Efficient use of available network in mobile for executing mobile application with real time constraints.

A.D.Saraf [1],Prof Uma Thakur [2], Prof L.H.Patil [3]
[1].Department of Computer Science and Engineering
Mtech Sudent PIET
Nagpur, India
akshaydsaraf@gmail.com
[2,3] Department of Computer Science and Engineering
Asst Professor PIET
Nagpur, India

***Abstract*— In today's world mobile fulfill most of computational needs of average persons in form of different applications. As smart phones works in multiprocessing mode, no of applications gets executed simultaneously executed in mobile device. Though mobile configurations are improving in (battery, processing speed) it still needs additional computing speed to run apps with real time constraints. Thus to fulfill computational needs of real time mobile applications such as augmented reality , some part of execution can be shared with cloud having no of cloud service provider such as Amazon Cloud, Windows Azure. Mobiles can also use machines and other mobile devices such as laptops, smart phones with higher configurations of ram and processor, tablets etc for transferring execution load of applications running on mobile using different wireless connections such as wifi network or blue tooth network. Mobile must able to utilize all available networks in an efficient way to provide best solution in running augmented reality application. In this paper we are providing modified design of Adhoc networks for providing better computation efficiency as well as terms of services in augmented reality application.**

I.INTRODUCTION

As the number of applications are ready to get executed on mobile day by day[12], it is covering the possible needs of other computational devices such as laptops, desktops, notebooks etc. some of mobile application needs higher computational power in spite of descent computational power available in mobile handset.

Solution to the above problem is to offload part of application to clouds. One way is to offload parts of application in such way that overall latency generated should be minimum. As algorithm developed in [3] application can be subdivided in no of subparts taking into account of network latency, computational latency. As described in [6] think-air provides efficient way of data transfer while offloading code between cloud server and mobile device assuming precondition of having secure data transfer between host and cloud server. To enhance overall performance and to reduce latency in [7] author developed CRAM (Cloud resource allocation for mobile application) algorithm which works on two layer architecture. Such approach suffers with high network latencies[11] and require ample amount signal strength(i.e require more battery power). Another approach is to execute the parts of application by resources available in the local network to mobile devices such as AdHoc networks. As described in [10] Offloading done on the method level provides better results in terms of energy conservation than that of offloading done of package level. Traditional AdHoc network produce computational latencies as each time the offloading goes through networks server to other devices available in network. Thus any device in the network having offloading will first transfer data to server in turn server will allocate devices and will transfer data to allocated machine for processing which increases overall execution of application. Also as described in[8] As no of hops increases more two in AdHoc network optimized performance is missed by AdHoc network.

In this paper we propose a design in AdHoc structure as presented in [2]. We also propose mobile services to compute computational power borrowed by system.

In section 2 we have proposed AdHoc network design and its details. In section 3 we can find different proposed policies to design in section 2 . In section 4 we walk through related work in the given field. Finally in section 5 we conclude paper with future work.

II.PROPOSED DESIGN

As contrast to hierarchical structure in AdHoc network design each device having an offload to other device will form AdHoc network. Thus peer to peer programming comes in handy where each device can work as client as well as server as needed in program execution. Such structure provides minimum time delays and produces

correct output. In absence of Ad-hoc network mobile computes latency for transferring live data in form of stream over cloud. Depending upon the latency mobile share some part of program execution on cloud in such a way that overcomes time bound constraints placed by mobile application. For getting best output in augmented reality applications latencies for each available network for transferring live data from mobile to servers checked periodically. Considering a scenario where a mobile device have available local network with available cloud service, mobile peeks local network for offloading part of application than selecting cloud for offload. As given in figure 1 device having offloading will search other devices in its own Ad-hoc

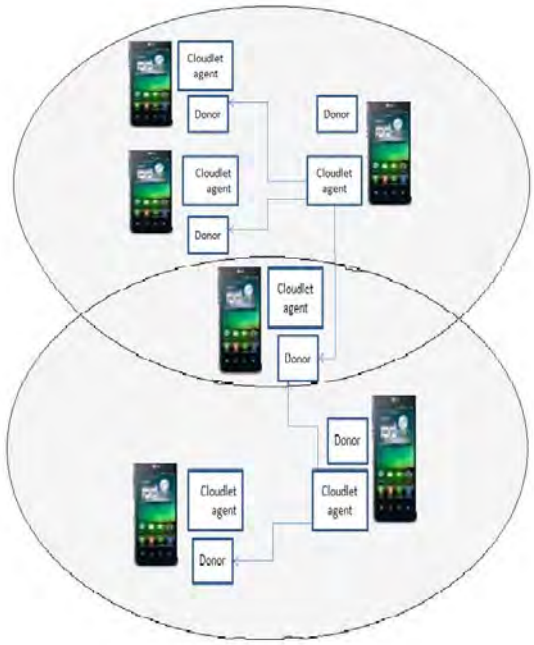

Figure1: AdHoc network Design

network with the help of hotspot, Bluetooth associated with system. After successful detection of resources the device will become AdHoc network agent and will govern decision making and scheduling techniques for offloading and result gathering.

A device participating as donor can donate competency to more than one device at time. In contrast the AdHoc network agent(acceptor)will not participate in any resource sharing activity.

After successful execution of application program in offloading device the AdHoc network is dissolve by AdHoc network agent. AdHoc network agent seeks the configuration, processor status, offloading latency(assumed to be minimum as devices are local to offloading device) of systems available in its network and computes the time required to execute the offloaded parts. According distribution of components in application is performed.

III.PROPOSED POLICIES

At staring of application firstly standard latency for each local and cloud are considered. As execution of an application advances latencies for cloud and Ad-hoc network computed periodically and following steps are taken 1) Firstly we Compute the total time local(ttl) required by system to execute application on device with help of available resources(processing speed and memory).

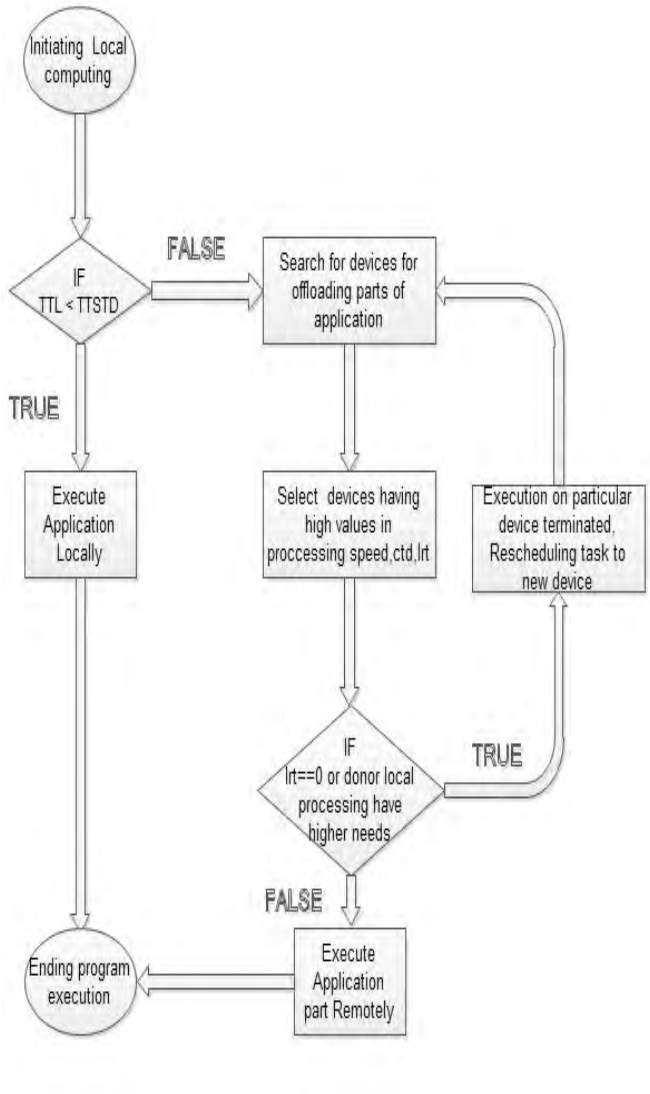

Figure2: Flow control of offloading.

2) If computed times(ttl)is less than total time standard (ttstd) then application gets executed locally else other devices are searched through blue tooth, hot-spot associated with device.

3)As task completion probability is higher in systems having higher processing speeds, such devices gets selected for offloading generating better performance . Thus system having better processing power, higher credential total

of device(ctd), and lower load remaining time(lrt) gets selected.

4)In each offloading few points gets awarded to each device which participate as donor. Points awarding value depends on efficient execution of task performed by donor. The value becomes 1 if task given to donor gets completed in time else the value degrades according to extra time required to complete offloaded task.

5) As mobile executes task assigned to it, some amount of power is lost by mobile. Mobile must reserve some predefined amount of power for executing regular operation in mobile. Thus local reserve time indicates the power in remaining time mobile can allocate can allocate to execute task given by other mobile devices.

IV.CURRENT EXECUTION

In our paper we are using augmented reality application for demonstrating real time mobile constraint. In a augmented reality application an image is search thru mobile camera. When camera previews the target image, event associated with target images gets executed. For current implementation we have taken 10 target images and on target image detection we are rendering 3D images for each target respectively. As Android is capturing most of market we have developed our application in android. We have used OPEN Gl for generation of 3D

images . The native codes are used for rendering 3D images which are integrated with our android application. Java remote sensing technique is used to performing cross platform connectivity.

In our execution we are taking reading help of two mobiles

Sony Experia tupo dual with processing speed of 800 Mz , 2 GB Ram and Samsung Galaxy with processing speed of 1 GHz and 2GB Ram.

App Execution	Image1	Image2
Sony Experia		
1) Mobile	340 ms	359 ms
2) Cloud	290 ms	314 ms
3) Adhoc Network	110 ms	123 ms
Samsung Galaxy		
1) Mobile	298 ms	310 ms
2) Cloud	279 ms	340 ms
3) Adhoc Network	101 ms	118 ms

V.RELATED WORK

Outsourcing the resources in form of Processing and memory is termed as cyber forging [4]. System providing resources are called donors.

Tim Verbelen, Tim Stevens, Filip De Turck, Bart Dhoedt in graph partitioning algorithm [1]discussed details about using mobile apps with cloud in context with graph partitioning to partition application in efficient way for minimizing total time required to execute application. The above system has been implemented in java and AdHoc network framework.

Tim Verbelen in "AdHoc network bringing cloud to user"[2] have implemented a prototype of framework in Java, which allows it to run on most hardware such as laptops, desktops and servers, but also on Android, the most popular mobile operating system . Which generates resources in near vicinity of system.

Byung-Gon Chun,Sunghwan Ihm in CloneCloud: Elastic Execution between Mobile Device and Cloud [5] implemented prototype of CloneCloud partitioning and migration on the cupcake branch of the Android.

Muhammad Shiraz1, Abdullah Gani in Mobile Cloud Computing: Critical Analysis of Application Deployment in Virtual Machines[9] studies different scenarios where virtual machines gets allocated to satisfy clooudlet requirements. It also provides results in terms of variant use of resources which increases total execution time.

VI.CONCLUSION AND FUTURE WORK

In this paper we presented new design for AdHoc network, we also shown its helps in increasing computing efficiency by reducing the amount of time spend on offloading part of application to AdHoc network agent. We also discuss different policies for sharing resources, enabling efficient use of available system. In future we would like to work on different scheduling algorithms with parameters discussed in above paper.

ACKNOWLEDGMENTS

Our thanks to CSE dept PIET for providing continues background support .

REFERENCES

[1] Tim Verbelen, Tim Stevens, Filip De Turck, 2012. Graph partitioning algorithms for optimizing software deployment in mobile cloud computing.
[2] T. Verbelen, P. Simoens, F. De Turck, B. Dhoedt, MCS '12, 2012. AdHoc networks: Bringing the cloud to the mobile user, in: Proc. of the 3rd ACM Workshop on Mobile Cloud Computing & Services.
[3] Byung-Gon ,Chun,Petros Maniatis 2010 Dynamically Partitioning Applications between Weak Devices and Clouds.
[4] Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang 2011 A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches.
[5] Byung-Gon Chun,Sunghwan Ihm,Petros Maniatis,Mayur Naik,Ashwin Patti 2011 ACM , CloneCloud: Elastic Execution between Mobile Device and Cloud
[6] Sokol Kosta, Andrius Aucinas, Pan Hui ,Richard Mortier Xinwen Zhang 2011 Q2 earning call Google ThinkAir: Dynamic resource allocation and parallel execution in cloud for mobile code offloading.
[7] M. Reza Rahimi1, Nalini Venkatasubramanian 2012 ACM *MAP*Cloud: Mobile Applications on an Elastic and Scalable 2-Tier Cloud Architecture.
[8] Debessay Fesehaye, Yunlong Gao, Klara Nahrstedt Impact of AdHoc networks on Interactive Mobile Cloud Applications
[9] Muhammad Shiraz1, Abdullah Gani IPCSIT vol. 27 (2012*)* Mobile Cloud Computing: Critical Analysis of Application Deployment in Virtual Machines.
[10] Dejan Kovachev, Yiwei Cao and Ralf Klamma Information Systems & Database Technologies RWTH Aachen University 2012 Mobile Cloud Computing: A Comparison of Application Models.
[11] Juntunen, Antero, Kemppainen, Matti, Luukkainen, Sakari ICMB 2012 Mobile computation offloading factors affecting technology evolution.
[12] http://www.techno-pulse.com/