Mobioff: The Efficiency and Secured Dynamic Execution Offloading in Mobile Cloud Computing

N Kandavel, A Kumaravel

Abstract*: Smart Mobile Phone devices are widely used for current daily usages. However it has the limited Capacity in terms of the parameters as Memory, CPU, Power Consumption and network connectivity. So the main issue is to address the problem of complex and huge applications which run on mobile phones, to decide offloading schedule dynamically. so that to optimize the battery lifetime, memory usage and CPU utilization. In this paper first to extract and estimate the run time parameters in mobile devices with given specification for applications like sorting and image processing techniques. With simulation of task manager and operating system interfaces. Secondly to find out the the optimal values using with the programming parameters in case the remote cloud execution by offloading. In the later case hence we combine these two experiments to identify the suitable configuration and types of applications and complexity of the same. In additionally once decided the offloading decision, using Cloudme tool to authenticate and encrypted the application data before it send to cloud.

Keywords: Mobile Computing, Mobile Cloud Computing, Computation Offloading, Security, MatLab and Cloudme.

I. INTRODUCTION

Smart Mobile phones are mainly used for storing and processing of data parallel, but continuous this process is not possible because of due to limitations of battery, memory and CPU usage. So overcome this problem using the mobile cloud computing (MCC). MCC is the two portions one is mobile computing and another one is cloud computing. Mobile phone works are offloading to the remote, the huge number of resources, so it increases the battery life time, memory usage and CPU utilization. In cloud computing clients works with different application via with internet, it refers the infrastructure for both data storage and data processing. On demand basis MCC is the main useful for minimizing the development and execution cost of mobile applications. In some complex or huge computation task mobile phone is not possible computation to the work, so the work is offload to cloud. Some main parameters are memory, battery and CPU usage.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author

N Kandavel*, Research Scholar, Department of Computer Science and Engineering, BIHER - Bharath Institute of Higher Education and Research, Chennai, India.

A Kumaravel, Professor, Department of Information Technology, BIHER - Bharath Institute of Higher Education and Research, Chennai, India.

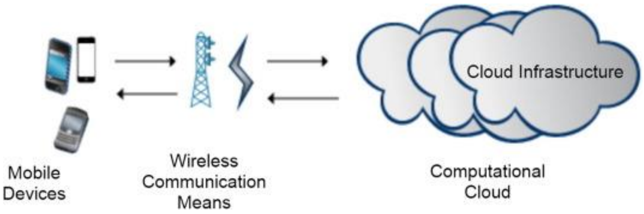


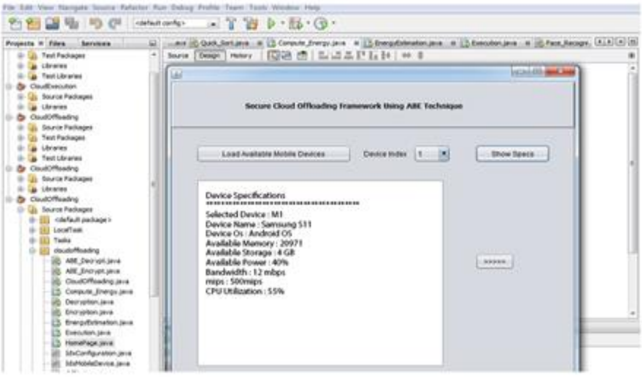
Fig.1 Mobile Offloading Technique

In this Paper, novel work to extract and estimate the battery usage, memory usage and CPU utilization based on application the run time.. Next to find out the optimal values using the integer programming. Based on the optimal values offloading decision is made. Using java code in net beans we extract the values of parameters for the applications as (Quick sort, Face detection and Gaussian blur).We get the optimal values from the Objective function.

Based on the optimal values we decide the mobile application will work on locally or cloud. The optimization model is provides the offloading decision dynamically at run time based on the constraints such as CPU utilization, Memory and energy consumption. Four different types of mobile phones are used in the experimental studies to test the optimization model to improve the application performance additionally used the Cloudme tool for authenticate the data transferring for security purpose.

II. RESEARCH METHODOLOGY

Reader: This is initial module, it is responsible for read all parameters such as size, CPU, Memory and Power. Once all the values of availability in the mobile nodes tried to communicate the estimator module. Here as an example we use four mobile nodes parameters.

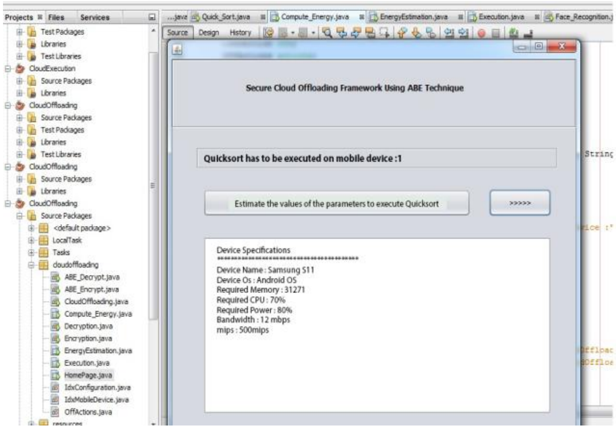


Secure Cloud Offloading Framework Using ABE Technique

```
Device Specifications
*************************************
Selected Device : M1
Device Name : Samsung S11
Device Os : Android OS
Available Memory : 20971
Available Storage : 4 GB
Available Power : 40%
Bandwidth : 12 mbps
mips : 500mips
CPU Utilization : 55%
```

Fig.2 Available Capacity of Parameters

Estimator: The Behavior for this module is to calculate the required parameters to computation the application. The values are extracted from the task manager by using through java code using net beans. Once the values are estimated based on the application (Quick sort, Face detection and Gaussian Blur) all are communicate to the Profile Module.



Secure Cloud Offloading Framework Using ABE Technique

Quicksort has to be executed on mobile device :1

Estimate the values of the parameters to execute Quicksort

Device Specifications
****
Device Name : Samsung S11
Device Os : Android OS
Required Memory : 31271
Required CPU : 70%
Required Power : 80%
Bandwidth : 12 mbps
mips : 500mips

Fig.3 Required Capacity of Parameters for the Applications

Profile: This profile module is responsible for getting the values of Execution time,CPU utilization, Memory and Energy Consumption from the estimator module. Additionally to generate new database for updating each data, this process mainly used for decision maker module.

Decision Maker: After This is the core module of the framework, it contains the integer programming model in the runtime with three constraints based on Power Consumption, CPU usage and Memory Usage. Based on the four mobile nodes and objective function produced the optimal values for these nodes. Using the MatLab tool to getting to getting the optimal values for these constraints. Based on this optimal values we decided the execution run on locally or Remote (Cloud)x_i=0 (local)or x_i-=1(cloud)

The Objective function is

$$\text{Min}(R_{transfer} * W_{tr} + R_{memory} * W_{memory} + R_{CPU} * W_{CPU} + R_{power} * W_{power}) \quad € \{0,1\} \qquad (1)$$

where

$$R_{transfer} = \sum_{i=1}^{n} input_i * x_i$$

$$R_{memory} = \sum_{i=1}^{n} memory_i * (1-x_i) + \sum_{i=1}^{n} memory\_Sec_i * x_i$$

$$R_{CPU} = \sum_{i=1}^{n} CPU_i * (1-x_i) + \sum_{i=1}^{n} CPU\_Sec_i * x_i$$

$$R_{power} = \sum_{i=1}^{n} power_i * (1-x_i) + \sum_{i=1}^{n} power\_Sec_i * x_i$$

$R_{trasfer}$,R_{memory},R_{CPU} and R_{power} represent the Required for input size, Memory used,CPU utilization and Energy consumption for i respectively.

W_{tr},W_{memory},W_{CPU} and W_{power} are the weights for each of these requirements, based on the objective function three constraints are,

Memory usage Constraints

$$\sum_{i=1}^{n} memory_i * (1-x_i) + \sum_{i=1}^{n} memory\_Sec_i * x_i \leq M_{th} \quad (2)$$

In memory two sections are used one execution in local another one is execution in remote. Once it decides made in offload prior it must be encrypted and send.M_{th} is the total size of memory.

Total Execution Time constraints: In the processing time execution in the cloud less than the application run in mobile devices, S_M and S_C Processor speeds and C be the number instructions involved in the Methods. C_{sec} represents number of instructions encrypted in that prior to send D network bandwidth and B time required to transfer this data.

$$Exe\_time\_local > Exe\_time\_remote \qquad (3)$$

$$Exe\_time\_local = \frac{C}{SM} * x_i$$

$$Exe\_time\_remote = \left(\frac{Csec}{SM} + \frac{C}{Sc} + \frac{D}{B} + t_{overhead}\right) * x_i$$

Total energy consumption for application constraints: P_M need to consume watts to execute the application in local, P_d idle time and P_r W for transferring data to the cloud.

$$Energy\_Con\_local > Energy\_con\_remote \qquad (4)$$

$$Energy\_Con\_local = P_m + \frac{C}{SM} * x_i$$

$$Energy\_con\_remote = \left(P_{sec} + \frac{Csec}{SM}\right) + \left(P_d + \frac{C}{SC}\right) + \left(P_r + \frac{D}{B}\right) * x_i$$

After the optimization model, we decided the execution place. if suppose the execution on cloud, before the values are authenticated through the Cloudme tool and also encrypted. After that the data should be sending to cloud. CloudMe is a file storage services operated by CloudMe AB that offers cloud storage file synchronization. We use separate user name and password. We create space for authenticate the security services.

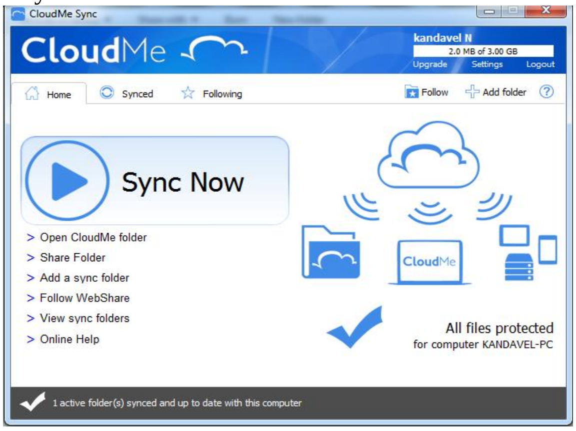


Fig.4 CloudMe Tool for Storing and Authenticate the Data Transfer

III. FRAMEWORK EXECUTION FLOW

Input: Mobile phones Memory, CPU usage and Energy consumption

Output: Decide place for work (Local or Remote)

```
1: Read Mobile Nodes (M1, M2,M3,M4)   //Consider four mobile  i.e n=4
2: Each of i do
3: Read $R_{transfer}$,$R_{memory}$,$R_{CPU}$,$R_{power}$ Output of Estimator
4: Get the solution from the optimization model to determine the decision variables  x1,x2,x3,x4
5: if the decision is offloading        //i.e $x_i$=1
```

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

```
6: send the application to cloud for remote execution
   else
7: Execute the application locally.
   end if
```

First we read the input parameters, consider four mobile nodes. For each nodes i read one by one. By using the Estimator output required CPU,Memory and Power consumption for applications. Second used the Required parameters to get the solution from optimization model to determine the decision for x1,x2,x3,and x4.if the decision is offloading x_i=1 or application executed in local.

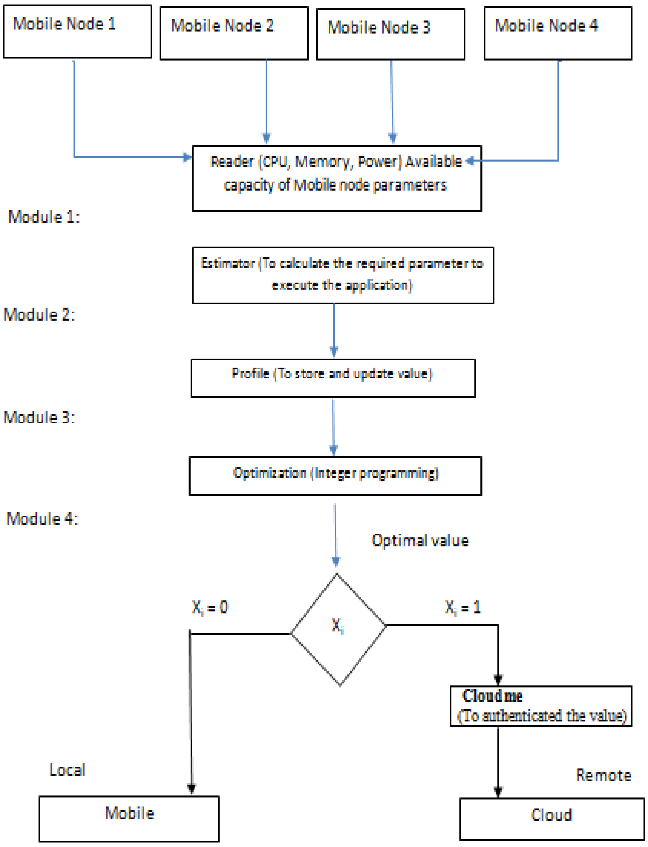

Mobile Node 1
Mobile Node 2
Mobile Node 3
Mobile Node 4
Reader (CPU, Memory, Power) Available capacity of Mobile node parameters
Module 1:
Estimator (To calculate the required parameter to execute the application)
Module 2:
Profile (To store and update value)
Module 3:
Optimization (Integer programming)
Module 4:
Optimal value
X_i = 0
X_i
X_i = 1
Cloud me (To authenticated the value)
Local
Remote
Mobile
Cloud

Fig.5 Flowchart for Execution Flow

IV. RESULTS AND DISCUSSION

Tabulated for the Co-efficient of the parameters for input, memory and power are shown in table 1 to 3.

Table.1 Mobile nodes parameter for available and required value for quick sort application

S.No	Mobile Nodes (M1, M2, M3, M4)	Input Parameters(Availability)				Output Parameters(Requirements)			
		CPU	Memory (KB)	Power	Size	R_CPU	R_Memory (KB)	R_Power	R_Size
01	M1	55%	2097	40%	500	70%	3127	80%	500
02	M2	68%	2563	75%	380	70%	2952	75%	380
03	M3	76%	2194	85%	1000	76%	2254	78%	1000
04	M4	55%	2071	60%	750	75%	3548	85%	750

Table.2 Mobile nodes parameters for available and required value for Face detection application

S.No	Mobile Nodes (M1, M2, M3, M4)	Input Parameters(Availability)				Output Parameters(Requirements)			
		CPU	Memory (KB)	Power	Size	R_CPU	R_Memory (KB)	R_Power	R_Size
01	M1	60%	2090	65%	400	70%	3123	82%	400
02	M2	72%	2258	78%	550	70%	1252	75%	550
03	M3	68%	2059	55%	700	76%	2562	81%	700
04	M4	81%	2002	71%	900	75%	2005	83%	900

Table.3 Mobile Nodes parameters for available and required value for Gaussian Blur application

S.No	Mobile Nodes (M1, M2, M3, M4)	Input Parameters(Availability)				Output Parameters(Requirements)			
		CPU	Memory (KB)	Power	Size	R_CPU	R_Memory (KB)	R_Power	R_Size
01	M1	65%	15585	62%	100*100	70%	21252	75%	100*100
02	M2	78%	18580	75%	200*200	78%	19920	81%	200*200
03	M3	80%	20190	57%	512*512	88%	21217	85%	512*512
04	M4	58%	22295	63%	512*1024	68%	22585	91%	512*1024

The results of quick sort application are shown in figure 6.

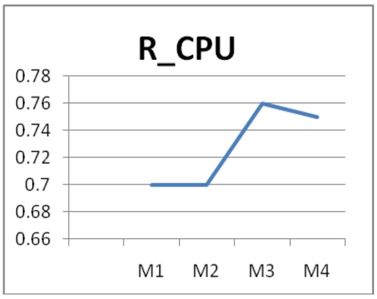


Fig.6(a) Required CPU utilization

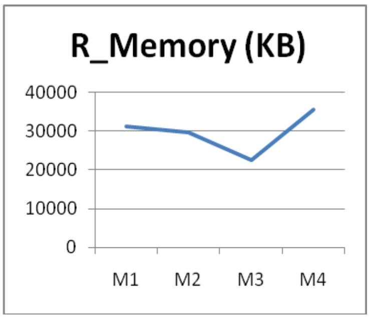


Fig.6(b) Required Memory(KB)

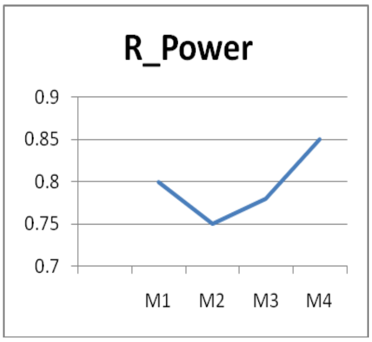


Fig.6(c) Required Energy Consumption

Retrieval Number: A2065109119/2019©BEIESP
DOI: 10.35940/ijeat.A2065.109119
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

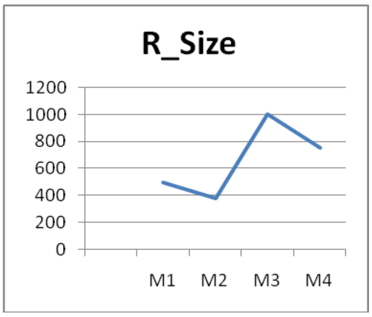


Fig.6(d) Size of the input

The results of face detection application are shown in figure 7.

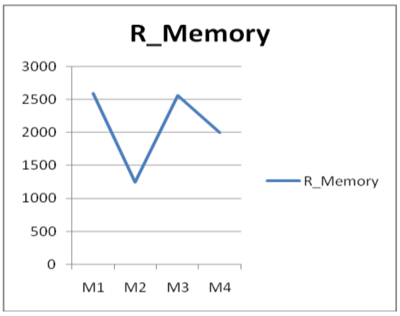


Fig.7(a) Required Memory

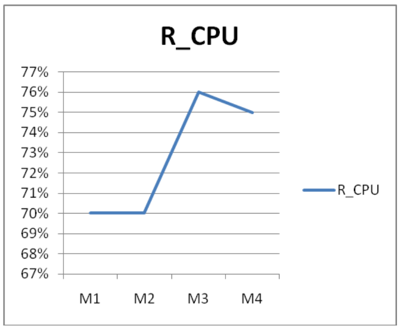


Fig.7(b) Required CPU Utilization

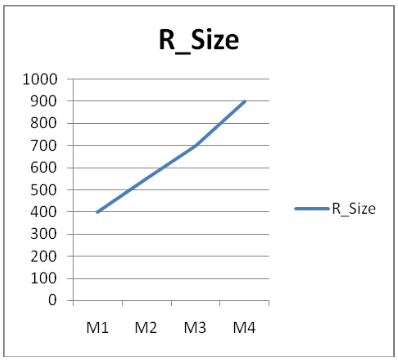


Fig.7(c) Required Size

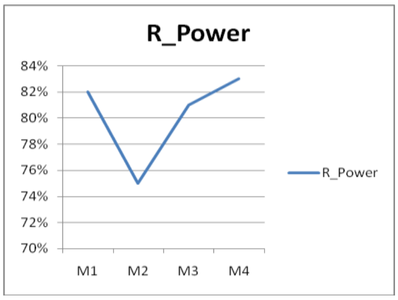
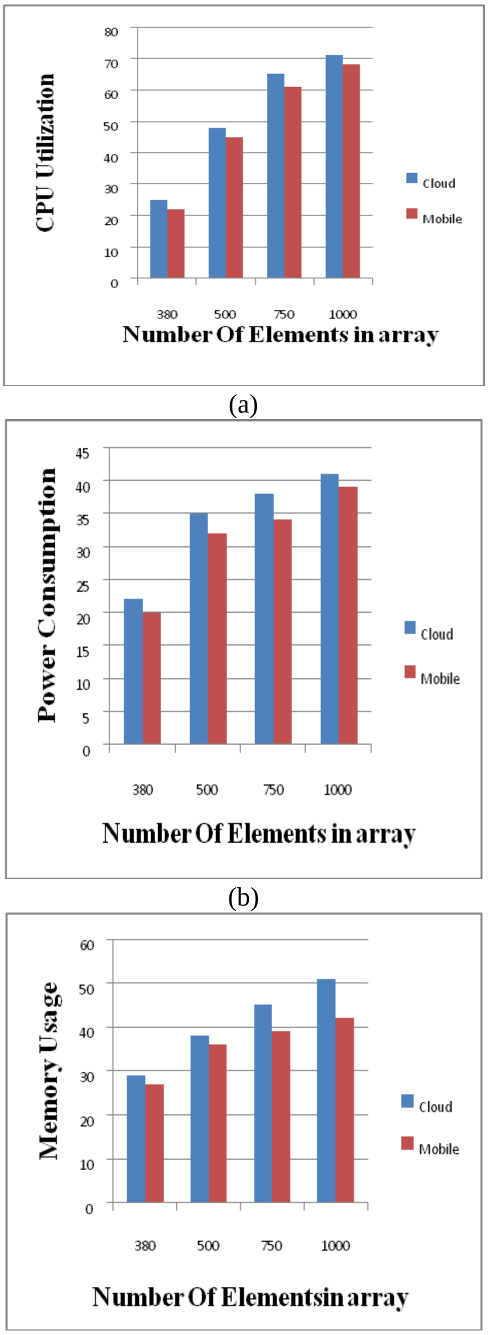


Fig.7(d) Required Power Consumption

In the proposed work, integer programming model is applied to the applications. Based on optimal values the execution place is decided locally or Cloud. But in that experimental work two different cases are used to prove it. Case one for application methods executed in locally. Another case for application methods executed in cloud.

In the solution the proposed integer programming model is applied to the constraints based on this to select the decision for these applications (sorting, image processing techniques).In that experimental work two cases are used. One is application executed in Mobile devices and another one is application executed in Cloud. Additionally cloudme used for to protect the data to secured transfer.

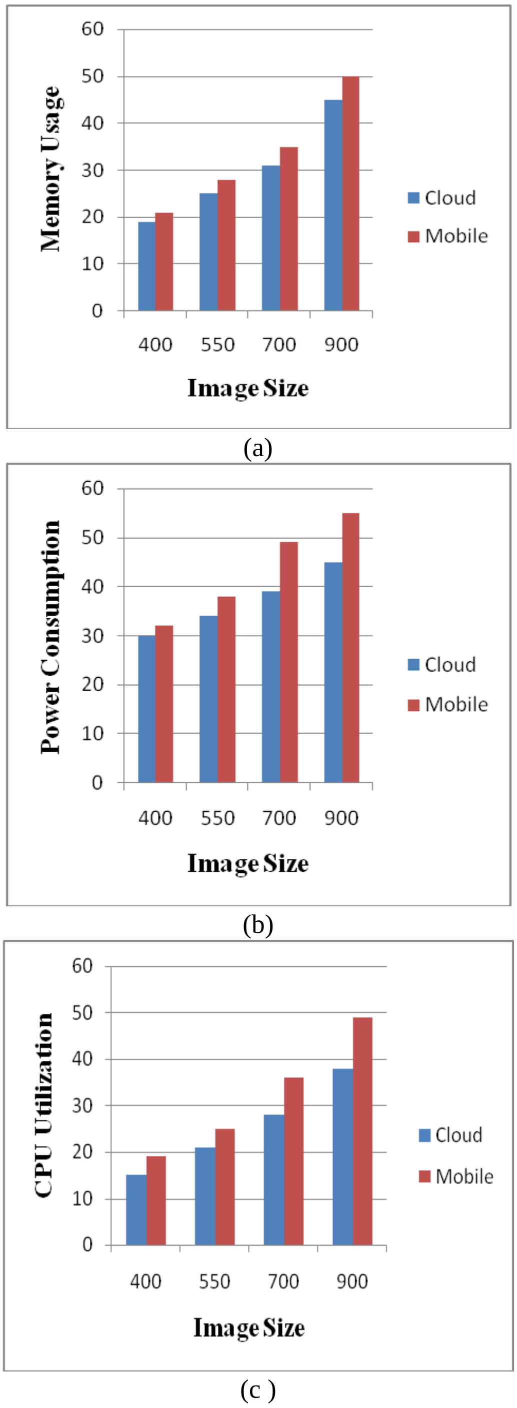


(a)

(b)

(c)

Fig.8 Evaluation Result for the quick sort application



(a)

(b)

(c)

Fig.9 Evaluation Result for the Face detection application

V. CONCLUSION

In this paper, a novel secured framework is optimized to the computation and data offloading to interaction between the mobile devices to cloud. The main issue is to be addressed Huge and complex applications are run on mobile devices due to limited capacity of parameters such as CPU, Memory and Energy consumption. The required capacity of the parameters calculated by the estimator module. The values are apply to the objective function and used the three constraints to get the optimal values for (x1,x2,x3 and x4).Based on the optimization technique and integer programming the decision is made. In that proposed, to identify the suitable configuration, types of applications and execution placed. The result is too improved reduce the consumption of battery, CPU utilization and memory usage. Once the decision is made, to execute the application in cloud, using the Cloudme tool for storing and authenticate the data to encrypted. In addition simultaneous to execute the cloud to reduce the consumption of battery, Memory usage and execution time.

REFERENCES

1. Ibrahim A Elgendy, Wei-Zhe Zhang, Chuan-Yi Liu, and Ching-Hsien Hsu, "An Efficient and Secured Framework for Mobile Cloud Computing", Ieee Transactions on Cloud Computing, 2018, DOI 10.1109/TCC.2018.2847347.
2. N. Vallina-Rodriguez and J. Crowcroft, "Energy management techniques in modern mobile handsets," IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 179–198, 2013.
3. B. Sosinsky, Cloud Computing Bible. Wiley, 2010.
4. G.Motta, N. Sfondrini, and D. Sacco, "Cloud computing: An architectural and technological overview", International Joint Conference on Service Sciences, vol. 3, pp. 23–27, 2012.
5. D.Kovachev, Y.Cao, and R.Klamma, "Mobile cloud computing: A comparison of application models",Computer Science, 2012.
6. A.U.R. Khan, M.Othman, S.A. Madani, and S.U. Khan, "A survey of mobile cloud computing application models", IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 393–413, 2014.
7. M. Shiraz, A.Gani, R. H. Khokhar, and R.Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1294–1313, 2013.