithms have been used in several studies. The majority of the studies context sensitive. Studies have focused on power consumption, process runtime, internet connection status, location, and time contexts. Most of these studies have been implemented in simulation or experimental environments.

Android operation system requires using of AsyncTask class for complex processes. In addition the processes involving internet connection must be operated in AsyncTask class [8]. So execution of a complex process is spread over a long period and using less resources are provided. Developers must use this class otherwise their applications will be crashed. Most of applications in this field are developed for android operation system. But none of these studies provide information about AsyncTask. Using AsyncTask class makes very difficult the calculation of power consumptions. This study is unique in this area in the context of using AsyncTask class.

CloneCloud [4] and MAUI [5] solve integer linear programming problems with their decision engine, while ThinkAir [9] calculates the averages of historical execution costs on the cloud server and on the mobile device, and chooses the better one. Namboodiri and Ghose [10] also propose an algorithm to determine whether running an application in the cloud is more power efficient. Wolski et al. [11] implement several bandwidth measurement methodologies and compares their performance in grid computing offloading. None of the works [4] [5] [10] [11] makes the offloading decisions based on rich contexts. Several important studies in this area are summarized below.

Xie et al. [12] have developed an application called phone2cloud for process transferring from mobile phone to the cloud environment. The purpose of this application is determined to reduce the power consumption and improve the performance. According to the results of this study each process type reveals different performance values in mobile or cloud environments. For example number sorting process is a suitable process for mobile platforms. GPS position determining process shows better performance in cloud platforms. Counting word process shows similar performance on mobile and cloud environment. A user who uses phone2cloud application can achieve better efficiency by transferring suitable process to cloud platform.

Wu et al. [13] have researched on mobile cloud offloading. The aim of the study is to transfer process from mobile to cloud platform for high performance and short runtime. On the basis, the work will be transferred to the cloud platform must be evaluated in terms of performance and power states. Two formula were developed, first is time based and the second is power based.

The variables used in formulas are shown in below.

- P<sub>m</sub> : current power
- P<sub>i</sub> : execution power for process
- P<sub>tr</sub> : power for sending and receiving data
- B : internet bandwidth
- D : exchanged data
- T<sub>m</sub> : execution time on the mobile device
- T<sub>s</sub> : execution time on the server

$$T_m > T_s + \frac{D}{B}$$

$$P_m * T_m > P_i * T_s + P_{tr} * \frac{D}{B}$$

When it is determined that the cloud platform is advantageous compared to mobile platform via formulas, related process will be executed in cloud platform. According the researchers, in order to take the decision about transferring a process to the cloud, internet bandwidth value is very important. In the high Internet bandwidth environments processes are usually transferred to the cloud platform. In addition when the phone charge level is low the results of first formula are taken into consideration. Or when the process requires high performance the results of second formula are taken into consideration.

Lin et al. [14] developed a decision engine named “CADA” for mobile cloud offloading. Researchers designed and evaluated CADA, which consists of four components: context profiler, context-aware decision algorithm, power model, and context database. The context profiler is a background service that collects various contexts required for making the offloading decisions like signal strength transmission time, time-of-day, location. Time-of-day and location contexts are used in decision algorithm. This means that for a given time-of-day, a user is likely to visit the same place, have the same wireless network condition, and run the same applications. CADA algorithm learns from the past execution records and can give the best decisions. Users profile information and operated process records are saved in database. CADA is installed in four mobile phones for three days. Experiment results are examined for reducing the processing times and power consumption. 80% efficiency has been achieved through accurate predictions.

Lim and Lee [15] studied about prediction of the power for offloading a process to cloud. Predicting the correct amount of consumed power is reported to be critical in the mobile cloud offloading. Researchers designed a power model for predicting power consumption. This model is compared with linear regression, decision table, local weighted learning, nearest neighbors and kstar models on video processing, chess and face detect processes. According to the implementation results designed model, kstar and linear regression models have achieved the best results.

## 2.1. MEASUREMENT ENERGY CONSUMPTION ON MOBILE DEVICES

Instant power consumption of mobile devices can be measured in two ways. The first one is reaching the correct value by using additional hardware and the second one is reaching approximate values with a power modelling tool which access the device's system data. First method can measure the accurate values but this method is applicable only for experimental usage. Achieving power consumption values on mobile devices is also possible by only software without using a hardware. According to the success of the power model and software, it is possible to reach the closest results in second method. It is determined that PowerTutor applications is used in similar studies were examined [16] [17] [18] [19].

PowerTutor is an application for Android devices that displays the amount of power consumed by major system components such as CPU, network interface, display, GPS receiver and other applications [20]. PowerTutor builds a model of power consumption by direct measurements. The model provides an estimation of power consumption with an estimate error of 5% [16].

## 2.2. ASYNCTASK AND RUNNING IN BACKGROUND

The process working with AsyncTask class can be run in background without affecting the main flow related business processes such as screen touching, phone calls. In android operating system complicated process must be run in AsyncTask class otherwise the related application will be crashed.

In this research complex processes are determined and these processes are operated with AsyncTask class. When a complex process is operated without AsyncTask this process use large amount of mobile devices resources and ends up in a short time, which is run by AsyncTask class this process use fewer resources and ends up in longer times. Due to this situation measurement the performance of a process becomes more difficult.

## 3. MOBILE CLOUD OFFLOADING SYSTEM

The system consist of two applications. First one is a mobile application works on mobile devices which have android operating system. Second one is a web application works on cloud environment. The general structure of the system and the relationships of these two applications is shown in Figure 1.

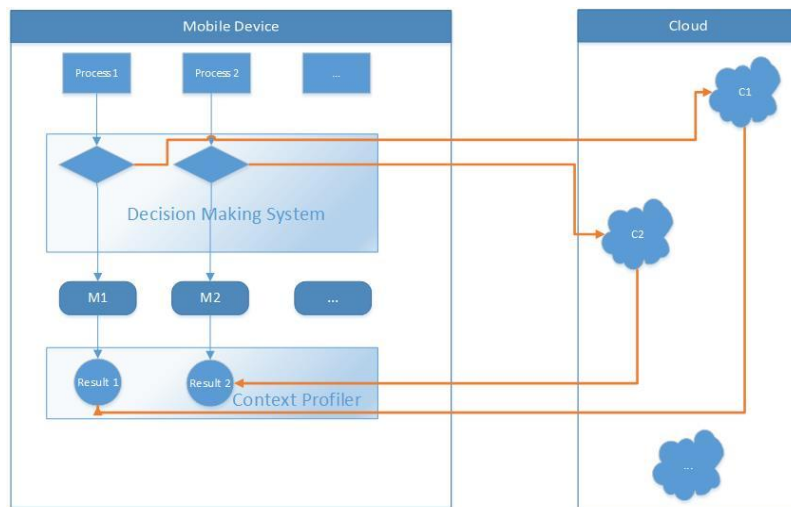


Figure 1. Overview of the system

### 3.1. MOBILE APPLICATION

Mobile application was developed with Android SDK (Software Development Kit) on Eclipse application development environment [21] [22]. SQLite was used as the database [23]. PowerTutor application was configured and integrated on the application. All processes were executed with AsyncTask class.

The application runs in the background and has no user interaction. In every 10 minutes a process is selected randomly from among priorly defined processes. Whether to select mobile or cloud platform is determined by decision algorithm. Then selected process is executed on determined platform and then performance value of process is saved.

#### 3.1.1. PROCESS TYPES

Complex process types have been selected for mobile cloud offloading system. These process types are Eight Queen Problem, Character Counting in Text File, Face Detection and Optical Character Recognition (OCR). OpenCV library which widely used in the literature and used commercially selected for face detection process type [24]. Tesseract Library selected for OCR process type [25]. Three processes are defined depending on difficulty for each process type. For example three text file created for character counting process and their size are given respectively 70 KB, 150 KB and 300 KB. So that 12 kind of process have created.

#### 3.1.2. CONTEXT PROFILER MODULE

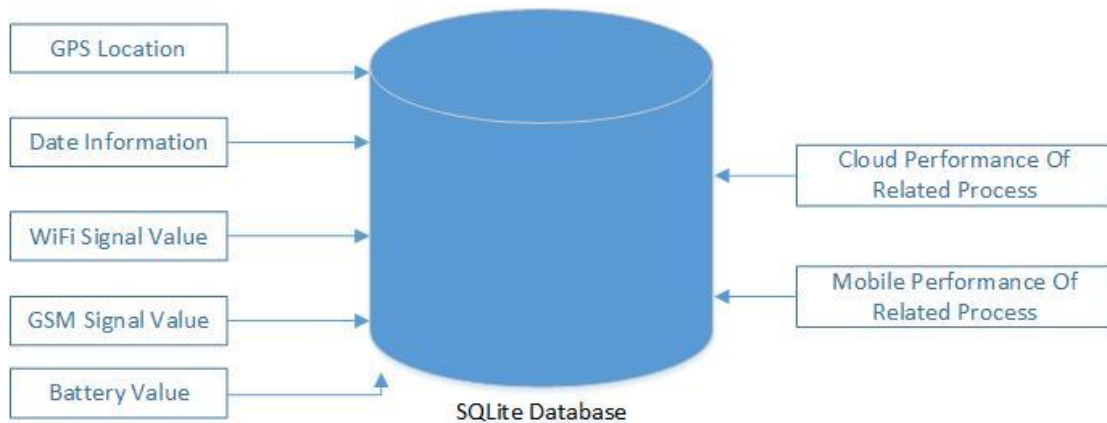


Figure 2. Overview of the Context Profiler Module

Location data, Wi-Fi and GSM (Global System for Mobile) connection values, and charging status information of mobile device are saved to database for specific time intervals during the day by Context Profiler Module. In addition when a process completed, power consumption and elapsed time information of related process are recorded. Thanks to context profiler module the profile of mobile device user and platform based performance history of each process are obtained.

### 3.1.3. Decision Making Module

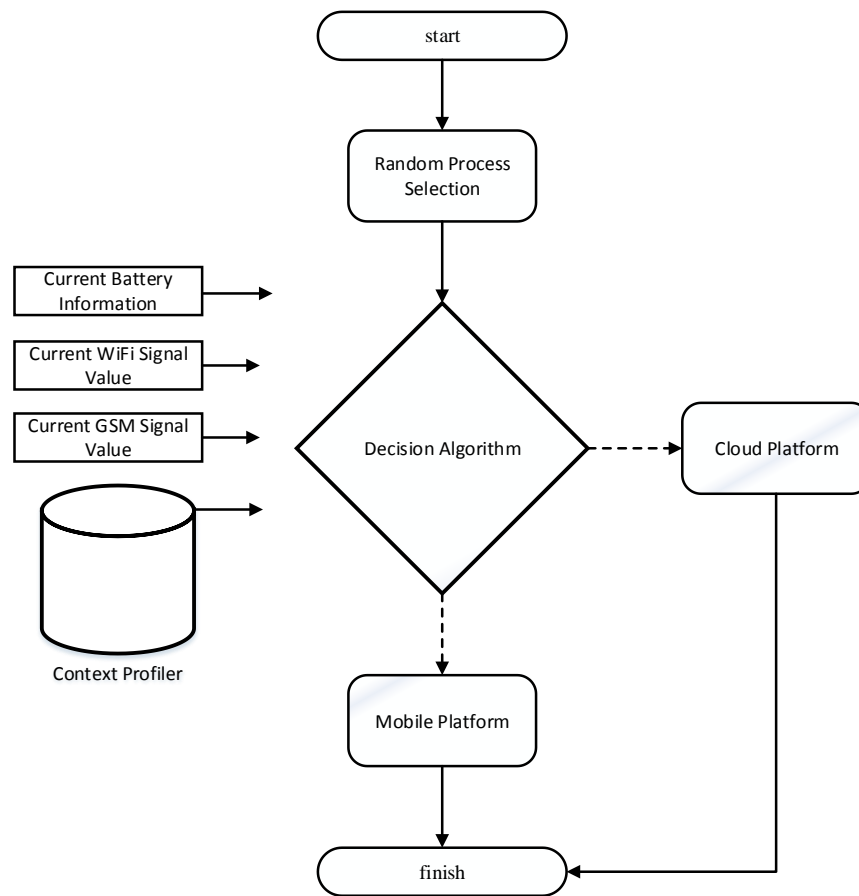


Figure 3. Overview of the Decision Making Module

Overview of the decision making module is shown in Figure 3. The information provided by context profiler module is very important for this module. In addition to profile information instant battery level, Wi-Fi and GSM internet connection values and location data are used in decision making algorithm. The aim of this algorithm is giving the best decision about mobile cloud offloading. At this stage, a non-resolvable by conventional methods NP (Non-deterministic Polynomial Time) problem emerges.

### 3.2. CLOUD APPLICATION

Cloud application is a web project created by using Java programming language. This application has been developed to respond to requests from mobile application. For the comparison of performance completely same library and codes have been used in mobile and cloud applications. This application basically consists of two parts. First part is, responding to web service requests from mobile applications. In this part for web services technology rest service structure is used which is more efficient and faster. The second part is saving mobile applications data to central database. Mysql database is created for performance data of process operated on mobile applications and event logs.

### 3.3 DATABASE STRUCTURE AND DATA

The database structure and data of application are shown below. Constant data are shown in Table 1. Twelve processes created for application are listed in Table 2. Performance date of each process is saved to Table 3. Context profile data of mobile device user is saved to Table 4.

Table 1. Constant Data

Platforms		ConnectionTypes		Process Types	
id	name	id	name	id	name
1	mobile	1	No connection	1	Eight Queens
2	cloud	2	Wi-Fi	2	Character Count
		3	GSM	3	OCR
				4	Face Detection

Table 2. Process Data

id	type_id	level	value
1	1	1	8
2	1	2	9
3	1	3	10
4	2	1	log1.txt
5	2	2	log2.txt
6	2	3	log3.txt
7	3	1	ocr1.png
8	3	2	ocr2.png
9	3	3	ocr3.png
10	4	1	pic1
11	4	2	pic2
12	4	3	pic3

Table 3 Historical Data of Operated Process

id	process_id	time	power	loc_lat	loc_lon	platform	con_type	con_level	date
1	1	6	0	39.93..	32.82..	1	0	0	..
2	3	150	150	39.93..	32.82..	2	1	45	..
3	8	50	80	39.93..	32.82..	1	0	0	..
4	12	480	152	39.93..	32.82..	2	2	12	..
5	4	250	450	39.93..	32.82..	2	1	90	..
6	10	300	280	39.93..	32.82..	2	2	40	..
7	6	40	80	39.93..	32.82..	1	0	0	..
..	..	..	..	..	..	..	..	..	..

Table 4. Context Profile Data

id	loc_lat	loc_long	con_type	Wi-Fi_level	GSM_level	date
1	39.93..	32.82..	1	50	0	..
2	39.93..	32.82..	2	55	12	..
3	39.93..	32.82..	2	25	0	..
4	39.93..	32.82..	1	0	25	..
..	..	..	..	..	..	..

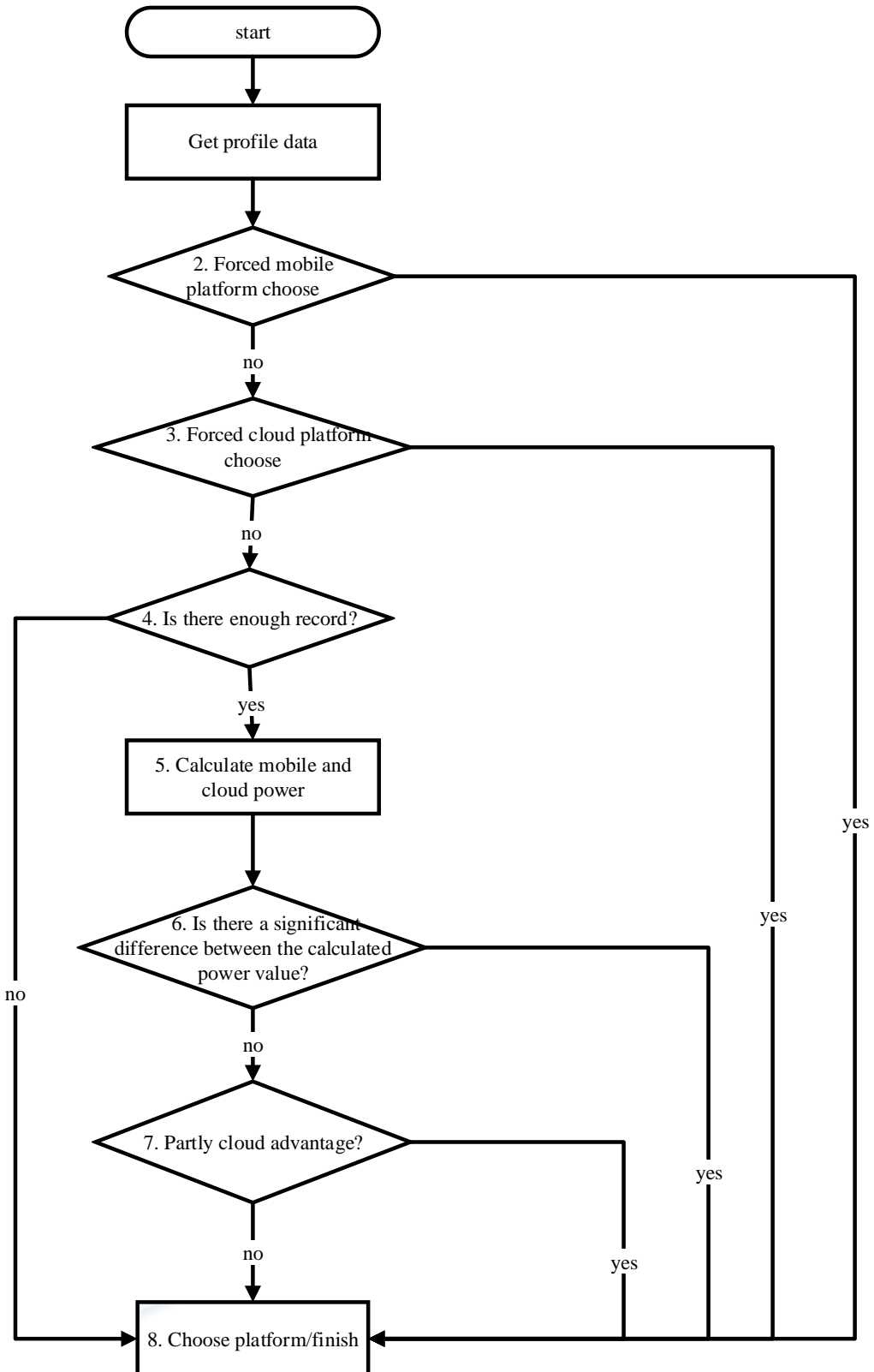


Figure 4. Flowchart of Decision Algorithm

Each section is located in the algorithm are described below.



### 3.4 DECISION ALGORITHM

The structure of the decision algorithm is shown in Figure 4. Some constant variables have been created for using in algorithm. These variables are given below.

*Critical battery level: 20%*

*Perfect Wi-Fi connection level: 45%*

*Perfect GSM connection level: 25%*

*Sufficient Wi-Fi connection level: 45%*

*Sufficient GSM connection level: 25%*

*Maximum records number of relevant process for the algorithm: 5*

*Power calculation coefficient: 1*

*Significant advantages coefficient for the environment comparison: 2*

*Partly advantages coefficient for the environment comparison: 1*

#### 3.4.1. GET PROFILE DATA

In this section, context information such as the charge level of the mobile device, the plugging condition of mobile device, the GPS location data of the mobile device and current Internet connection type and level are obtained to be used in the following sections. Besides this information determining the location where user have sufficient cloud transfer conditions is important point for algorithm. Additionally accessed information is configured to use in algorithm. For example the internet connection level classified as *perfect*, *sufficient* and *not enough*, cloud server status is defined as *available* or *not available* and the user's appropriate locations for cloud transfer is converted as an array.

#### 3.4.2. FORCED MOBILE PLATFORM CHOOSE

If there is no Internet connection or the cloud server's status is defined as not available, the algorithm gives mobile platform selection decision without considering other stages.

#### 3.4.3. FORCED CLOUD PLATFORM CHOOSE

If the Internet connection is perfect and the mobile device is plugged in and the occupancy rate of battery is above the critical level, the algorithm gives cloud platform selection decision without considering the other stages.

#### 3.4.4. IS THERE ENOUGH RECORD

In the early days the number of records may not be enough for the algorithm. In this case if the Internet connection is perfect, the occupancy rate of battery is above the critical level and user and user's location is among the best locations algorithm gives cloud platform selection decision. In this section usually mobile platform is chosen.

#### 3.4.5. CALCULATE MOBILE AND CLOUD POWER

Complex processes are chosen for application and these processes are operated on AsyncTask class. For example, a process that can be completed in 10 seconds with all resources of mobile device are finished in 100 seconds in the background while using fewer resources spread over time.

After a process is operated its performance data is saved as in Table 3. To determinate the power consumed by a process, looking into start and end power values would not be enough. In this stage run time of a process plays an important role. The run time value and difference in power values must be used together in an algorithm to calculate accurate consumed power. Variables and formulas of developed algorithm are given below.

$M_t$ : average completion time for related process on mobile environment

$M_p$ : consumed power for related process on mobile environment

$C_t$ : average completion time in similar location and internet connection conditions for related process on cloud environment

$C_p$ : average consumed power in similar location and internet connection conditions for related process on cloud environment

$K$ : comparison coefficient

$P_m$ : The average value of the 10 highest consumed power for related process on mobile environment

$P_c$ : The average value of the 10 highest consumed power for related process on cloud environment

$X_m$ : calculated power for related process on mobile platform

$X_c$ : calculated power for related process on mobile platform

$$X_c = (1 + K * \frac{C_p}{P_c}) * C_t$$

$$X_m = (1 + K * \frac{M_p}{P_m}) * M_t$$

#### 3.4.6. IS THERE A SIGNIFICANT DIFFERENCE BETWEEN THE CALCULATED POWER VALUES?

Calculated mobile and cloud power values are obtained as  $X_m$  and  $X_c$  at previous step. *Significant advantages coefficient for the environment comparison* is predetermined for algorithm. If there is a significant power value, advantageous platform will be selected.

#### 3.4.7. PARTLY CLOUD ADVANTAGE?

*Partly advantages coefficient for the environment comparison* is predetermined for this section. If partly cloud advantage is detected and cloud transfer conditions is perfect, cloud platform is selected by algorithm in this section.

### 4. EXPERIMENTAL IMPLEMENTATION

The developed application has implemented as an experiment. The application is installed on one mobile phone and implemented for two weeks. Model of the test phone is General Mobile Discovery 2. Its processor has eight-core and 2GHz speed. It has 2 GB RAM and its operation system is Android 4.4.2 KitKat. The operation system of cloud server is Windows Server 2012 64 bit. It has Intel i5-4590 3.30 GHz processor and it has 4 GB RAM.

Installed application was operated in background and had no user interaction. In every ten minutes a randomly selected process was operated. Problems encountered during the test have been detected and fixed. Additionally the test results and abnormal data are analyzed and algorithm improvements have been made.

Installation file of the application is presented on the internet. When the appropriate conditions are met (excellent Wi-Fi connection, plugged in charging status) on mobile devices during the implementation, log and performance data are transferred to the database in the cloud environment. So installation and the transferring log and performance data to cloud servers are done entirely online. In addition the log records can be tracked in real time and necessary improvements and interventions can be made.

## 5. EXPERIMENTAL RESULTS

The results of experimental implementation are described below.

### 5.1. EVALUATION OF THE DECISION GIVEN BY ALGORITHM

Log records of algorithm are saved to database for evaluation of the decision given by algorithm. Some of the saved records are shown in Table 5. Operated process, Wi-Fi and GSM internet levels, whether relevant location is between the best locations, comparison value and date information are shown on this table. So it is possible to evaluate the algorithm.

Table 5. Algorithm Log Records

id	p_id	Wi-Fi	GSM	best_loc	value	platform	date
1	1	45	15	false	0.215	2	04.5.2015 10:07
25	12	30	8	true	0.825	2	04.5.2015 19:00
82	3	0	15	false	null	1	05.5.2015 14:21
150	5	30	10	false	1.152	1	06.5.2015 08:02
315	9	80	25	true	0.586	2	07.5.2015 21:30
..	..	..	..	..	..	..	..

### 5.2. COMPARISON OF MOBILE AND CLOUD ENVIRONMENTS

Historical data of operated processes are shown in Table 3. The average power and time information used for the execution of each job on the basis of the platform are obtained with these records. With these information, the number of transfers in the cloud and the performance of mobile and cloud environments can be accessed.

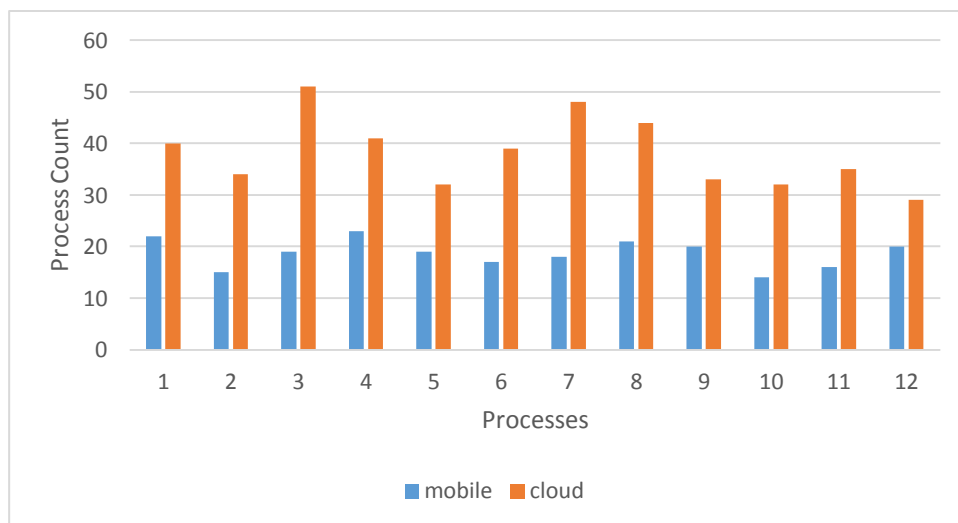


Figure 5. Platform Based Process Counts

Platform Based Process Counts are shown in Figure 5. According to the figure, for all processes count of cloud platform is higher than mobile platform.

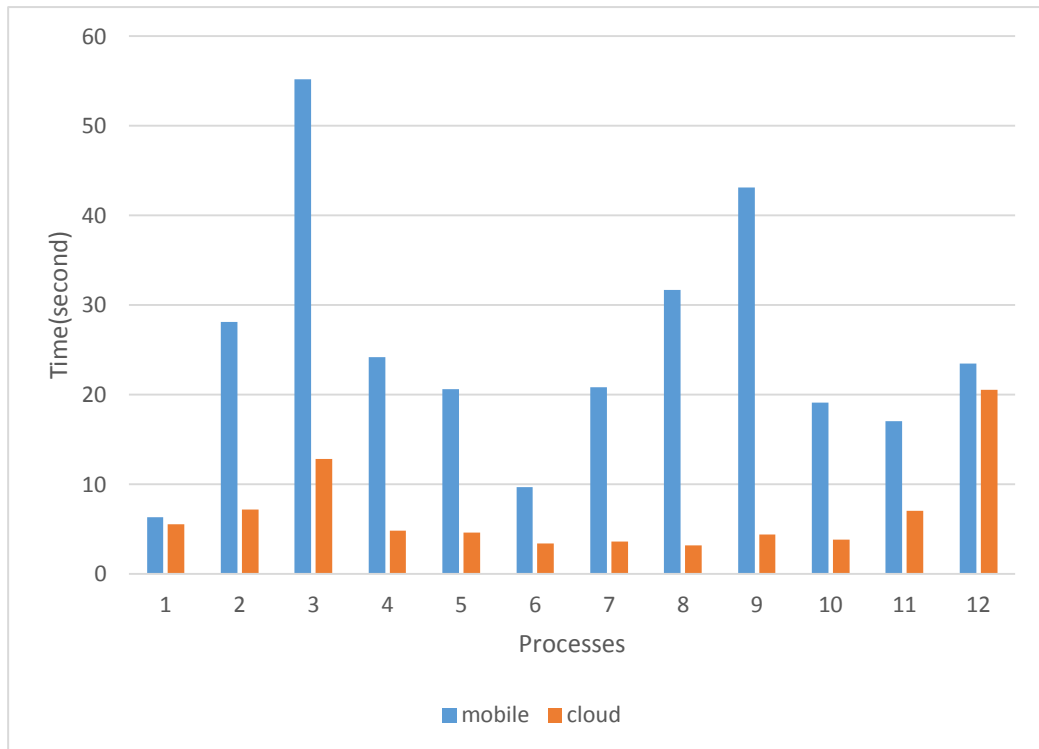


Figure 6. Platform Based Time Values of Processes

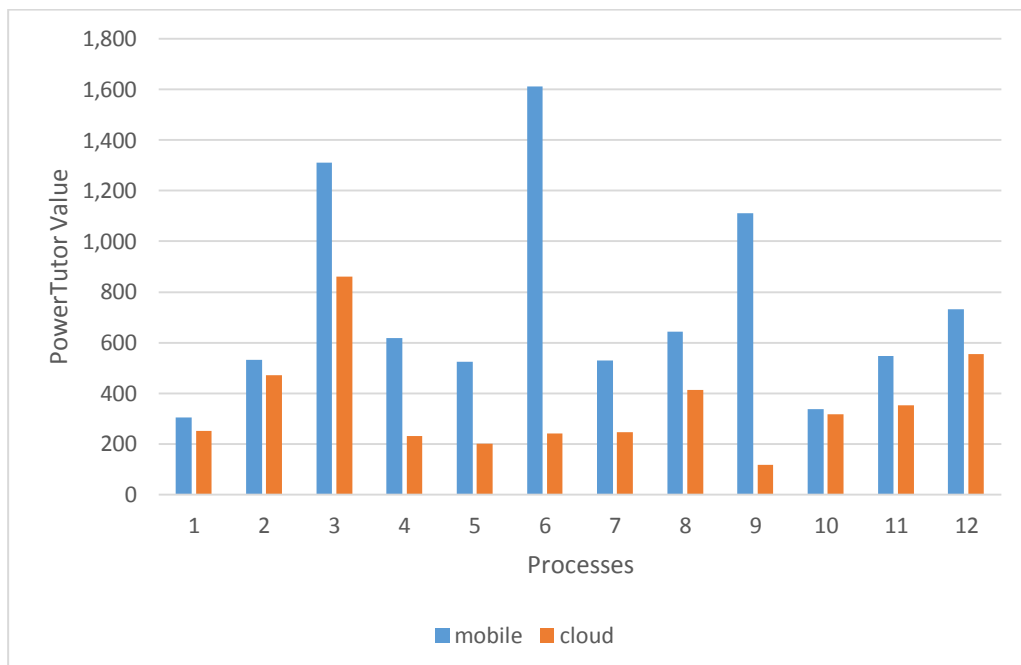


Figure 7. Platform Based Power Values of Processes

Platform based average power and time values of processes are shown in Figure 6 and Figure 7. According to the figures average power and time values of mobile platform are usually higher than cloud values. As a result cloud offloading rate is increasing while mobile platform performance is decreasing.

### 5.3. PERFORMANCE COMPARISON OF GSM AND WI-FI INTERNET CONNECTIONS

Count of cloud transferred processes are grouped by internet connection type and shown in Figure 8. Compared to the GSM connection type Wi-Fi connection type has more records. The mobile phone user who participated in experimental implementation usually found in places such as home and office which has access to Wi-Fi throughout the experiment and this had a major influence in the emergence of such a result.

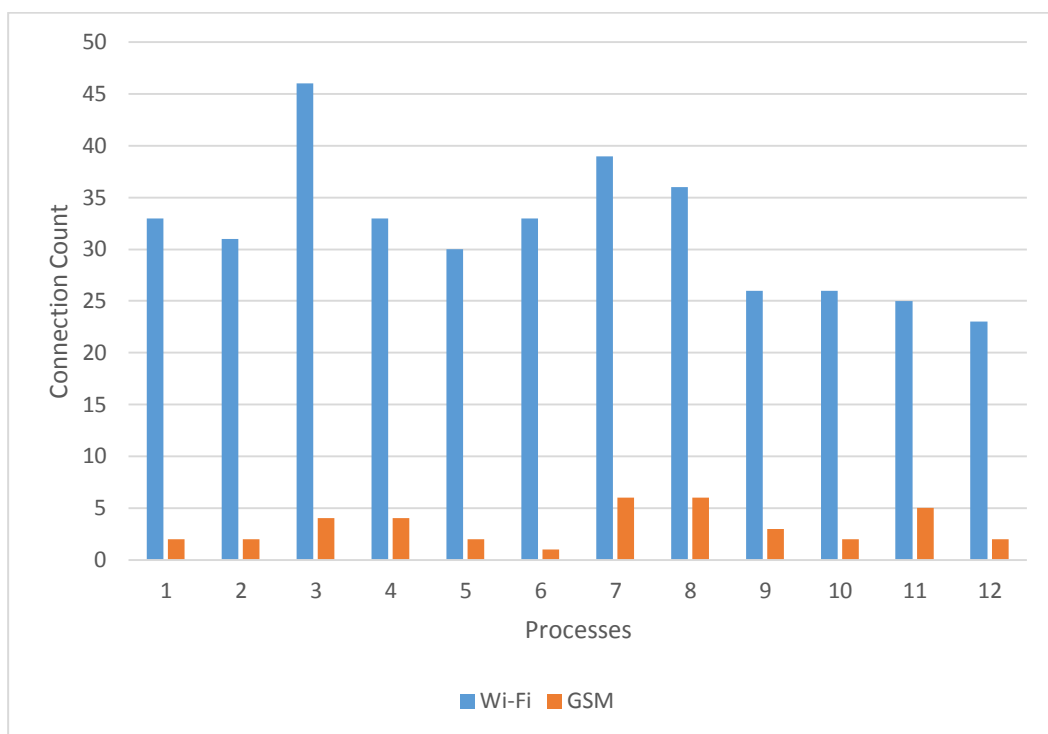


Figure 8. Connection Type Based Count of Processes

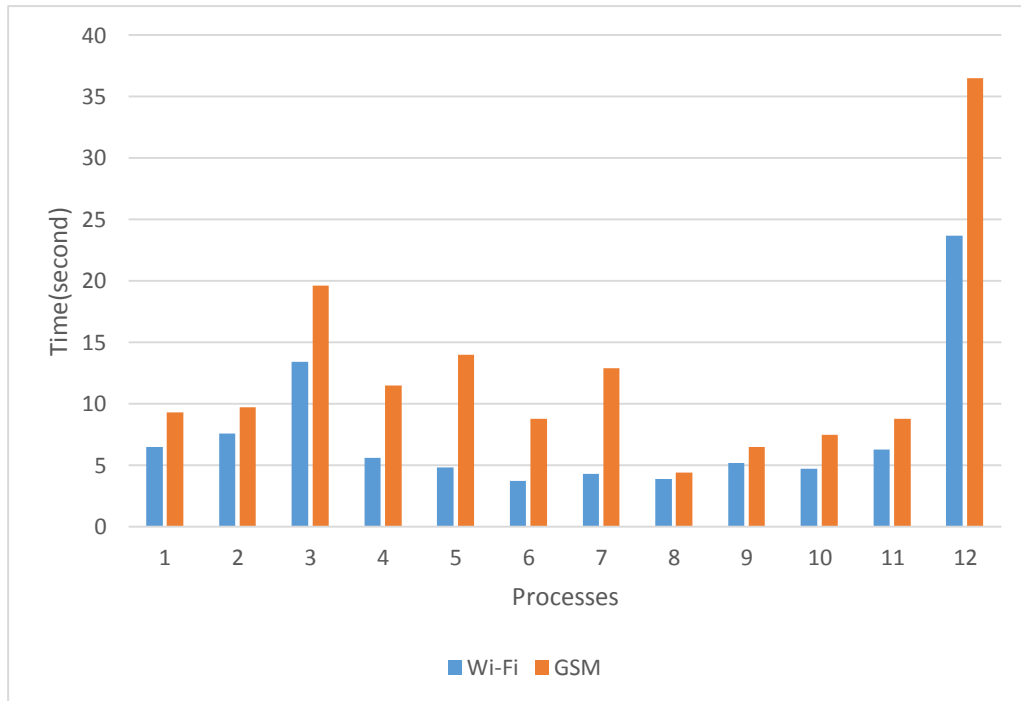


Figure 9. Connection Type Based Time Values of Processes

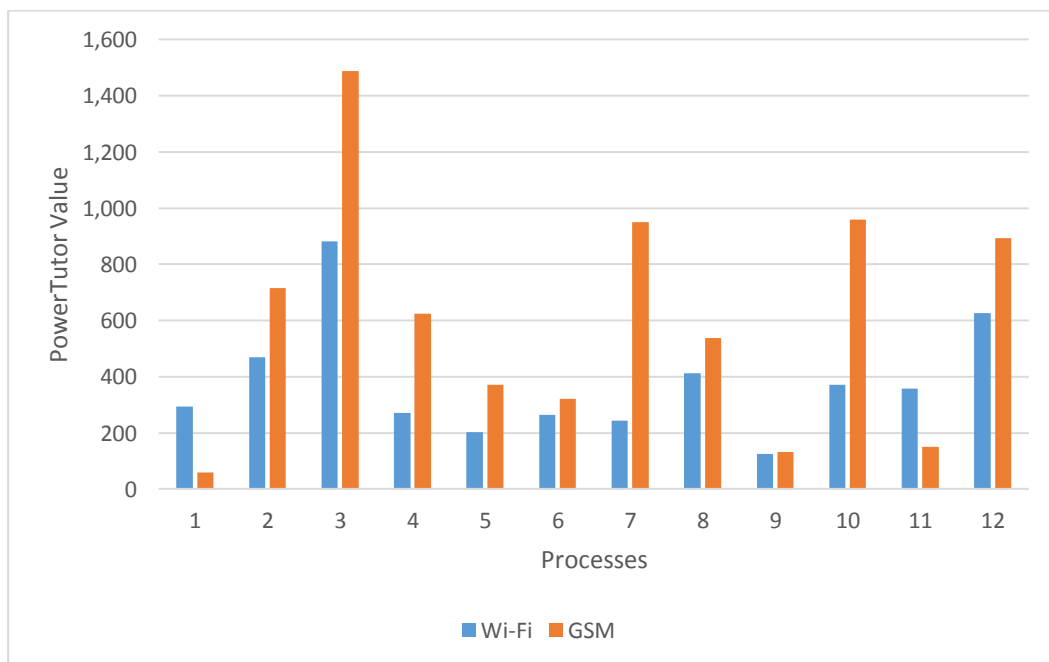


Figure 10. Connection Type Based Power Values of Processes

Connection type based average power and time values of processes are shown in Figure 9 and Figure 10. According to the figures average power and time values of GSM connection type are usually higher than Wi-Fi connection type. As a result, it is understood that WiFi connection type is more efficient.

## 5.4. ANALYSIS OF USER CONTEXT INFORMATION

Mobile device records are transferred to cloud environment along with a unique identification number therefore application records can be analyzed for each mobile device user. It is assumed that mobile device users will have different profiles so that context records and differences may be analyzed. For example, the test user has spent more time at home and office with excellent level of Wi-Fi internet connection so mostly cloud environment and Wi-Fi internet connection are preferred by the algorithm.

## 6. CONCLUSION

A context aware decision making system was developed for mobile cloud offloading in this research. In practice, the dynamic power measurement and requirement of using AsyncTask class encountered the biggest problems. Open source PowerTutor application is integrated and specialized on scope of the application for dynamic power measurement. Web services and chosen process are operated with AsyncTask class. So developed application has overcome these problems.

Context awareness is an important factor for mobile applications. Context information such as the charge level of mobile device, the plugged condition of mobile device, the GPS location data of mobile device and current Internet connection type and level are used in decision algorithm. The decision algorithm was developed for choosing platform to be used in mobile cloud offloading. As a result of using AsyncTask class completing a process takes long time. For this reason calculating the measurement power for executing a process has become more difficult. In addition to the decision algorithm a power consumption algorithm was developed.

Experimental results showed that internet connection level and user's location is vitally important in decision-making stages for mobile cloud offloading. In addition to these information performance historical data of processes and instant battery levels determined as critical information. Another important factor is the state that has a Wi-Fi connection. Because cloud transfer is usually done in cases in which the Wi-Fi connection is optimal. As a result, user's context information is determining factor for the system.

In this research three processes was defined depending on difficulty for each process. When the results are analyzed on the basis process. In cases where the degree of difficulty increases of process are increasing the amount necessary resources and cloud transfer rate is becomes higher. In further studies, it is planned to improve the system and actualize real implementation. The obtained results will be examined in detail on the basis of user context information, type of process and internet connection types.

## REFERENCES

- [1] Qi, H., Gani, A., "Research on Mobile Cloud Computing: Review, Trend and Perspectives", Arxiv.org, 2012.
- [2] Khan, A., & Ahirwar, K., "Mobile Cloud Computing As a Future of Mobile Multimedia Database". International Journal of Computer Science and Communication, 2(1), 219-221, 2011.
- [3] Huang, D., "Mobile Cloud Computing", IEEE, 2011.
- [4] Chun, B., Naik, M., Ihm, S., Patti, A., Maniatis, P., "Clonecloud: Elastic execution between mobile device and cloud," in Proc. of European Conference on Computer Systems (EuroSys'11), Salzburg, Austria, April 2011, pp. 181–194.
- [5] Cuervo, E., Balasubramanian, A., Cho, D., Wolman, A., Saroiu, S., Chandra, R. and Bahl, P., "Maui: Making smartphones last longer with code offload," in Proc. of International Conference on Mobile Systems, Applications, and Services (MobiSys'10), San Francisco, CA, USA, June 2010, pp. 49–62.

- [6] Kosta, S., Aucinas, A., Hui, P., Mortier, R., Zhang, X., “Thinkair: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading”, IEEE, 2012.
- [7] Lin, Y.D., Edward T.H., Lai, Y.C., Huang, T.J., “Time-and-Energy-Aware Computation Offloading in Handheld Devices to Coprocessors and Clouds”, IEEE, 2013.
- [8] AsyncTask, date: 01 November 2015, <http://developer.android.com/reference/android/os/AsyncTask.html>.
- [9] 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 Proc. of IEEE INFOCOM, Orlando, FL, USA, March 2012, pp. 945–953.
- [10] Namboodiri, V., Ghose, T., “To cloud or not to cloud: A mobile device perspective on energy consumption of applications,” in Proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM’12), San Francisco, CA, USA, June 2012, pp. 1–9.
- [11] Wolski, R., Gurun, S., Krintz, C. and Nurmi, D., “Using bandwidth data to make computation offloading decisions,” in Proc. of International Parallel & Distributed Processing Symposium (IPDPS’08), Miami, FL, USA, April 2008, pp. 1–8.
- [12] Xia, F., Ding, F., Li, J., Kong, X., Yang, L.T., Ma, J.,” Phone2Cloud: Exploiting Computation Offloading For Energy Saving On Smartphones In Mobile Cloud Computing”, Information Systems Frontiers, 16, 95-111, 2014.
- [13] Wu, H., Wang, Q., Walter, K., “Tradeoff Between Performance Improvement And Energy Saving In Mobile Cloud Offloading Systems”, IEEE, 2013.
- [14] Lin, T.Y., Lin T.A., Hsu, C., King, C., “Context-Aware Decision Engine for Mobile Cloud”, IEEE, 2013.
- [15] Lim, K.H., Lee, B.D., “History-based Dynamic Estimation of Energy Consumption for Mobile Applications”, IEEE, 2014.
- [16] Bedregal, V., Gutierrez, C., Robert, E., Arisaca, M., “Optimizing Energy Consumption per Application in Mobile Devices”, IEEE, 2013.
- [17] Jian, L., Jin, X., Heni, A., James, W., Raouf, B.,” PowerGuide: Accurate Wi-Fi Power Estimator for Smartphones”, IEEE, 2013.
- [18] Pınarar, O., and Ozgovde, A.,” Characterization of Mobile Applications According To Their Energy Consumptions”, IEEE, 2014.
- [19] Zhang L., Tiwana, B., Qian Z., Wang, Z., Yang, L., Mao, Z., Dick, R., ”Accurate Online Power Estimation and Automatic Battery Behaviour Based Power Model Generation for Smartphones”, IEEE, 2013.
- [20] PowerTutor, date: 01 November 2015, <http://ziyang.eecs.umich.edu/projects/powertutor>.
- [21] Android, date: 01 November 2015, <https://developer.android.com/sdk/index.html>.
- [22] Eclipse, date: 01 November 2015, <https://eclipse.org/>.
- [23] Sqlite, date: 01 November 2015, <http://www.sqlite.org/>
- [24] OpenCV, date: 01 November 2015, [opencv.org](http://opencv.org).
- [25] Tesseract, date: 01 November 2015, <https://code.google.com/p/tesseract-ocr/>.
- [26] Corral L., Georgiev A., Sillitti A., Succi G., “A study of energy-aware implementation techniques: Redistribution of computational jobs in mobile apps”, Elsevier in Press, 2014.
- [27] Dey, A.K., Abowd, G.D., “Towards A Better Understanding of Context and Context-Awareness”, Technical Report GITGVU-99-22, Georgia Institute of Technology, College of Computing, 1999.
- [28] Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. "A Survey of Mobile Cloud Computing, Architecture, Applications, and Approaches", Wireless Communications and Mobile Computing, 2013.
- [29] Jia, M., Cao, J., Yang, L., “Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing”, IEEE, 2014.
- [30] Mulligan, G., Gracanin D., “A Comparison of Soap and Rest Implementations of A Service Based Interaction Independence Middleware Framework”, Proceedings of the 2009 Winter Simulation Conference, 2009.
- [31] Pu, L., Xu, J., Jin, X., Zhang, J., ”SmartVirtCloud: virtual cloud assisted application offloading execution at mobile devices' discretion”, IEEE, 2013.
- [32] Su, W.T., Siong, K., “Mobile Cloud with Smart Offloading System”, IEEE, 2012.
- [33] Yang, S., Kwon, Y., Cho, Y., Yi, H., Kwon, D., Youn, J., Paek, Y., “Fast Dynamic Execution Offloading for Efficient Mobile Cloud Computing”, IEEE, 2013.



## AUTHORS

**Mustafa TANRIVERDİ** was working in Department of Computer in Gazi University, Turkey until 2007. In 2011 he joined as Ph.D. student in Management Information System at Gazi University, Turkey. He has research interest are mobile applications, mobile learning and software development.



**Dr. M. Ali AKCAYOL** is professor in Engineering Faculty at Gazi University, Turkey. His research interests are mobile wireless networks, artificial intelligence, cloud computing, web technologies, web mining, smart buildings.

