Journal of Theoretical and Applied Information Technology
15th December 2018. Vol.96. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EARLY ANALYSIS OF SOFTWARE ARCHITECTURE TO ESTIMATE ENERGY CONSUMPTION IN ANDROID PLATFORM

HASAN SAJID ATTA AL NIDAWI[1], KOH TIENG WEI[2], AMMAR KHALEEL[3], KAREEM ABBAS DAWOOD[4]

[1,2,4] Department of Software Engineering and Information System, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. [3]Department of Computer, Faculty of Education for Girls, Universiti of Kufa, Iraq.

E-mail: [1]hassan.sajid44@yahoo.com, [2]twkoh@upm.edu.my, [3]ammar.khaleel@uokufa.edu.iq, [4]kareem.a.it@gmail.com

ABSTRACT

In recent years, the popularities of smartphones and its apps have been growing exponentially. This growth have made mobile applications more complex and require a large amount of energy to run in an efficient manner. It means mobile applications rely heavily on battery consumption. However, battery consumption is still a limiting factor in mobile applications development. To this end, a mobile software architecture plays a crucial role in determining the performance of a mobile application in terms of energy consumption. Up to date, there has been limited work to assist mobile applications developers to select the most suitable software architecture that intends to manage resources consumption during the design phase accordingly. Hence, this paper aims at presenting a consumption analysis of two different architectures, namely; Server-centric architecture and Mobile-centric architecture. This analysis helps to identify the least energy-consuming architecture. Moreover, Data Retrieval Information System (DRIS) in Android mobile applications has been used as a case study to prove the effects of software architectures concept in reducing energy consumption. The analysis results have shown that Mobile-centric architecture is less energy-consuming.

Keywords: *Software Architecture; Energy Consumption; Android Mobile Application, Data Retrieval System.*

1. INTRODUCTION

The popularities of smartphones and mobile apps have been increasing since the beginning of this century. According to B. Sanou (2015), the number of mobile-cellular subscriptions has increased from 738 million to about 7 billion currently [1]. Mobile apps and cloud services are consumed massively nowadays [2]. As reported by The Zettabyte Era (2016), the traffic of global mobile data is expected to exceed that of wired devices in 2016 [3]. For example, the data provided in the second quarter report of Facebook shows that about 88% of the active users log on to Facebook from a mobile device [4]. The widespread use of mobile applications is mainly due to the sustainability of those functionalities in terms of the devices' resources. On the other hand, the popularity of a mobile application relies heavily on its resource consumption such as battery use [5],[6] and network technology [7]. It is quite apparent that resource consumption such as for Wi-Fi are the determining factors in the success of any mobile application [8]. The energy consumption in network technology is intimately related to the characteristics of the workload and not just the total transfer size, e.g., a few hundred bytes transferred intermittently on 3G or Wi-Fi can consume more energy than transferring a megabyte in one shot.

Mobile applications that require huge resources are not recommended [8],[9] and according to Al Nidawi Hasan et.al software architecture plays a crucial role in determining the performance of a mobile application in terms of resource consumption [10]; therefore, the development of a mobile application should include analysis of software architecture. The most established architecture for mobile apps is the Server-centric (SC) approach [11], whereby mobile devices are acting as simple clients and tasks such as information storage, processing, and communication are delegated in the cloud. This approach is popular because it is able to delegate the processing workloads to the servers. Also, tasks such as aggregation of data coming from individual users is possible, thus facilitating the implementation and maintenance processes in different platforms. Nowadays, emerging mobile-centric architectures inspired by distributed processing are available [12],[13]. The choice of architectural approach (Server-centric (SC) architecture or Mobile-centric (MC) architecture) would determine the energy consumption of a mobile application especially in a Data Retrieval Information System (DRIS).

To support extensive applications in mobile phones that require retrieval of data from the phone or from remote data sources, there are many relational database systems like IBM's DB2 Everywhere 1.0, Oracle Lite, and Sybase's SQL etc. that work on hand-held devices and can provide local data storage for relational data acquired from enterprise relational databases [14]. Most of the existing systems that offer applications work based on the previous applied database that contains personal information; this information can be provided to the user for more progress. Unfortunately, the users are usually away from energy sources. So, sustainable and effective phone energy is quite essential to the functionality of the applications to keep it in running order. In a nutshell, any application that drains the battery's energy soon will be cast off by users [9], and will eventually lead to decrease in companies' revenue.

Nevertheless, resource management is one of the critical factors in determining the effectiveness of an application. Most studies focused on optimizing the resources upon the development of an application. However, work related to choosing suitable software architecture for optimal resource consumption is rather scarce [10]. Berrocal et al. [15] have followed a similar research direction, but their tested case studies, architectures, and real applications are quite limited.

The purpose of the current work is to determine the energy-efficient architecture for Data Retrieval of Criminal Information Checker System (DRCICS) in Android mobile applications. Thus, we present the energy consumption analysis of DRCICS. It is an effective application designed to be used in Android smartphones to execute primitive operations, especially retrieving data from applied databases, storing and measuring the energy consumption. This application builds by two different architectures: Server-centric architecture and Mobile-centric architecture in order to identify the least energy-consuming architecture for this kind of application. The remainder of the paper is organized as follows. Section 2 discusses methodology and the experiments. Section 3 explains the analysis, evaluation and results of data. Section 4 shows the related work, and Section 5 presents the conclusion and future works.

2. METHODOLOGY AND EXPERIMENTS

The application of DRCICS implemented by applying the Server-centric (SC) and Mobile-centric (MC) architectures, but its behavior differs depending on which architecture was used. With a Server-centric architecture, the database that contains the names of suspects was stored on a centralized server (see Figure 1.A: Server-centric architecture). In order to establish the communication with the server, Retrofit framework [16] was used. This framework was selected because it is widely used and it has better performance than the alternatives [17]. So, the app request (retrieve) the data from the server based on synchronization query. On the contrary, with a Mobile-centric architecture, which do not interact with the server, the SQLite relational database management system was used [18].The database contains the names of suspects was kept on their own mobile device and provided as a service for their application (see Figure 1.B: Mobile-centric architecture).The apps retrieved data from database, which containing the personal information of the suspects in order to manipulate these information. Besides that, the apps were employed to invoke the server or mobile in order to retrieve the content and to manage the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

reception of the response (storing data).

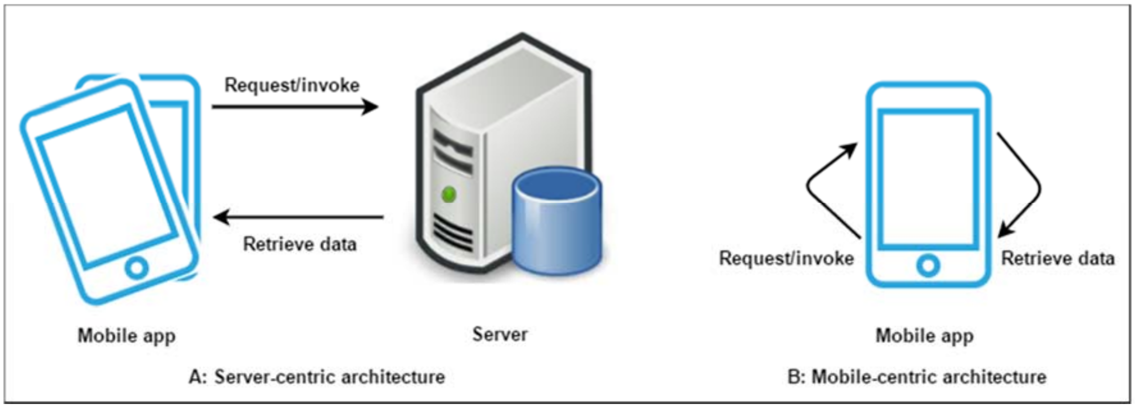

Figure 1: Application Architectures

Table 1: Primitive Operations for the DRCICS Apps

Name	Descriptions
Retrieve (content_size)	Gets a content of a given size either from the server or from the mobile.
Store (content_size)	Stores a content of a given size in the mobile device's local memory.

2.1 Experimental Setup

Predicting the resource consumption of an application is not a simple task. Trying to estimate that consumption under different architectures is even harder. To get the most accurate measurements, a prototype including the most significant functionalities would have to be built for each architecture. These prototypes would then be used to perform different simulations in conditions close to the real execution environments [19]. However, this is generally unfeasible because of the cost and effort required. Furthermore, the effort put in would not be reusable for measuring the consumption of other applications since these would also require their own prototypes with which to compare the different architectures.

To this end, we applied the most steps of the new approach (framework) presented by Berrocal (2016), and modified this framework based on our case study DRCICS [15]. To identify the commonest operations (e.g. Retrieve Data, Store) of an app and measures its consumption (see Table 1). Then, the important functionalities of an app can be composed from these primitive operations, and the expected consumption of an app can be extrapolated based on the consumptions of the primitives. This method has been used at high abstraction levels on social network case study [15]. In the current work, we developed two applications for DRCICS case study by applying the most of the technique utilized in social network case study.

Table 1 lists some of the relevant primitives for Criminal Information Checker System (CICS) apps which were used to validate the present case study. It includes operations (e.g. retrieve, store) for processing data such as the information sent by server or mobile. Several important parameters for these operations are size of the content to be retrieved/stored, interval of running time and battery temperature. In order to measure the battery life, we developed an application that executes the primitive operations (e.g. retrieve, store) in one package. Then, we registered their consumptions of battery power after each execution. During each execution, instruction such as 'run' was used to retrieve content during a specific interval of time. So, the apps were employed to invoke the server or mobile in order to pass the content and to manage the reception of the response.

Information on resources consumed during each interval of execution was stored in a log file. This log file contains data such as size of data, current power of battery in microampere-

hours (µAh) and battery-temperature, which were retrievable from BatteryManager class provided by Android OS. Processing these logs allows one to obtain the average consumption of each execution. For better results, this operation was performed without invoking other irrelevant operations. Also, it was not relegated to the background by the operating system. In other words, the data processing operations (i.e. retrieve and store) were executed while the device was at rest (without other running applications). Furthermore, These operations were encapsulated within an Android Service with an associated WakeLock [20] so that the execution of each operation was
neither stopped nor relegated to the background by the operating system.

The measured consumption pattern of an operation from a single execution might be contaminated by the consumptions of screen, app's interface, sensor, etc. Therefore, apps were developed to retrieve and store data at a specific time interval. Thus, In order to identify the architecture that is consuming less energy in an Android mobile application; readings were taken at the following six different time intervals: (A) 0-10 min, (B) 0-20 min, (C) 0-30 min, (D) 0-40 min, (E) 0-50 min, and (F) 0-60 min. The consumptions of two different architectures were measured. In general, the consumptions of retrieving and storing operations depend on the content size, battery temperature and execution time interval. Therefore, we measured the phone temperature and the package size for each execution time interval based on the usual content shared by DRCICS.

Based on the above we were able to launch an execution, put the device to rest, acquire a large set of measurements, and then stop the execution. To add this functionality, the primitive operations were encapsulated in a Timer configured to run each operation (or query) every 1000 ms. It has been reported earlier by Berrocal (2016), that the operation was executed 3000 times for 50 mins [15]. In the present research, the operation was executed for at least 600 times (for 10 min) and at most 3600 times (for 60 min). All the tests were carried out on a Samsung Galaxy S3 with Android 4.3.1 as operating system. The smartphone was in perfect condition, and the battery was purchased for the experiment. In addition, the tests were performed while the mobile was at rest (irrelevant operations, including screen, were not running). However, since the interactions with the server were achieved through Wi-Fi network, the Wi-Fi mode in the smartphone was turned on.

2.2 Data Collection

The quantitative data collecting method is conducted to analyze the primary data and measurements of (Size of Data (byte), temperature of battery (C^{O}), and Energy consumption (µAh)) for two architectures, i.e. Server-centric and Mobile-centric are collected from two mobile applications utilized for this purpose. In a nutshell, the experimental setup outline above applied on DRCICS for analysis in order to evaluate which of these architectures is less energy-consuming.

2.2.1 Mobile-centric architecture

In Mobile-centric architecture, the application is meant to retrieve data from the internal database. Once a user opens the app and start, the user has six options to select time interval (10 min, 20 min, 30 min, 40 min, 50 min, and 60 min) to operate the app. The system goes to next state of retrieving data which is retrieving data from the internal relational database and then displaying the data record and data size along with the current power of battery and battery temperature. The system proceeds to the next state of checking if the log file exists in the document folder, it will proceed to save the data, if not then creates it and saves data in it. Then checks if the request time by the user has elapsed or not, if yes it will end, but if it is no, it will proceed to next state of retrieving with data, and call data from the internal database and repeat the same process till the user requested time finish. (see Fig. 2 data retrieval of Mobile-centric architecture).For each time interval, the above experiment was repeated many times and after the fifth execution for each interval, one can notice that there is no great difference in data. Hence the experiment should be stopped in order to avoid saturation. The data collected from the log files in this architecture are tabulated in Table 2.

2.2.2 Server-centric architecture

In Server-centric architecture, the application is meant to retrieve data from the centralized server. Once a user opens the app and start, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

user has six options to select time interval (10 min, 20 min, 30 min, 40 min, 50 min, and 60 min) to operate the app. The system goes to next state of retrieving data which is retrieving data from the remote data sources on server and then displaying the data record and data size along with the current power of battery and battery temperature. The system proceeds to the next state of checking if the log file exists in the document folder, it will proceed to save the data, if not then creates it and saves data in it. Then checks if the request time by the user has elapsed or not, if yes it will end, but if it is no, it will proceed to next state of retrieving with data, and call data from the remote database on the server and repeat the same process till the user requested time finish. (see Fig. 3 data retrieval of Server-centric architecture). For each time interval, the above experiment was repeated many times and after the fifth execution for each interval, one can notice that there is no great difference in data. Hence the experiment should be stopped in order to avoid saturation. The data collected from the log files in this architecture are tabulated in Table 3.

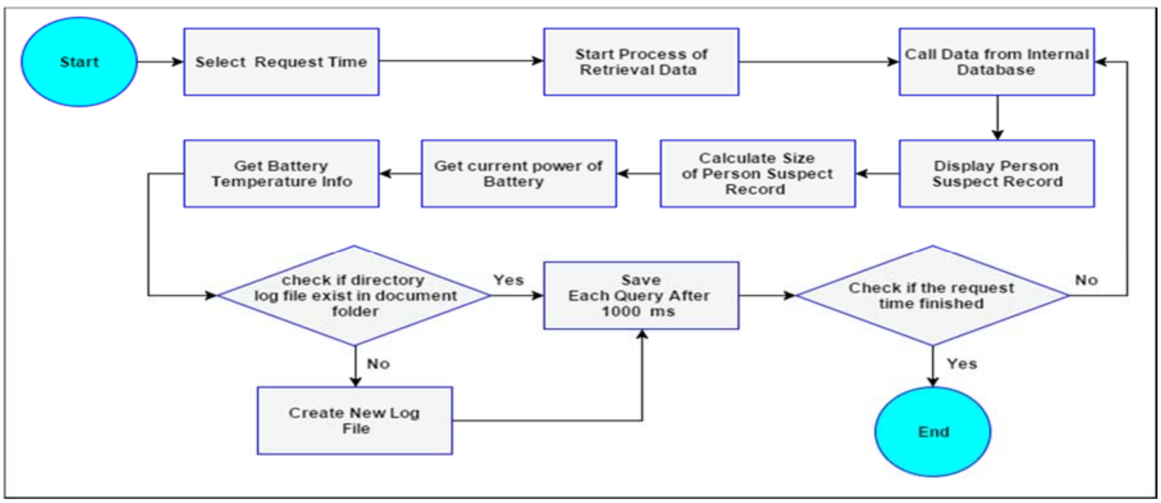

Figure 2: Data Retrieval of Mobile-centric Architecture

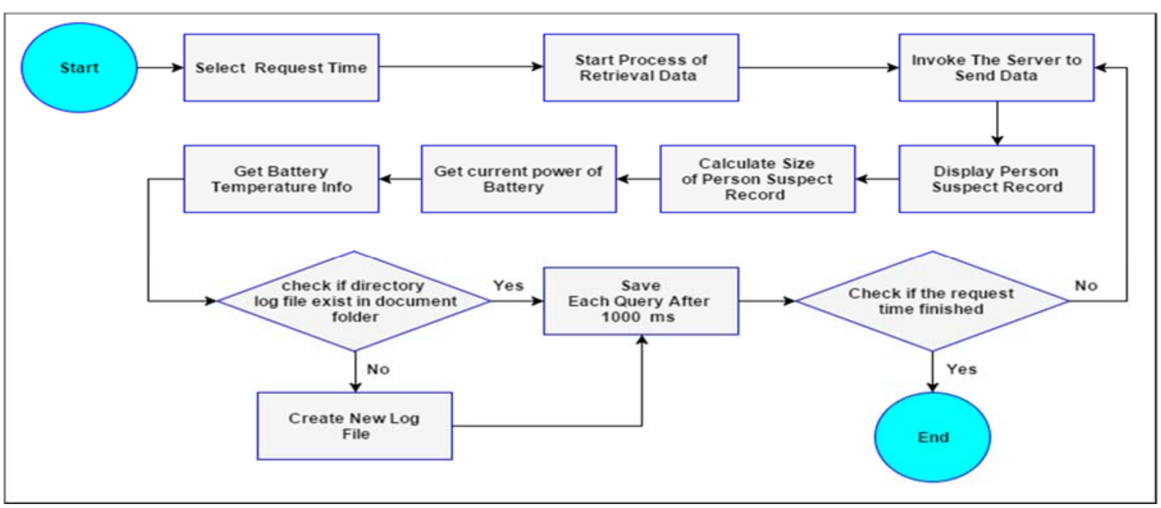

Figure 3: Data Retrieval of Server-centric Architecture

Journal of Theoretical and Applied Information Technology
15th December 2018. Vol.96. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Data Collected from Mobile-centric Architecture

Interval	Info.	Execution 1	Execution 2	Execution 3	Execution 4	Execution 5
A (0-10 min)	Size of Data (byte)	27272.00	27257.00	27275.00	27286.00	27293.00
	Average of temperature (C^O)	33.56	33.54	31.84	32.08	31.43
	Energy consumption (µAh)	21000.00	21000.00	21000.00	21000.00	21000.00
B (0-20 min)	Size of Data (byte)	54460.00	54460.00	54460.00	54460.00	54460.00
	Average of temperature (C^O)	34.71	34.29	34.47	34.46	34.72
	Energy consumption (µAh)	63000.00	63000.00	63000.00	63000.00	63000.00
C (0-30 min)	Size of Data (byte)	81576.00	81548.00	81548.00	81565.00	81589.00
	Average of temperature (C^O)	31.66	34.20	33.24	30.80	32.13
	Energy consumption (µAh)	63000.00	105000.00	105000.00	63000.00	63000.00
D (0-40 min)	Size of Data (byte)	108569.00	108569.00	108569.00	108590.00	108569.00
	Average of temperature (C^O)	34.13	33.50	33.80	30.97	34.43
	Energy consumption (µAh)	147000.00	147000.00	147000.00	84000.00	147000.00
E (0-50 min)	Size of Data (byte)	135720.00	135720.00	135708.00	135709.00	135720.00
	Average of temperature (C^O)	34.15	34.42	30.64	31.90	35.03
	Energy consumption (µAh)	189000.00	189000.00	105000.00	105000.00	189000.00
F (0-60 min)	Size of Data (byte)	162810.00	162801.00	162801.00	162801.00	162830.00
	Average of temperature (C^O)	31.75	34.24	34.30	34.63	32.70
	Energy consumption (µAh)	126000.00	210000.00	210000.00	210000.00	126000.00

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: Data Collected from Server-centric Architecture

Interval	Info.	Execution 1	Execution 2	Execution 3	Execution 4	Execution 5
A (0-10 min)	Size of Data (byte)	27292.00	27252.00	27275.00	27262.00	27267.00
	Average of temperature (C^O)	30.14	32.43	32.14	32.50	32.01
	Energy consumption (µAh)	21000.00	42000.00	21000.00	21000.00	42000.00
B (0-20 min)	Size of Data (byte)	54465.00	54459.00	54432.00	54426.00	54415.00
	Average of temperature (C^O)	35.18	32.85	32.29	32.63	32.55
	Energy consumption (µAh)	84000.00	63000.00	63000.00	63000.00	63000.00
C (0-30 min)	Size of Data (byte)	81553.00	81569.00	81569.00	81594.00	81559.00
	Average of temperature (C^O)	33.21	35.02	32.14	32.18	31.72
	Energy consumption (µAh)	105000.00	126000.00	84000.00	84000.00	84000.00
D (0-40 min)	Size of Data (byte)	108559.00	108546.00	108564.00	108578.00	108529.00
	Average of temperature (C^O)	34.69	33.21	32.35	32.47	31.27
	Energy consumption (µAh)	168000.00	126000.00	126000.00	126000.00	105000.00
E (0-50 min)	Size of Data (byte)	135718.00	135701.00	135718.00	135752.00	135676.00
	Average of temperature (C^O)	35.15	35.36	35.44	32.63	32.01
	Energy consumption (µAh)	210000.00	210000.00	210000.00	147000.00	147000.00
F (0-60 min)	Size of Data (byte)	162726.00	162754.00	162754.00	162716.00	162651.00
	Average of temperature (C^O)	36.09	32.25	32.84	31.15	34.61
	Energy consumption (µAh)	252000.00	168000.00	168000.00	168000.00	252000.00

3. Data Analysis

The aim of this approach is to analyze the collected data in order to evaluate which architecture (Mobile-centric or Server-centric) is less energy-consuming in android mobile applications. The following procedures were followed to analyze the primary data for both architectures:

1. The SPSS version 23.0 was used to analyze the collected data due its efficiency in quantitative data analysis.
2. Descriptive statistics prepared to evaluate to identify the level of temperature, size of data and energy consumption based on interval time.
3. The normality of the data is confirmed through skewness and kurtosis values.
4. The univariate outliers were identified by considering the distributions of Z scores (standardized variables) of the observed data.
5. Pearson correlations were applied to study the presence of linear relationships and also to determine the significant relationships between temperature, size of data and energy consumption for each interval of time.
6. Multiple Regression analysis used to determine the effect of size and temperature on energy consumption.
7. T test used to compare the level of energy consumption between Mobile-centric architecture and Server-centric architecture due to normal distribution of both architectures.

3.1 Descriptive Statistics

Descriptive data analysis provides a summary of features of variables and their measures and is essential for better understanding of data and clarification of results. Descriptive analysis of collected data was done through summarizing and describing results in form of tables based on mean and standard deviation of variables.

Mean: $\bar{X} = \frac{\Sigma Xi}{n}$

Standard deviation: $S = \sqrt{S^2}$

Where $S^2 = \frac{\sum(Xi-\bar{X})^2}{n-1}$

This part of analysis was prepared to evaluate and identify the level of temperature, size of data and energy consumption for both architectures (Mobile-centric and Server-centric) based on interval time (see Table 4).

3.2 Test of Normality

Normality refers to the shape of data distribution for an individual metric variable and its correspondence to the normal distribution. Normality can be assessed to some extent by obtaining skewness and kurtosis values [21]. So, in this study, skewness and kurtosis test was to examine the normal distribution of data. Normality consists of univariate normality, which can be tested by examining the skewness and kurtosis.

Skewness = $\frac{\sum_{i=1}^{N}(Yi-\bar{Y})^3/N}{S^3}$

Kurtosis = $\frac{\sum_{i=1}^{N}(Yi-\bar{Y})^4/N}{S^4}$

Where $\bar{Y}$ is the mean, S is is the standard deviation, and N is the number of data points.

Common rule of thumb statistics test normality was conducted in order to get skewness and kurtosis within -2 and +2 [22], the data are accepted as normal if the skewness and kurtosis value falls between -2 and +2. In this study the skewness test for normality produced a range of -0.583 to 0.56 2 all variables. The kurtosis test also produced a range of -1.278 to -0.446 for all variables. According to Table 4.4, both statistics fall within -2 and +2 and therefore, the studied variables are normally distributed (see Table 5).

3.3 Test of Outliers

The univariate outliers were identified by considering the distributions of Z scores (standardized variables) of the observed data, as suggested by (Seo. 2006) the basic idea of this rule is that if X follows a normal distribution, then Z follows a standard normal distribution, and Z-scores that exceed 3 in absolute value are generally considered as outliers [23].So, z-scores between –2 and 2 are not unusual. z-scores should not be more than 3 in absolute value. z-scores larger than 3 in absolute value would indicate a possible outlier.

Z-SCORE = $\frac{\boldsymbol{Xi-\bar{X}}}{\boldsymbol{sd}}$

Where Xi ~ N, $\bar{X}$ is the mean, and sd is the standard deviation of data. The result in Table 6 showed that the standardized (z) scores of the all variables ranged from (–1.916) to (2.08), representing that none of the variable exceeded this threshold.

Journal of Theoretical and Applied Information Technology
15th December 2018. Vol.96. No 23
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: Descriptive Statistic for Temperature, Size of Data and Energy Consumption

TYPE	TIME	Size		Temperature		Energy	
		Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
Mobile-centric architecture	**A (0-10 min)**	27276.60	13.83	32.49	1.00	21000.00	0.00
	B (0-20 min)	54460.00	0.00	34.53	0.18	63000.00	0.00
	C (0-30 min)	81565.20	17.85	32.41	1.33	79800.00	23004.35
	D (0-40 min)	108573.20	9.39	33.37	1.38	134400.00	28174.46
	E (0-50 min)	135715.40	6.31	33.23	1.87	155400.00	46008.70
	F (0-60 min)	162808.60	12.58	33.52	1.24	176400.00	46008.70
Server-centric Architecture	**A (0-10 min)**	27269.60	15.04	31.84	0.97	29400.00	11502.17
	B (0-20 min)	54439.40	21.62	33.10	1.18	67200.00	9391.49
	C (0-30 min)	81568.80	15.66	32.85	1.33	96600.00	18782.97
	D (0-40 min)	108555.20	18.59	32.80	1.26	130200.00	23004.35
	E (0-50 min)	135713.00	27.77	34.12	1.66	184800.00	34506.52
	F (0-60 min)	162720.20	42.20	33.39	1.96	201600.00	46008.70

Table 5: Normality Test for Temperature, Size of Data and Energy Consumption

Type	Variable	Skewness	Std. Error	Kurtosis	Std. Error
Mobile-centric	Size	-0.001	0.427	-1.277	0.833
	Temperature	-0.583	0.427	-1.139	0.833
	Energy	0.308	0.427	-1.06	0.833
Server- centric	Size	-0.002	0.427	-1.278	0.833
	Temperature	0.562	0.427	-0.446	0.833
	Energy	0.386	0.427	-0.688	0.833

Table 6: Test of Outlier for Temperature, Size of Data and Energy Consumption

Type	Zscore	Minimum	Maximum
Mobile-centric	Zscore(Size)	-1.441	1.440
	Zscore(temperature)	-1.916	1.298
	Zscore(Energy)	-1.352	1.689
Server-centric	Zscore(Size)	-1.441	1.439
	Zscore(temperature)	-1.948	2.080
	Zscore(Energy)	-1.453	1.996

Table 7: Criteria for Interpreting Strength of Relationship between Two Variables

R	Strength of Relationship
<0.2	Slight relationship
0.2-0.4	Low correlation, definite but small
0 .4-0.7	Moderate correlation, substantial relationship
0.7-0.9	High correlation, marked relationship
>0.9	Very high correlation, very dependable relationship

Source*: Guildford Rule of Thumb (1973)[24]*

3.4 Pearson Correlations

Pearson correlations were applied to study the presence of linear relationships and also to determine the significant relationships between temperature, size of data and energy consumption. The correlation helps to clarify how the variables are related in strength and magnitude. The Pearson correlations coefficient, r, values ranged from -1 to +1 . Table 7 shows

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the criteria for interpreting strength of relationship between variables.

$$r = \frac{n\left(\sum XY\right) - \left(\sum X\right)\left(\sum Y\right)}{\sqrt{\left[n\sum X^2 - \left(\sum X\right)^2\right]\left[n\sum Y^2 - \left(\sum Y\right)^2\right]}}$$

According to the results of Mobile-centric architecture in Table 8, in the first interval of time (0-10 min) the relationship between size and temperature was negative (r = -0.83) while due to constant level of energy the correlation coefficient was not applicable (NA) or not calculated. In the second interval (0-20 min) also because of constant level of energy and size the correlation coefficient were not calculated. In the third interval (0-30 min) of study a significant and negative relationship between size and energy (r = - 0.880, P = .049) was found while the relationship between temperature and energy was strong, significant and positive (r = 0.899, P = .038) while the relationship between size and temperature was not significant (r = -0.643, P = -0.26). Results showed that for next interval (0-40 min) a significant and negative relationship between size and energy (r = - 0.999, P<0.001) was found and the relationship between temperature and size also was strong, significant and negative (r = -0.968, P = .038). The relationship between energy and temperature was significant and positive(r = 0.968, P = 0.007).Results interval (0-50 min) showed a significant and positive relationship between size and energy (r = 0.998, P<0.001) and the relationship between temperature and size also was strong, significant and positive (r = 0.968, P = 0.007). The relationship between energy and temperature was significant and positive (r = 0.956, P=0.011). Results for last interval (0-60 min) showed only significant and positive relationship between energy and temperature (r = 0.955, P = 0.011).

According to the results of Server-centric architecture in Table 9, in the first interval of time (0-10 min) the relationship between size and temperature was negative (r = -0.893, P = 0.042). In the second interval (0-20 min) only the relationship between temperature and energy was positive and significant (r = 0.958, P = 0.002) .In the third interval (0-30 min) of study also only the relationship between temperature and energy was strong, significant and positive (r = 0.986, P = .002). Results showed that for next interval (0-40 min) showed a positive relationship energy and temperature which was significant (r = 0.959, P = 0.01). Results interval (0-50 min) showed only a significant and positive relationship between energy and temperature was significant and positive(r = 0.989, P = 0.001). Results for last interval (0-60 min) showed only significant and positive relationship between energy and temperature (r = 0.913, P = 0.03).

3.5 Multiple Regression Analysis

Multiple Regression analysis is one of the most popular methods for studying the relationship between an outcome variable and several independent predictor variables. The goal is to find precisely which independent variables best predict the dependent variable. In this study Multiple Regression analysis was used to determine and test eight hypotheses in order to show the effect of size of data and battery-temperature on energy consumption. Table 10: shows the criteria of multiple regression and Table 11:shows the results of regression analysis for both Mobile-centric architecture (F=150.8, p<0.001) and Server-centric architecture (F=388.2, p<0.001) were significant.

According to standardized regression coefficient the both size of data (β=0.856, p<0.001) and temperature of battery (β=367, p<0.001) had a positive and significant effect on energy consumption in Mobile-centric architecture. These results for Server-centric architecture also indicated that both size (β=0.795, p<0.001) and temperature (β=359, p<0.001) had a positive and significant effect on energy consumption .

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 8: Correlation between Temperature, Size of Data and Consumption for Mobile-centric

TIME			Size	Temperature
A (0-10 min)	**Temperature**	R	-0.83	
		p value	0.082	
	Energy	R	NA	NA
		p value	-	-
B (0-20 min)	**Temperature**	R	NA	
		p value	-	
	Energy	R	NA	NA
		p value	-	-
C (0-30 min)	**Temperature**	R	-0.624	
		p value	0.26	
	Energy	R	-.880*	.899*
		p value	0.049	0.038
D (0-40 min)	**Temperature**	R	-.968**	
		p value	0.007	
	Energy	R	-0.999**	.968**
		p value	<0.001	0.007
E (0-50 min)	**Temperature**	R	.968**	
		p value	0.007	
	Energy	R	.998**	.956*
		p value	<0.001	0.011
F (0-60 min)	**Temperature**	R	-0.638	
		p value	0.247	
	Energy	R	-0.827	.955*
		p value	0.084	0.011

** Correlation is significant at the 0.05 level (2-tailed).*

*** Correlation is significant at the 0.01 level (2-tailed).*

Table 11: Results of Regression Analysis on Energy Consumption for Both Mobile- centric Architecture and Server-centric Architecture

TYPE		B	SE	B	t value	P value	R^2	F
Mobile-centric	**(Constant)**	-558167	83514.14		-6.684	<0.001	0.918	150.8
	Size	1.13	0.073	0.856	15.465	<0.001		
	Temperature	16710.96	2517.335	0.367	6.638	<0.001		
Server-centric	**(Constant)**	-526375	55030.5		-9.565	<0.001	0.966	388.2
	Size	1.132	0.054	0.795	21.023	<0.001		
	Temperature	16267.77	1714.797	0.359	9.487	<0.001		

Table 9: Correlation between Temperature, Size of Data and Consumption for Server-centric

TIME			Size	temperature
A (0-10 min)	Temperature	R	-.893*	
		p value	0.042	
	Energy	R	-0.613	0.353
		p value	0.272	0.561
B (0-20 min)	Temperature	R	0.726	
		p value	0.165	
	Energy	R	0.662	.985**
		p value	0.224	0.002
C (0-30 min)	Temperature	R	-0.135	
		p value	0.829	
	Energy	R	-0.275	.986**
		p value	0.654	0.002
D (0-40 min)	Temperature	R	0.34	
		p value	0.575	
	Energy	R	0.415	.959**
		p value	0.487	0.01
E (0-50 min)	Temperature	R	0.091	
		p value	0.884	
	Energy	R	-0.033	.989**
		p value	0.958	0.001
F (0-60 min)	Temperature	R	-0.352	
		p value	0.562	
	Energy	R	-0.686	.913*
		p value	0.201	0.03

** Correlation is significant at the 0.05 level (2-tailed).*
*** Correlation is significant at the 0.01 level (2-tailed).*

Table 10: The Criteria of Multiple Regression

Source	Df	SS	MS	F	R^2
Regression	1	SSR	SSR/(1)	MSR/MSE	$\frac{SSR}{Total\ SS}$
Error	$n-2$	SSE	SSE/(n-2)		
Total	n -1	Total SS			

We can test following hypotheses for Mobile-centric architecture:
H0 the effect of size-of data on energy consumption is not significant in Mobile-centric.
H1 the effect of size-of data on energy consumption is significant in Mobile-centric.

To study the effect of size on energy consumption, t test for regression analysis was used. The results showed that t test = 15.465 fall in the rejection region and H0 is rejected. Therefore H1 will be accepted and the size-of data had significant effect on energy consumption in Mobile-centric (see Figure 4).

H0 the effect of battery-temperature on energy consumption is not significant in Mobile-centric.
H1 the effect of battery-temperature on energy consumption is significant in Mobile-centric.

To study the effect of battery-temperature on energy consumption, t test for regression analysis was used. The results showed that t test = 6.638 fall in the rejection region and H0 is rejected. Therefore H1 will be accepted and the battery-temperature had significant effect on energy consumption in Mobile-centric (see figure 5).

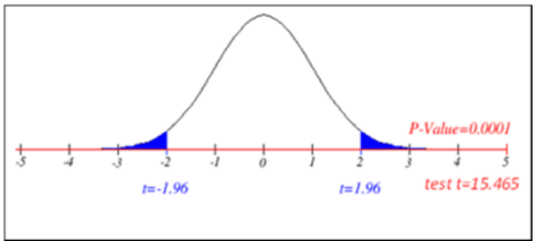

Figure 4: Significant Effect of Size of Data on Energy Consumption in MC

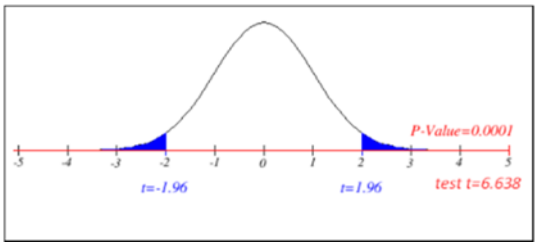

Figure 5: Significant Effect of Battery-Temperature on Energy Consumption in MC

While Server-centric architecture had the following hypothesis:
H0 the effect of size-of data on energy consumption is not significant in Server-centric.
H1 the effect of size-of data on energy consumption is significant in Server-centric.

To study the effect of size on energy consumption, t test for regression analysis was used. The results showed that t test = 21.023 fall in the rejection region and H0 is rejected. Therefore H1 will be accepted and the size of data had significant effect on energy consumption in Server-centric (see Figure 6).

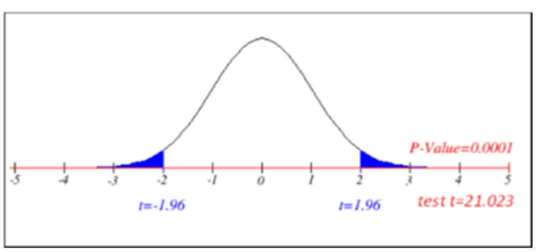

Figure 6: Significant effect of size on energy consumption in SC

H0 the effect of battery-temperature on energy consumption is not significant in Server-centric.
H1 the effect of battery-temperature on energy consumption is significant in Server-centric.
To study the effect of battery-temperature on energy consumption, t test for regression analysis was used. The results showed that t test = 9.487 fall in the rejection region and H_0 is rejected. Therefore H1 will be accepted and the battery-temperature had significant effect on energy consumption in Server-centric (see Figure 7).

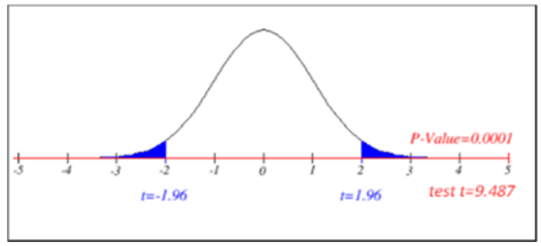

Figure 7: Significant Effect of Battery-temperature on Energy Consumption in SC

3.6 T Test

To evaluate and compare which approach; server centric architecture or mobile-centric architecture, is less energy-consuming in android mobile applications, independent t test was used due to normal distribution for both architectures. The results in Table 12 showed that the mean of energy consumption in Mobile-Centric architecture (M=105000, SD=11346.8) was lower than Server-Centric architecture (M=118300, SD=12227.7). Also, testing the following hypothesis clarifies the result above.

H0 the energy consumption in Mobile-centric architecture is lower than Server-centric architecture.
H1 the energy consumption in Mobile-centric architecture is higher than Server-centric architecture.

To study the energy consumption for each architecture, t test was used. The results showed that t test = -0.797 fall does not fall in the rejection region and H0 is not rejected. Therefore H0 will be accepted and the energy consumption in Mobile-centric architecture is lower than Server-centric architecture (see Figure 8).

Journal of Theoretical and Applied Information Technology
15th December 2018. Vol.96. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 12: Results of Mean Comparison of Energy Consumption between Mobile-centric Architecture and Server-centric Architecture

TYPE	Mean	SE	t value	p value
Mobile-centric	105000	11346.88	-0.797	0.429
Server-centric	118300	12227.74		

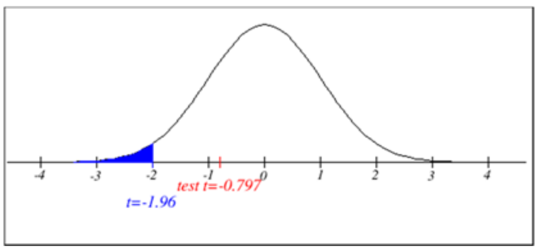

Figure 8: Testing hypothesis for MC and SC

Based on Figure 9 below, it is clear to conclude that Server-centric architecture is going to consume battery-energy higher than Mobile-centric architecture in terms of retrieving data from applied database, especially when the focus was on Data Retrieval Information System (DRIS).

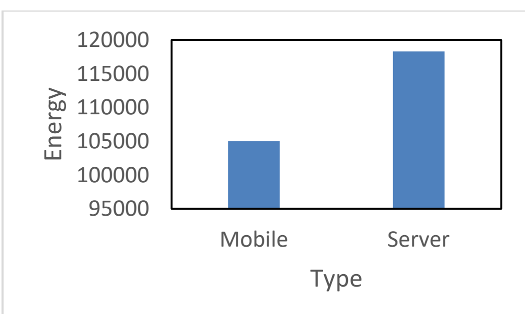

Figure 9: Mean comparison of energy consumption between Mobile-centric architecture and Server-centric architecture

4. RELATED WORKS

A resource of mobile devices is generally limited. One of the most important resources in mobile devices is energy recourse (battery). Various strategies have been proposed in [25] in order to reduce battery consumption. Technique such as off-loading resource-consuming tasks to cloud servers, for example, has been adopted by commercial mobile applications.

However, it is not applicable for application that is processing data stored locally (not on server). Besides that, managing resource consumption at the level of the device's operating system has been proposed; however, most developers found that is challenging to perform this task. Studies such as [7],[26] focus on the battery consumptions of different mobile networking technologies including Wi-Fi and 3G and the authors have proposed a new communication protocol to reduce energy consumption by delaying some communications or increasing data traffic through pre-fetching information.

Various energy-saving methods such as scheduling data transmission between mobile devices and cloud servers are reported in [27],[28]. In order to characterize the energy consumption, energy demands of mobile devices are determined from both hardware and software [29]. Their study has led to the creation of an energy-aware operating system for mobile devices designed to reduce the energy consumption of mobile applications.

The resource consumptions within specific applications are studied in [30]. As reported, a fine-grained energy profiler for smartphone applications is applied in order to measure the energy spent within an application in performing tasks such as rendering images on the screen or building an internal database for the application. While this information is beneficial for developers seeking to improve resource consumption, the application must be built before the analysis can be executed. Thus, this

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

strategy is not useful at the design stage of a mobile application.

The resource consumption of a wide array of sensors embedded in mobile applications has been studied by Moamen and Jamali [31]. They have proposed a solution to manage the sensing requirements of all the applications running on a mobile device in order to reduce the energy consumption. However, detailed information is not provided on how to design the least-consuming application.

While , a set of indicator have been proposed by [32] to measure power consumption. The authors concluded that McCabe cyclomatic complexity, weighted methods per class, nested block depth, number of overridden method, number of methods, total lines of code, method lines of code and number of parameters have strong bivariate correlations with the power consumption. Therefore, these metrics can be adopted as indicators to estimate the power consumptions of mobile applications.

So far, various techniques have been proposed to measure energy consumption such as external power monitor [33],[34]. Also, the consumption information from the battery and the modified kernel has been evaluated by [35]. In general, consumption information obtained from the devices is reliable for different types of analyses and experiments such as those proposed in the present work. A conceptual framework has been proposed by Berrocal et al. to help mobile developers during the architectural decision making process [15]. By estimating the energy consumption of mobile applications constructed under different software architectures, the proposed framework allows developers to analyze the resource consumption and its variations as the applications are scaled up. To that end, the framework analyzes the consumption of a set of primitive operations that can be used to compose complex social applications.

In short, topic such as resource consumption of mobile devices has garnered significant attention in recent years [36]. Most of the studies focus on optimizing the consumption of applications upon the development stage. However, work related to choosing the most suitable software architecture for mobile applications in terms of resource consumption is rather limited except Berrocal et al. [10]. They have described a conceptual framework to evaluate the resource consumption of a mobile application built under different software architectures. However, the number of case studies, architectures, and real applications are limited.

5. CONCLUSION

Due to the rapid growth and wide spread of mobile applications, saving energy becomes an urgent necessity. The success of mobile applications depends largely on the resources that it consumes. Here, an application that drains battery's energy soon will be rejected by users and will automatically lead to a decrease in companies' income. The core aspect in this energy consumption is the software architecture applied in its development. In this paper, we have presented the evaluation of two architectures (i.e. server-centric and mobile-centric architectures) and analysed how their battery consumption depends on the software architecture used to run the application. Generally, two mobile applications of DRIS are implemented for evaluation purposes. The application of DRIS is implemented by applying the Server-centric (SC) and Mobile-centric (MC) architectures, but its behavior differs depending on which architecture was used. The result of the evaluation shows that Mobile-centric architecture is less energy-consuming in Android mobile applications in term of retrieval of data from the applied database. The results aforementioned provide useful guidelines for the developers in terms of energy consumption for the development of mobile applications needed to connect to remote databases or relational databases. In the future, we are planning to apply the preliminary analysis of mobile software architecture to a greater number of case studies, architectures, and real applications which will enrich the evidence for the developers in terms of energy consumption.

ACKNOWLEDGMENT

This research work has been funded by the Fundamental Research Grant Scheme (FRGS) under the Malaysian Ministry of Education (MOE) for the project no. 08-01-16- 1850FR (5524957). Sincere gratitude extends to the lecturers, students and other individuals who are either directly or indirectly involved in this project.

Appendix A: Supplementary data and additional information

The average consumption values of the primitive operations were obtained using an app developed for this purpose. The source code of this app and the logs with the information of executing the primitive operations for DRCICS can be downloaded from the website below : **https://sites.google.com/view/analysis-of-architectures/home**

REFERENCES

[1] AL NIDAWI, H.S.A., et al., *ENERGY CONSUMPTION PATTERNS OF MOBILE APPLICATIONS IN ANDROID PLATFORM: A SYSTEMATIC LITERATURE REVIEW*. Journal of Theoretical & Applied Information Technology, 2017. **95**(24).
[2] Sanou, B., *ICT data and statistics division: Facts & figures*. 2015, Geneva, Switzerland: International Telecommunication Union (ITU). Retrieved from https://www. itu. int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures 2015. pdf.
[3] *Cisco, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019, Feb 2015, http://www.cisco.com/c/en/us/solutions/ collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf.*
[4] *Cisco, Cisco Visual Networking Index: The Zettabyte EraTrends and Analysis, May 2015, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.pdf.*
[5] *Facebook, Facebook Reports Second Quarter 2015 Results, July 2015, http://files.shareholder.com/downloads/AMDA-NJ5DZ/820215548x0x842047/E4B54985-9654-42C5-AEED-2489704715D1/FB_News_2015_7_29_Financial_Releases.pdf.*
[6] Qian, H. and D. Andresen. *Extending mobile device's battery life by offloading computation to cloud*. in *Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems*. 2015. IEEE Press.
[7] Merlo, A., M. Migliardi, and L. Caviglione, *A survey on energy-aware security mechanisms.* Pervasive and Mobile Computing, 2015. **24**: p. 77-90.
[8] Balasubramanian, N., A. Balasubramanian, and A. Venkataramani. *Energy consumption in mobile phones: a measurement study and implications for network applications*. in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. 2009. ACM.
[9] *A. Technologies, Android App Performance Report, Q4 2014, http://now.avg.com/wp-content/uploads/2015/02/avg_android_app_performance_ report_q4_2014.pdf.*
[10] 10. Wilke, C., et al. *Energy consumption and efficiency in mobile applications: A user feedback study*. in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*. 2013. IEEE.
[11] Xu, G., et al., *A data privacy protective mechanism for wireless body area networks.* wireless communications and mobile computing, 2016. **16**(13): p. 1746-1758.
[12] Tsai, F.S., et al., *Design and development of a mobile peer-to-peer social networking application.* Expert Systems with Applications, 2009. **36**(8): p. 11077-11087.
[13] Guillen, J., et al., *People as a service: a mobile-centric model for providing collective sociological profiles.* IEEE software, 2014. **31**(2): p. 48-53.
[14] Rahaman, M.A. and M. Bauer. *A comparison of software architectures for data-oriented mobile applications*. in *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*. 2013. IEEE.
[15] Berrocal, J., et al., *Early analysis of resource consumption patterns in mobile applications.* Pervasive and Mobile Computing, 2017. **35**: p. 32-50.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[16]Square Open Source, Square, 2015, http://square.github.io/retrofit/.
[17]J. Ruesch, Android async http clients: Volley vs retrofit, 2013, http://instructure.github.io/blog/2013/12/09/volley-vs-retrofit/.
[18]SQLite Home Page. [Online]. Available: https://www.sqlite.org/. [Accessed: 20-Mar-2017].
[19] Mittal, R., A. Kansal, and R. Chandra. *Empowering developers to estimate app energy consumption*. in *Proceedings of the 18th annual international conference on Mobile computing and networking*. 2012. ACM.
[20]Android OS, Wakelock, 2015, http://developer.android.com/reference/android/os/PowerManager.WakeLock.html.
[21] Pallant, J., *SPSS Survival Manual: A Step By Step Guide to Data Analysis Using SPSS for Windows (Version*. 2005, Allen & Unwin Pty, Crows Nest, Australia.
[22] Darren, G. and P. Mallery, *SPSS for Windows Step by Step: A simple guide and reference.* '^'eds.'): Book Spss for Windows Step by Step: A Simple Guide and Reference, Needham Heights, MA: Allyn & Bacon, 1999.
[23] Seo, S., *A review and comparison of methods for detecting outliers in univariate data sets*. 2006, University of Pittsburgh.
[24] Kameli, S. and R.B. Baki, *The impact of vocabulary knowledge level on EFL reading comprehension.* International Journal of Applied Linguistics and English Literature, 2013. **2**(1): p. 85-89.
[25] Vallina-Rodriguez, N. and J. Crowcroft, *Energy management techniques in modern mobile handsets.* IEEE Communications Surveys & Tutorials, 2013. **15**(1): p. 179-198.
[26] Rice, A. and S. Hay, *Measuring mobile phone energy consumption for 802.11 wireless networking.* Pervasive and Mobile Computing, 2010. **6**(6): p. 593-606.
[27] Terefe, M.B., et al., *Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing.* Pervasive and Mobile Computing, 2016. **27**: p. 75-89.
[28] Shi, T., et al., *An energy-efficient scheduling scheme for time-constrained tasks in local mobile clouds.* Pervasive and Mobile Computing, 2016. **27**: p. 90-105.
[29] Vallina-Rodriguez, N., et al. *Exhausting battery statistics: understanding the energy demands on mobile handsets*. in *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*. 2010. ACM.
[30] Pathak, A., Y.C. Hu, and M. Zhang. *Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof*. in *Proceedings of the 7th ACM european conference on Computer Systems*. 2012. ACM.
[31] Moamen, A.A. and N. Jamali. *Share sens: An approach to optimizing energy consumption of continuous mobile sensing workloads*. in *Mobile Services (MS), 2015 IEEE International Conference on*. 2015. IEEE.
[32] Keong, C.K., et al. *Toward using software metrics as indicator to measure power consumption of mobile application: A case study*. in *Software Engineering Conference (MySEC), 2015 9th Malaysian*. 2015. IEEE.
[33] Zhang, L., et al. *Accurate online power estimation and automatic battery behavior based power model generation for smartphones*. in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*. 2010. ACM.
[34] Jung, W., et al. *DevScope: a nonintrusive and online power analysis tool for smartphone hardware components*. in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*. 2012. ACM.
[35] Yoon, C., et al. *AppScope: Application Energy Metering Framework for Android Smartphone Using Kernel Activity Monitoring*. in *USENIX Annual Technical Conference*. 2012.
[36] Hans, R., et al. *Knowledge for a longer life: development impetus for energy-efficient smartphone applications*. in *Mobile Services (MS), 2015 IEEE International Conference on*. 2015. IEEE.