**International Journal of Technical Research and Applications** e-ISSN: 2320-8163,
www.ijtra.com Volume 4, Issue 3 (May-June, 2016), PP. 272-275

# AN IMPROVED SCHEDULING ALGORITHM FOR TASK OFFLOADING ON CLOUD

**Madhurika A. Joshi[1], Aditya U. Tornekar[2], Rahul K. Sawkar[3], Akash A. Kandarkar[4]**
Computer Department
[1,2,3,4]JSPM's ICOER,
Pune, India
[1]madhu2264@gmail.com,
[2]adityatornekar@gmail.com,
[3]rahulsawkar29@gmail.com,
[4]akash.kandarkar926@gmail.com

***Abstract*— Nowadays mobile devices have become one of the basic necessities of our daily lives. These mobile devices resolve our numerous tasks which earlier used to take a lot of time without such devices. This paper is used to describe some of the methodologies using which we can manage such tasks more efficiently. The basic disadvantage of mobile devices is its battery consumption, due to which we need to use our devices in a much optimized way. To support this idea, we have proposed an organizational level application through which we can segregate between tasks which have much higher battery and resource utilization and tasks which require minimum resources. The former tasks are dispatched or rather offloaded to the cloud servers and the generated output is sent back to the device which reduces the battery consumption of the device. To distinguish between the tasks we use some scheduling algorithms at both levels.**

***Index Terms*— Task offloading, Cloud computing, Scheduling algorithm**

## I. INTRODUCTION

Today most of our daily life tasks are completed using mobile devices. Mobile devices come with various utilities which minimize our physical work load. As our lives have become an online grid through which we like to stay connected to our mobile devices for every aspect of our life but we always get hooked up to our chargers as battery consumption of these mobile devices has reached a whole new level. Phones with large battery outputs suffer overheating issues, to overcome these hindrances we have proposed an organizational level system where people can use their mobile devices in a much optimized way. In this system, we try and transfer some tasks which might have heavy battery and resource utilization. If such tasks are offloaded to the cloud then we can process these tasks using the cloud services which have a much higher computational efficiency. The output generated can be sent back to the device. Thus, reducing the battery consumption of the device and also increasing the efficiency of the device.

In this paper we have discussed how to offload heavy tasks to the cloud. Cloud can be divided in to three primary types: Public, private and hybrid cloud. These clouds come with various services such as PaaS (Platform as a Service), IaaS (Infrastructure as a Service) and SaaS (Software as a Service).This paper further describes about the scheduling algorithms used in the system.

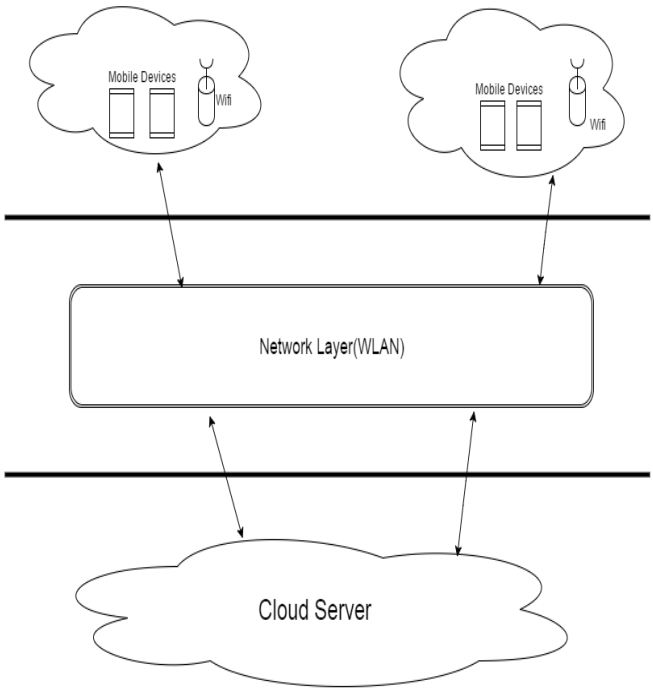
This system has a lot of scope in the future, as some more modules will be added to the existing system and improvised functionalities will remove all the loopholes in the system.

## II. ARCHITECTURE

At the user level, every mobile device has an application specially designed for offloading tasks to the cloud. The application has a list of tasks such as sending an email, comparing images, EMI calculator, etc. Among these tasks few are migrated to the cloud for their execution whereas other tasks which consume less power are executed on the mobile device itself. The user needs to first register to the server to use these services.

The network level is simply a wireless network which connects the server that is the cloud and our mobile devices so that the devices can interact with the cloud. To offload a task the mobile devices should always be connected in the same network.

The server level consists of cloud which is used to receive the tasks from the user level and processing of these tasks is done using cloud services. Then the generated output is sent back at the user level. This server level also consists of a database used to store some user level credentials and also the processing time.



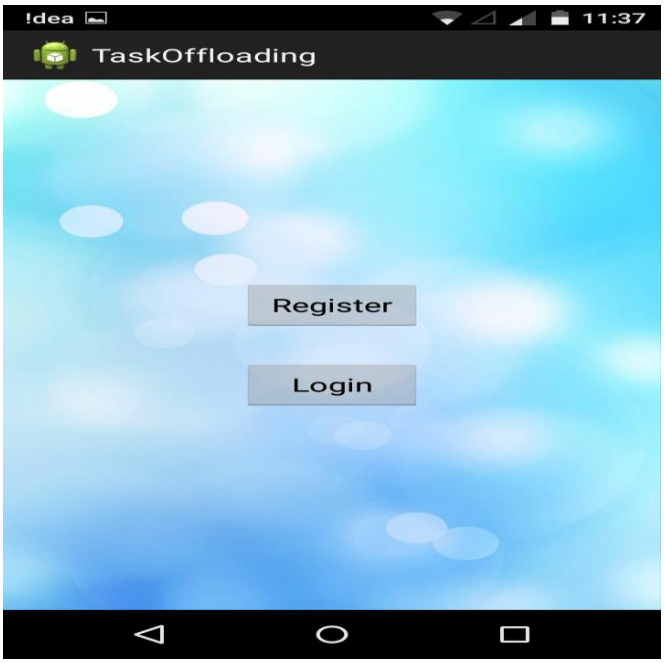

1. Architecture Diagram

## III. WORKING

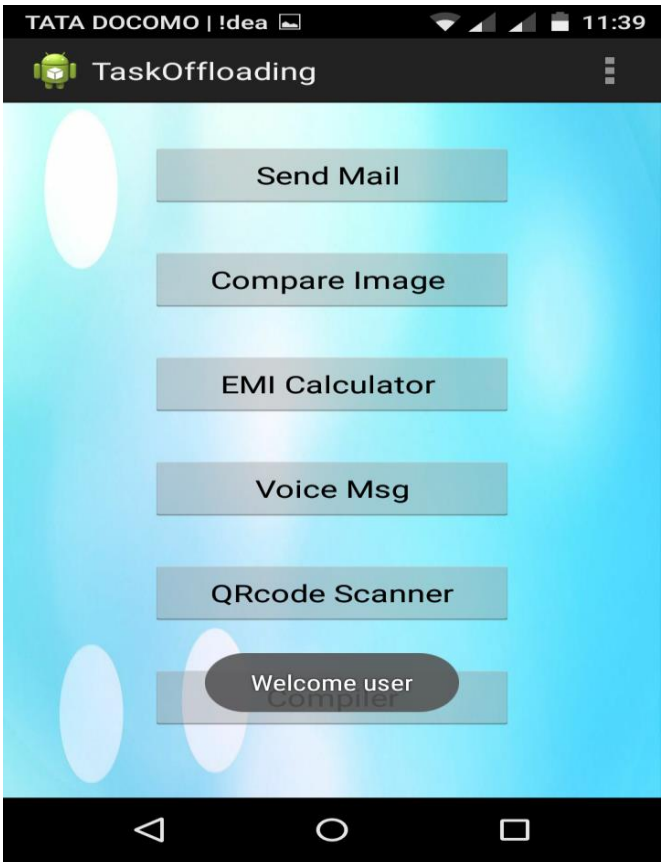
User needs to install an application, using this application user should register to the organization server. Then the login page pops up. Here, the user can login using the registered credentials to use the offloading functionality and cloud services. As the proposed system is an organizational level system user needs to stay connected to the organization network.

When the task is offloaded to the cloud and task hits the server we use the time function to get the current time Ts. When the computed output is dispatched back to the user's mobile device, once again same time function is used to get the time which is denoted by Te. The total time of execution is the difference between the two which is calculated as Tt=Te-Ts. This total time is used to assign priorities to tasks and these priorities are stored in the database. These stored priorities are used next time where tasks with higher execution time gets lower priority and tasks with lower execution time gets higher priority. $T_t = \{T_1, T_2, T_3, \dots, T_n\}$ where $T_t$ is pool of tasks ($T_1$ to $T_n$). Every task in the pool is assigned with a priority from high to low (1 to n) depending on the execution time taken.

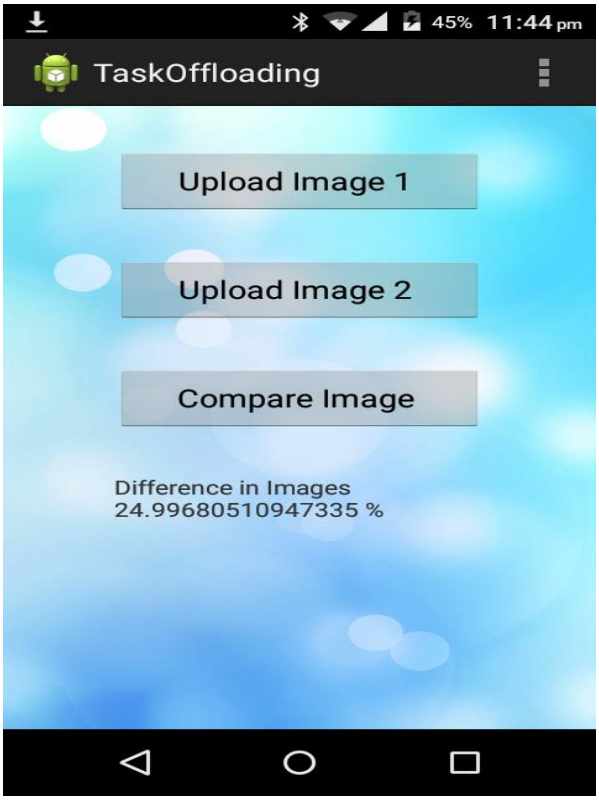
## IV. SCREENSHOTS



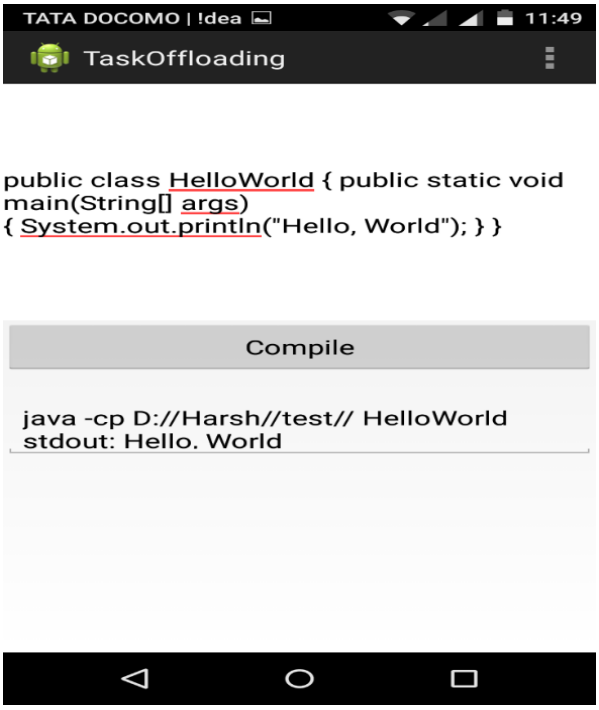

2. Login and Registration Screen




3. Task List




4. Compare image task using cloud server




5. Java compiler using cloud server

## V. ALGORITHM

1. Calculate time taken for all the tasks $T_t$ in the pool
2. Assign priority for each task
3. Execution of tasks according to priorities

For a task $T_n$
{
$T_{nt} = T_{ne} - T_{ns}$ ; // Calculating the total execution time

$T_{npri} = P_n$ ; // Assign priority to the task

if ($T_{npri} = 1$)
{

Execute task $T_n$ ;

}
else
{
Execute task ($T_1$ to $T_{n-1}$) ;
}
}

4. Exit

where,
$T_n$ is a task from the pool
$P_n$ is the priority assigned to the task
$T_{npri}$ is the priority of task $T_n$
$T_{ns}$ and $T_{ne}$ are the start and end time of task $T_n$

## VI. FUTURE SCOPE

The system is an organizational level system which means it has limitation in its range of usage. The range can be extended to higher levels like cities or states. We can extend the list of applications or tasks to be offloaded from time to time as per requirements. This functionality has been developed only for android mobile devices, it can be further developed for iOS based devices like iPhones, iPad etc. and also Windows based mobile devices. More scheduling criterion can be applied for much optimized results.

## VII. ACKNOWLEDGEMENT

This research would not have been feasible without encouragement and guidance of *Prof. Devendra P. Gadekar*. We are heavily indebted to him. He patiently discussed ideas with us and gave us scheduling ideas for the project. We are grateful to *Prof. S. R. Todmal*, Head of Computer Department, Imperial College of Engineering and Research for always being ready to help with most diverse problem that we have encountered along the way. We express our sincere thanks to all staff and colleagues who have helped us directly or indirectly in completing this paper. We pay our respects and

love to our parents and all other family members and friends for their love and encouragement throughout our career.

REFERENCES

[1] Z. Zhu, B. Sun, X. Li, X. Zhou, "Application-aware Group Scheduler for Android".
[2] M. Qiu, Z. Chen, L. T. Yang, X. Qin, B. Wang, "Towards Power-Efficient Smartphones by Energy-Aware Dynamic Task Scheduling".
[3] S. C. S. Chen, S. Weng, C. Li, "Design a Low-Power Scheduling Mechanism for a Multicore Android System".
[4] M. Altamimi, "A Task Offloading Framework for Energy Saving on Mobile Devices using Cloud Computing".
[5] S. E. Chang, K. Chiu, "Cloud Migration: Planning Guidelines and Execution Framework" 978-1-4799-8993-5/15/$31.00 ©2015 IEEE.
[6] V. Andrikopoulos, T. Binz, F. Leymann, and S. Strauch, "How to Adapt Applications for the Cloud Environment," Computing, vol. 95, no. 6, pp.493–535, Dec. 2012.
[7] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud Migration Research: A Systematic Review," IEEE Trans. Cloud Compute., vol. 1, no. 2, pp.142– 157, 2013.
[8] T. Binz, F. Leymann, and D. Schumm, "C Motion: A Framework for Migration of Applications into and between Clouds," 2011 IEEE Int. Conf. Serv. Compute. Appl. (SOCA), pp.1–4, 2011.
[9] S. Vahora, R. Patel, H. Patel, S. Patel, "Efficient Virtual Machine Management Based on Dynamic Workload in Cloud Computing Environment".
[10] M. Altamimi, A. Abdrabou, K. Naik, A. Nayak, "Energy Cost Models of Smartphones for Task Offloading to the Cloud".
[11] L. Gkatzikis, I. Koutsopoulos, "Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems".
[12] T. T. Hum, C. K. Tham , D. Niyato, "To Offload or to Wait: An Opportunistic Offloading Algorithm for Parallel Tasks in a Mobile Cloud" 978-1-4799-4093-6/14 © 2014 IEEE DOI 10.1109/CloudCom.2014.33.
[13] D. Kovachev, R. Klamma, "Framework for Computation Offloading in Mobile Cloud Computing".
[14] Y. Z., H. Liu , L. Jiao, X. Fu, "To offload or not to offload: an efficient code partition algorithm for mobile cloud computing".