An Efficient Algorithm for Mashup Services in Mobile Cloud Computing

Kuljeet Kaur
Department of Computer Science and Engineering
Guru Nanak Dev University, Punjab, India.
kuljeetkaur032@gmail.com

Kamaljot Singh
Department of Computer Science and Engineering
Lovely Professional University, Jalandhar, Punjab, India.
Kamaljotsingh2009@gmail.com

***Abstract* — Advances in mobile communication networks and improved mobile experience had lead to increased users of smart phones. Due to limited battery power and limited computational power, it is difficult to carry out some heavy computational tasks over the mobile as they drains battery faster and consumes more time. In this paper, we proposed a new technique in form of an algorithm and a software prototype, demonstrating the working of task offloading algorithm in relation to processing over mobile. We had considered the computational power of mobile device, Task load, Task type for the designing of an effective trade off for task offloading and evaluated the results under the response time and mobile device energy consumption evaluation metrics.**

Keywords- Cloud Computing; Gray-scale; Task Offloading; Mobile cloud Computing; histogram Equalization.

I. Introduction

Advances in mobile communication networks and improved mobile experience had lead to increased users of smart phones (both smart phones and tablets). According to [1], Figure 1, demonstrates the smartphone users count in billions and expected count in future years.

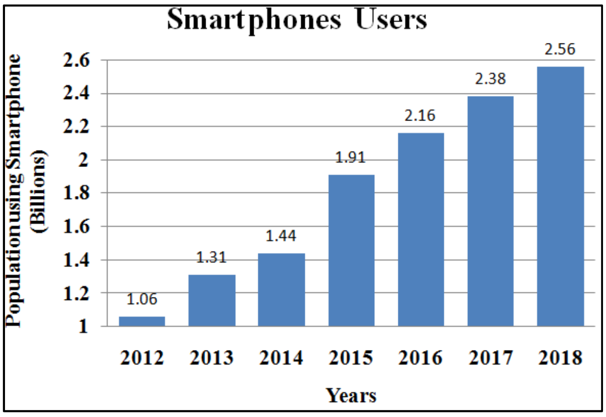

Figure 1. Smart-phones users in billions.

We can visualize form the Figure 1, that the smart phone users were 1.06 billion and increased to 1.91 billion in 2015, and expected to increase to 2.56 in 2018. However, due to the limited computing power, limited energy, and increasing sophisticated applications demanded by users the mobile devices had reached to the bottleneck of hand-held mobile devices.

Thus, the limitation of energy has been resolved by emerging cloud computing technology by a new enhanced technique to compute or process the data by offloading the task to the cloud or by mashing up the services for better efficiency. A Mashup is a Web page or application that uses and combines data, presentation or functionality from two or more sources to create new services [2], [3]. In this paper, we proposed a new technique in form of a software prototype that does application specific task offloading from mobile device to cloud. We had considered basic tasks in imaging like gray-scaling and histogram equalizations. Our proposed technique makes use of mobile computational power, task type, image resolution to decide whether to process the task over mobile or offload it to cloud.

This paper is organized as, In Section II, we had discussed about some close works to our proposed work, In Section III, Problem is formulated in form of an algorithm. Experimental Settings are described in Section IV. Results, Conclusion and Future work is detailed in Section V and VI. Finally, the paper is closed with References.

II. Related Works

Magurawalage et al. [4], had proposed a system for mobile cloud computing(MCC) that consist of a middle layer between mobile device and their cloud infrastructure called cloudlets. Cloudlets basically acts as a localized service point nearer to mobile device, improving the performance of mobile cloud services. They had considered the energy consumption for task execution and the network status for response time. They also introduced data caching mechanism at cloudlets for overall improvement of MCC performance. Their results had demonstrated the effectiveness and efficiency of proposed algorithm in terms of response time and energy consumption.

J. Luzuriaga et al. [5], had evaluated the trade offs of computations offloading using a specific application of face recognition for smart phones, where the recognition is a service in the cloud. They presented an application for mobile devices that performs computations over mobile and cloud device and measure the performance of both.

Qi H. and Gani A. [6], concluded that there are three main optimization approaches in MCC, which are focusing on the limitations of mobile devices, quality of communication, and

DOI 10.5013/IJSSST.a.16.06.09

ISSN: 1473-804x online, 1473-8031 print

division of applications services. With the high increasing of data computation in commerce and science, the capacity of data processing has been considered as a strategic resource in many countries. Mobile cloud computing (MCC) as a development and extension of mobile computing (MC) and cloud computing (CC) has inherited the high mobility and scalability.

H. T. Dinh et al. [7] proposed that Mobile cloud computing is one of mobile technology trends in the future since it combines the advantages of both mobile computing and cloud computing, thereby providing optimal services for mobile users. Author describes the applications supported by mobile cloud computing including mobile commerce, mobile learning, and mobile healthcare have been discussed which clearly show the applicability of the mobile cloud computing to a wide range of mobile services. Then, the issues and related approaches for mobile cloud computing have been discussed.

III. Problem Formulation

Services mashup are websites or applications that combine service from more than one source into an integrated application. Mash up service is a new web application model that use existing web services to create composite service. The mash up services provides flexible and dynamic services in which QoS metrics play an important role.

In the mobile cloud computing lot of limitations like low bandwidth, power consumption, energy etc. exists as identified in the literature survey. Along with, in the web services two particular characteristics of a smart phone are restricted i.e. power supply and low-end hardware resources, compared to high-end servers as available in distributed computing environments. Compute-intensive and rich graphics-based applications in a smart phone may fully utilize the CPU and consume a large amount of the battery power accordingly. This leads to inefficient processing on mobile devices and there is a need to integrate mobile device with cloud computing environment so that computational efficiency may be achieved.

A. Algorithm

For, the task offloading, we propose a task offloading algorithm for effective mashing up the services. For the designing of the algorithm, various different parameters are considered like device computational power, Task Load and Task Type.

- The *Device Computation Power* is the processor computing speed and the number of processors present in mobile device.
- The *Task Load* is the degree of computations have to be done in order to complete the task. This is fetched from the Image resolution. We had settled various Thresholds for this task classification.
- The *Task Type* is the type of task, we are going to perform. Gray-Scaling and Histogram Equalization are two basic tasks that are considered for this experiments, where the gray-scaling is the LESS load task and Histogram Equalization is of MEDIUM load task.

Algorithm 1, is a mashing up task offloading algorithm, to decide that whether to process task over mobile over the cloud.

Algorithm 1: TaskOffLoad(IMAGE, TASK)

```
PROCESS_LOCATION := NULL;

if DeviceComputationalPower is LOW then
  if TaskLoad is LOW then
    if TaskType is GRAYSCALE then
      PROCESS_LOCATION := MOBILE;
    else
      PROCESS_LOCATION := CLOUD;
    [end if]
  else
    PROCESS_LOCATION := CLOUD;
    [end if]

else if DeviceComputationalPower is MEDIUM then
  if TaskLoad is LOW then
    PROCESS_LOCATION := MOBILE;
  else if TaskLoad is MEDIUM then
    if TaskType is GRAYSCALE then
      PROCESS_LOCATION := MOBILE;
    else
      PROCESS_LOCATION := CLOUD;
    [end if]
  [end if]

else if DeviceComputationalPower is HIGH then
  if TaskLoad is LOW or MEDIUM then
    PROCESS_LOCATION := MOBILE;
  else if TaskLoad is HIGH then
    if TaskType is GRAYSCALE then
      PROCESS_LOCATION := MOBILE;
    else
      PROCESS_LOCATION := CLOUD;
    [end if]
  [end if]
[end if]

if PROCESS_LOCATION = MOBILE then
  Compute over Mobile
else
  Offload Task to Cloud
[end if]
```

B. Tasks

To evaluate the proposed algorithm, we had taken 2 very basic imaging tasks, one is digital image gray-scaling and the other is colored digital image histogram equalization. We considered the gray-scaling as a task of LESS load and colored digital image histogram equalization as MEDIUM load task.

1) Gray scale

In photography and computing, a grayscale digital image is an image in which value of each pixel is a single sample. It carries only one intensity information also known as black and white image. It is composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest [8]. There are many ways to convert and RGB image to grayscale, we had used averaging method for the conversion.

2) Histogram Equalization

Histogram equalization is a technique for adjusting image intensities to enhance contrast. It is not necessary that contrast will always be increase in this. There may be some cases were histogram equalization can be worse. In that cases the contrast is decreased. Basically, this technique applies to grayscale images, but we can apply this method over colored images by applying it individually to all RGB values of the image [9].

C. Evaluation metrics

1) Response Time

Response time is the total amount of time the algorithm takes to respond and process the request for service [10]. In this proposed work, we are calculating the response time (1) When the task is processed over mobile device and (2) When the task is processed over cloud. The Response time is total process time in case of mobile device and in case of offloaded task, the Response time is sum of task offload time, process time, download processed data and overheads. Overheads is small time gap before of after some task e.g.: Connection Setup and termination time etc.

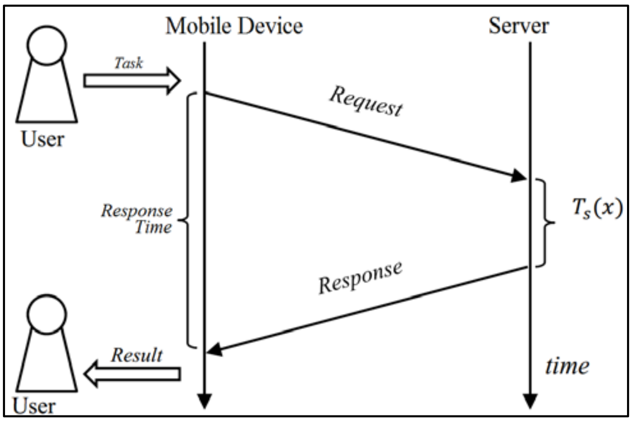

Figure 2. Response time.

Figure 2, demonstrates the response time of the offloaded task and $T_s(x)$ as task computation or process time.

$$ResponseTime = RequestTime + T_s(x) + ResponseTime + Overheads \quad (1)$$

Here, The Request Time is basically a time taken to upload the task to the cloud. The $T_s(x)$ is the process time, as already defined. The Response Time is the time taken to download the processed file. The overhead is the extra time consumption for time paddings, connection establishments and terminations etc.

2) Power Consumption

As mobile devices are limited powered devices, so energy consumption is an important evaluation metric. We had evaluated the power consumption on the basis of [11], and detailed as under:

a) The power consumption, when the computations are carried over mobile is estimated as:

$$E_{local} = P_{comp} * t_{mob}^{j} \quad (2)$$

Where, E_{local} is the mobile device energy consumption when the task is executed locally. P_{comp} is power consumption by mobile CPU per second and t_{mob}^{j} is the total execution time of the task.

b) The power consumption, when the task is offloaded to the cloud. The cloud application server runs on a continuous source of energy, so we only consider the power consumption of the device while data transmission and the CPU idle time, while the data is being computed over the cloud.

$$E_{cloud} = \left(P_{basic} * t_{cloud}^{i}\right) + E_{nic} \quad (3)$$

Where, E_{cloud} is the energy consumption of the device for data transmission and the energy consumption for the time the mobile CPU is idle and the Cloud server is computing. P_{basic} is the per sec mobile CPU energy consumption while it is idle. t_{cloud}^{j} is the mobile CPU idle time. E_{nic} is the energy consumption for the transmission of the task data to the cloud and download the processed task data. Here, in this work, we had only used Wi-Fi network for the task offloading. The E_{nic} here is E_{nic}^{wifi} and is estimated as:

$$E_{nic}^{wifi} = E_e + n.E_{trans} \quad (4)$$

Where, E_{nic}^{wifi} is energy consumption of mobile device while data transmission over Wi-Fi network. E_e is connection establishment energy and E_{trans} is the energy consumption while receiving and transmitting n bytes.

IV. Experimental Setting

For the Experiment, we had taken 5 similar images of different resolutions as 640x480, 1280x960, 1600x1200, 2048x1536 and 3264x2448 and we had considered 640x480 as LOW task load, 1280x960, 1600x1200 as MEDIUM task load and 2048x1536, 3264x2448 as HIGH task load.

We had also used different devices as LOW, MEDIUM and HIGH powered devices, we had taken:

- *High powered device:* Samsung Galaxy Note II, whose configuration includes Quadcore 1.6 Ghz Cortex A9 processor and Mali-400MP4 GPU Processor.
- *Medium powered device:* Samsung Tab 2, Whose configuration includes Dual core 1Ghz processor and PowerVR SGX540 GPU processor.
- *Low powered device:* Nexus One, with ARM armeabi-v7a 600Mhz processor without GPU processor.

One more parameter is also considered for algorithm designing is Task Type. We had taken 2 types of tasks as Digital Image Gray scaling as LESS load task and Image Histogram Equalization as MEDIUM load task.

The system that is used as cloud for evaluation is HP Pavilion dv4-1241tx, whose configuration includes Windows 7 Operating System, Intel Core 2 Duo CPU with 2.00GHz 2.00GHz clock rate processors, with 3.00 GB of RAM and 320 GB of Hard Drive. The processing time may vary on varying the system.

V. RESULTS

The Experiments are carried over already mentioned devices in previous section and results are estimated using various evaluation metrics as:

A. *Response Time*

Figure 3, demonstrated the time taken by the different powered mobile device to process the task over the mobile locally.

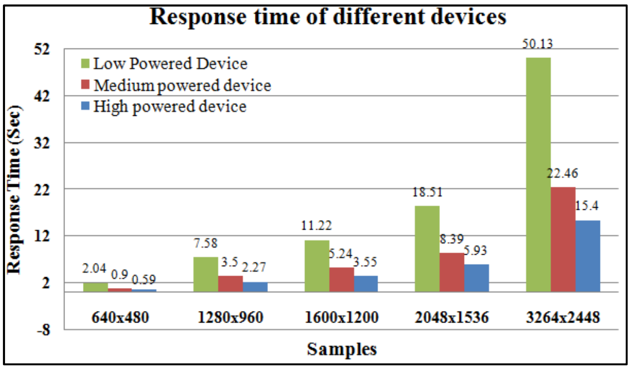

Figure 3. Response time of different devices over different image samples with Gray-Scaling as task.

It is depicted from the graph that the problem is not only limited to the device computation power, but also to the task load and task type. Figure 4, demonstrates the response time, when the Histogram Equalization task is carried over mobile devices.

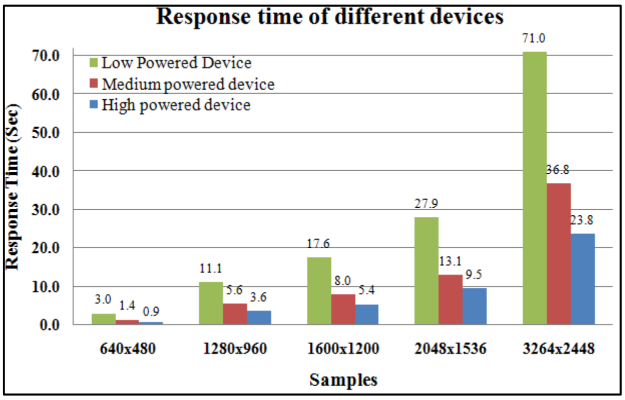

Figure 4. Response time of different devices over different image samples with Histogram Equalization as task.

From the Graph in Figure 4, it is clear that the response time is less for low resolution images and more for high resolution images. It is also depicted that Histogram Equalization task consumes more that than Gray-Scaling task.

So, to save this huge time consumption, one solution is to offload all tasks to the cloud. Figure 5 and Table 1, demonstrates the task upload time, process time, download time, overheads and response time, when the task is offloaded to the cloud.

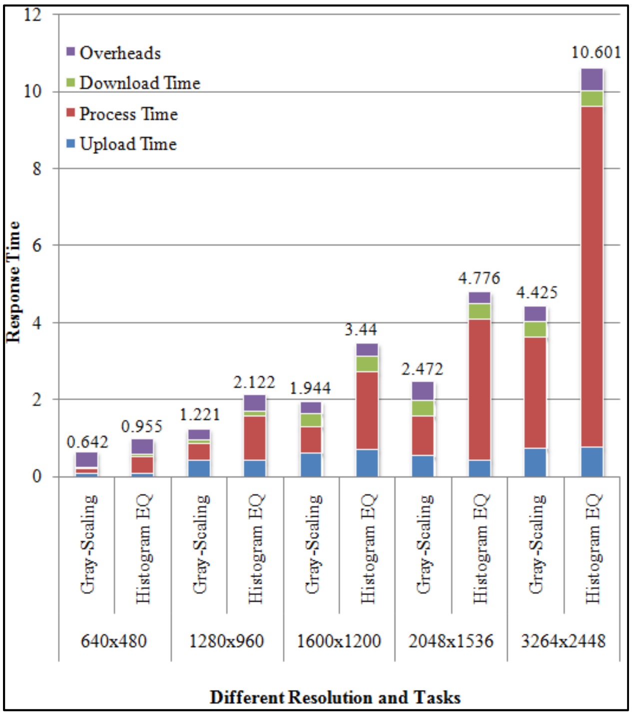

Figure 5. Response time of different Offloaded tasks.

TABLE I. TASK UPLOAD TIME, PROCESS TIME, DOWNLOAD TIME, OVERHEADS AND RESPONSE TIME WHILE OFFLOADING THE TASK TO THE CLOUD.

Image Resolution	Task	Upload Time	Process Time	Download Time	Overheads	Response Time
640 x 480	Gray-scaling	0.072	0.121	0.046	0.403	0.642
640 x 480	Histogram EQ	0.080	0.422	0.055	0.398	0.955
1280 x 960	Gray-scaling	0.396	0.436	0.114	0.275	1.221
1280 x 960	Histogram EQ	0.398	1.175	0.112	0.437	2.122
1600 x 1200	Gray-scaling	0.604	0.685	0.342	0.313	1.944
1600 x 1200	Histogram EQ	0.681	2.016	0.425	0.318	3.440
2048 x 1536	Gray-scaling	0.521	1.036	0.396	0.519	2.472
2048 x 1536	Histogram EQ	0.421	3.647	0.398	0.310	4.776
3264 x 2448	Gray-scaling	0.721	2.879	0.425	0.400	4.425
3264 x 2448	Histogram EQ	0.761	8.841	0.399	0.600	10.601

On the basis of both of the techniques described in Figure 3 Figure 4 and 5 we had designed a Mashup Algorithm detailed in Algorithm 1 that basically does a trade off for processing between the mobile computational device and cloud computational device and the evaluated results are as:

TABLE II. PROPOSED ALGORITHM - TASK OFFLOADING DETAILS.

Image Resolution	Task	Device Computational Power		
		LOW	MEDIUM	HIGH
640 x 480	Gray-scaling	Mobile	Mobile	Mobile
640 x 480	Histogram EQ	*Offload*	Mobile	Mobile
1280 x 960	Gray-scaling	*Offload*	Mobile	Mobile
1280 x 960	Histogram EQ	*Offload*	*Offload*	Mobile
1600 x 1200	Gray-scaling	*Offload*	Mobile	Mobile
1600 x 1200	Histogram EQ	*Offload*	*Offload*	Mobile
2048 x 1536	Gray-scaling	*Offload*	*Offload*	Mobile
2048 x 1536	Histogram EQ	*Offload*	*Offload*	*Offload*
3264 x 2448	Gray-scaling	*Offload*	*Offload*	Mobile
3264 x 2448	Histogram EQ	*Offload*	*Offload*	*Offload*

Table II, shows the results of the tasks that are offloaded to the cloud and the task that are processed locally. Table III, demonstrated the response time using the proposed Mashup algorithm.

TABLE III. RESPONSE TIME USING PROPOSED MASHUP ALGORITHM.

Image Resolution	Task	Device Computational Power		
		LOW	MEDIUM	HIGH
640 x 480	Gray-scaling	2.043	0.905	0.576
640 x 480	Histogram EQ	0.955	1.398	0.892
1280 x 960	Gray-scaling	1.221	3.508	2.273
1280 x 960	Histogram EQ	2.122	2.122	3.617
1600 x 1200	Gray-scaling	1.944	5.241	3.558
1600 x 1200	Histogram EQ	3.440	3.440	5.382
2048 x 1536	Gray-scaling	2.472	2.472	5.930
2048 x 1536	Histogram EQ	4.776	4.776	4.776
3264 x 2448	Gray-scaling	4.425	4.425	15.436
3264 x 2448	Histogram EQ	10.601	10.601	10.601

B. Power Consumption

As the mobile devices have very limited power, Table IV demonstrates the energy consumption for differed devices, when the task is processed over mobile device.

TABLE IV. POWER CONSUMPTION OF MOBILE DEVICE WHEN A TASK IS CARRIED OVER MOBILE DEVICE.

Image Resolution	Task	Device Computational Power		
		LOW	MEDIUM	HIGH
640 x 480	Gray-scaling	0.80	0.98	1.97
640 x 480	Histogram EQ	1.20	1.50	2.88
1280 x 960	Gray-scaling	3.07	3.82	7.30
1280 x 960	Histogram EQ	4.88	6.06	10.72
1600 x 1200	Gray-scaling	4.80	5.70	10.81
1600 x 1200	Histogram EQ	7.27	8.67	16.93
2048 x 1536	Gray-scaling	8.01	9.13	17.83
2048 x 1536	Histogram EQ	12.89	14.22	26.89
3264 x 2448	Gray-scaling	20.84	24.44	48.28
3264 x 2448	Histogram EQ	32.08	40.04	68.34

It is depicting form the table that the energy consumption is increases when we increase the images resolution, and it also increased when we move on towards LOW powered devices from HIGH powered Devices. Table V, demonstrates the energy consumption of the mobile device in transmission of the data and CPU idle time, when the task is offloaded to the cloud.

TABLE V. POWER CONSUMPTION OF MOBILE DEVICE WHEN A TASK IS OFFLOADED TO THE CLOUD.

Image Resolution	Task Type	Image Size (KB)	Energy Consumption for data Transmission (J)	CPU idle Time (Sec)	CPU idle Time Energy consumption (J)	Total Energy consumption (J)
640 x 480	Gray-scaling	69	3.383	0.121	0.1089	3.4919
640 x 480	Histogram EQ	69	3.383	0.422	0.3798	3.7628
1280 x 960	Gray-scaling	223	4.461	0.436	0.3924	4.8534
1280 x 960	Histogram EQ	223	4.461	1.175	1.0575	5.5185
1600 x 1200	Gray-scaling	327	5.189	0.685	0.6165	5.8055
1600 x 1200	Histogram EQ	327	5.189	2.016	1.8144	7.0034
2048 x 1536	Gray-scaling	504	6.428	1.036	0.9324	7.3604
2048 x 1536	Histogram EQ	504	6.428	3.647	3.2823	9.7103
3264 x 2448	Gray-scaling	1041	10.187	2.879	2.5911	12.7781
3264 x 2448	Histogram EQ	1041	10.187	8.841	7.9569	18.1439

The above table depicts that the energy consumption of mobile device, when the task is offloaded to the cloud. The energy of the mobile device is consumed in data transmission to and from the cloud and energy consumption of mobile CPU in idle state. The Data transmission energy consumption is increasing with the increase of the image resolution. Figure 6, shows the comparison of the energy consumption, when the task is carries over mobile device and when the task is carried over the cloud.

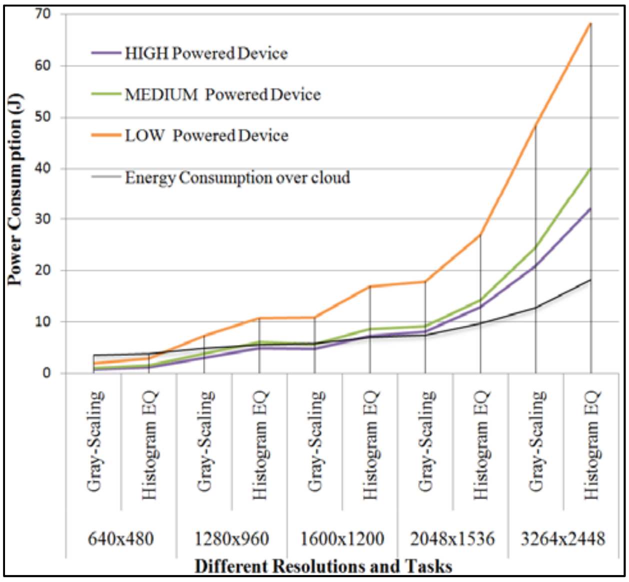

Figure 6. Power consumption of mobile device when a task is carried over mobile device.

From the graph in Figure 6, The black line demonstrates the energy consumption of the mobile device, which is far less than the energy consumption, when the task is carried over mobile device.

VI. CONCLUSION AND FUTURE SCOPE

We had considered the computational power of mobile device, Task load, Task type for the designing of task offloading algorithm and came to the conclusion that the proposed algorithm for task off loading is working better that the existing computation system in terms of the system response time and the energy consumption. The outcome of this work is a software prototype, that demonstrates the working of the proposed algorithm. The proposed algorithm is performing very well but their is further a room of improvement by using some enhanced approaches for better task offloading mechanism.

REFERENCES

[1] Statista, "Smartphone users worldwide 2012-2018 — statistic," date fetched: 14 June, 2015 http://www.statista.com/statistics/330695/number- of- smartphone- users- worldwide/.

[2] H. J. La, S. W. Choi, and S. D. Kim, "Technical challenges and solution space for developing SAAS and mash-up cloud services," in *e- Business Engineering, 2009. ICEBE'09. IEEE International Conference on*. IEEE, 2009, pp. 359–364.

[3] E. M. Maximilien, A. Ranabahu, and K. Gomadam, "An online platform for web apis and service mashups," *Internet Computing, IEEE*, vol. 12, no. 5, pp. 32–43, 2008.

[4] C. M. S. Magurawalage, K. Yang, L. Hu, and J. Zhang, "Energy-efficient and network-aware offloading algorithm for mobile cloud computing," *Computer Networks*, vol. 74, pp. 22–33, 2014.

[5] J. Luzuriaga, J. C. Cano, C. Calafate, and P. Manzoni, "Evaluating computation offloading trade-offs in mobile cloud computing: A sample application," in *CLOUD COMPUTING 2013, The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization*, Think- mind, Ed., May 27, 2013 2013, pp. 138–143.

[6] H. Qi and A. Gani, "Research on mobile cloud computing: Review, trend and perspectives," in *Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on*. IEEE, 2012, pp. 195–202.

[7] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[8] S. Johnson, *Stephen Johnson on digital photography*. O'Reilly Media, Inc., 2006.

[9] M. Abdullah-Al-Wadud, M. H. Kabir, M. Dewan, and O. Chae, "A dynamic histogram equalization for image contrast enhancement," *Consumer Electronics, IEEE Transactions on*, vol. 53, no. 2, pp. 593–600, 2007.

[10] M. Alhamad, T. Dillon, C. Wu, and E. Chang, "Response time for cloud computing providers," in *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*. ACM, 2010, pp. 603–606.

[11] N. Balasubramanian, 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 conference*. ACM, 2009, pp. 280–293.

Kuljeet Kaur, has received her masters degree from GNDU University in 2015. Her research interest includes Mobile computing, cloud computing, wireless sensor networks. She had several publications in reputed journals and in international conferences.

Kamaljot Singh, Assistant professor at Lovely Professional University, INDIA has received his master degree from DAV University in 2015. He had received a Gold medal from NASA, Johnson Space center in 2008 for rover designing. His research interest includes data mining, wireless sensor networks, natural language processing, Nano-magnetic materials and nature inspired computations. He had several publications in reputed journals and in international conferences.