Making Mobile Devices Last Longer Through 3-tier Cyber Foraging Approach in Mobile Cloud Computing

Dhwani Sanghavi[1], Prof. Jignesh Vania[2]
[1]Masters of Computer Engineering, L. J. Institute of Engineering and Technology,
[2]Professor, L. J. Institute of Engineering and Technology,
[1, 2]Computer Engineering Department, Gujarat Technological University, Ahmedabad, Gujarat, India

Abstract –In **recent years, the mobile devices are becoming more favorable due to small size, portability, etc. Despite of these improvements on mobile devices such as laptops, smart phones, etc the battery is always a limiting factor. Cyber foraging is a technique in which the small mobile devices offload the resource consuming data or applications from resource constrained mobile devices to resourceful devices. In the mobile devices CPU depletes resources very rapidly. In this paper, we found out that and show that CPU utilization has decreased and performance has increased for the mobile devices by utilizing cyber foraging approach. Several issues of data offloading such as performance of device, privacy of data from cloud vendor, storage on cloud have been tackled. A cloudlet defined as a new framework in architecture to represent middle-tier in 3-tier hierarchy–(mobile device-cloudlet-cloud).Comparative study has been done between varied mobile and non mobile devices and results show that non mobile devices give greater performance. Also comparison between 3G and Wi-Fi has been done and found out that 3G gives poor performance as compared to Wi-Fi. Moreover, experiments have been performed by statically offloading different sized applications from resource constrained mobile devices to resourceful cloud front end in our framework and results show that it is better to offload huge sized apps on cloud as CPU utilization is high when executed on mobile devices as compared to cloud and thus the storage issue of device has been tackled off. Finally the results conclude that the proposed 3-tier framework of cyber foraging technique reduces the overheads of latency, CPU Utilization and thereby improves the mobile device performance.**

Index Terms**-Mobile Cloud Computing, smart phones, battery life, offloading, cloudlet**

I. INTRODUCTION

Cyber foraging is a pervasive computing technique where resource poor, mobile devices offload some of their heavy work to stronger surrogate machines in the vicinity.[1] As mobile applications as well as cloud computing grow immensely mobile cloud computing (MCC) is introduced as evergreen technology in recent world. MCC integrates cloud computing and the mobile environment as well as overcomes the various issues on the mobile computing side which area related to performance which includes battery life, storage, environment in mobile computing. Overview of the MCC including the definition, comparison of cloud and mobile cloud, its architecture, and brief applications.[2]Cyber foraging ameliorates this performance disparity by utilizing nearby non-mobile computers called surrogates to run the either the parts or whole applications, which are offloaded from the mobile devices[3].

A. MOBILE CLOUD COMPUTING

Mobile devices are achieving familiarity and the people around the world are enormously using the devices such as laptops,smartphones,etc and PDAs for running different applications on daily basis.

B. MOBILE CLOUD COMPUTING PLATFORMS

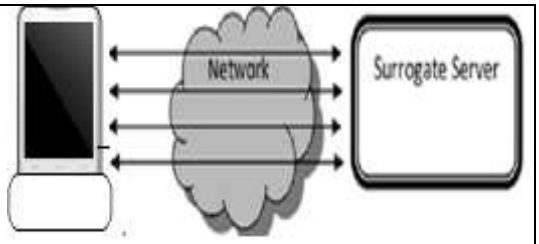
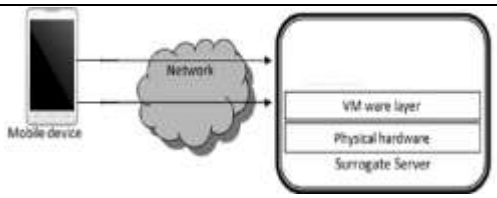
Google App Engine, the mobile cloud platform use the Client server model for enabling App Engine usage for different application servers on the computational nodes.[4]Amazon Elastic Compute Cloud (Amazon EC2) is the web service for providing the resizable compute capacity in cloud.[5]Mobile cloud computing is the combination of cloud computing and mobile network for bringing benefit to the cloud providers as well as the mobile users.The main goal of Mobile cloud computing is to enable the mobile rich applications to be executed onto the mobile devices, with user rich experience.It is defined as the rich mobile computing technology that leverage the resources of the cloud towards the storage and mobility to serve huge number of mobile devices anytime through Internet and anywhere regardless of the heterogeneous platforms as well as environments on the pay as per use principle.[6]

II. CYBER FORAGING

Users achieve benefit from mobile devices for using resource intensive applications.Cloud computing can be used to leverage the benefit for these mobile devices applications.Resource poverty create shortcomings in the quality of mobile devices. Large applications require huge CPU utilization,battery lifetime and high memory as compared to available resource constrained mobile devices.Therefore, the mobile devices remain more resource constrained devices than the traditional stationary computers.Cyber foraging is the computing technique in which mobile device offloads resource consuming tasks to

nearer surrogates for preserving the energy there by increasing its performance.[7]The greatest challenge for mobile devices is that they deliver low performance as compared to traditional non-mobile and less-constrained computers due to their self- consciousness in size, weight and mobility in spite of the advancements in todays world.[8]

III. OFFLOADING IN MOBILE CLOUD COMPUTING

Client-Server Based (RPC based)	 Fig 1:Client-Server Architecture E.g.: Maui[13],GAE Advantages: Trust and security are available easily, Portable ,well supported API Disadvantages: needs prior installation, no support for task migration
Virtual ization Based	 Fig 2:Virtualization based architecture Advantages: less code rewriting due to cloning Disadvantages: VM overlays must be compatible

A. TYPES OF OFFLOADING IN MOBILE CLOUD COMPUTING

Tab1: Comparisons of Offloading Techniques [2].

IV. PLATFORMS USED

A. GOOGLE APP ENGINE AND EC2

Google cloud platform has been used for deploying varied apps on the cloud and there after runs them over sandbox provided by GAE. In this research work, Google cloud platform is used for deploying the applications[4].EC2 instances are used for creating cloudlet architecture.

B. STEPS OF IMPLEMENTATION IN GAE(CLOUD BASED DEPLOYMENT)[20]

Basic Requirements

1. eclipse
2. GWT (google plugin) for eclipse

C. GETTING STARTED WITH GOOGLE APP ENGINE

>Sign up for an App Engine account
>Download the App Engine SDK
>Set up app using eclipse.
>Install a GWT Pluggin.
>Register an App.

1. In the console of App Engine type: https://appengine.google.com/
2. Sign up for App Engine using your Google account.
3. For creating a new application:
   a) Click Create an Application.
   b) Register using an application ID, unique to this application.
   c) Domain is appspot.com and its URL is http://*application-id*.appspot.com/

>For uploading from Eclipse.
Download GWT pluggin in eclipse.
For deploying application using Eclipse, click the App Engine deploy from the toolbar .
Give your previously entered application ID in web.xml file.Eclipse after gets\ting the application ID from the appengine-web.xml file, upload the contents of its war/directory.
> Accessing your Application.
Now your application has been deployed and is running on App Engine.

D. STEPS OF IMPLEMENTATION IN EC2(CLOUDLET CREATION)

1.Sign up for AWS by creating an Amazon Web Services Account.
2.Create a new EC2 VM instance in the desired region,.
3.Connect the server through SFTP port 22.
4.Upload App of same size on different instances.
5.Calculate the CPU Utilization for all Instances of server.

V. PROPOSED 3-TIER FRAMEWORK

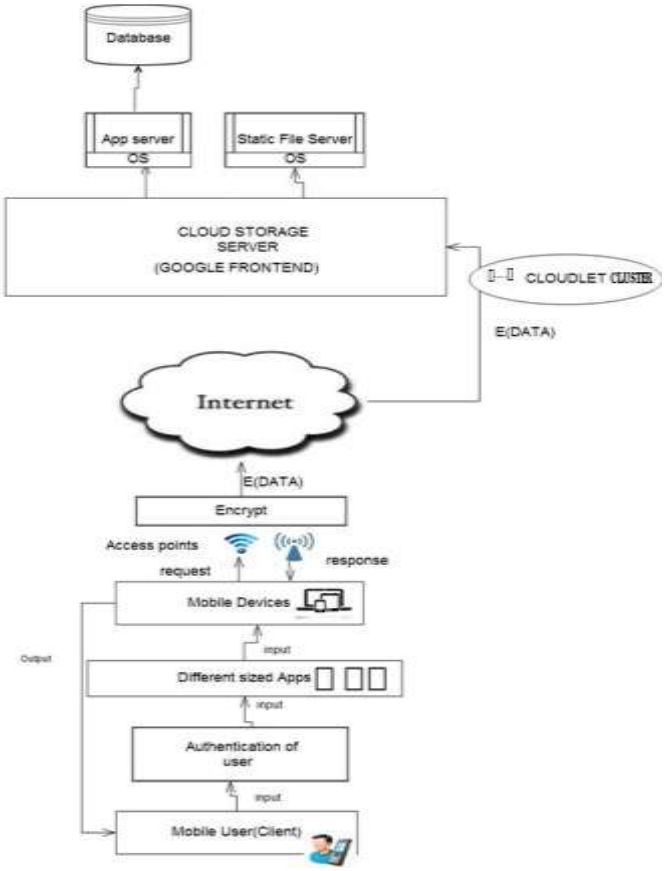


Fig 3:Flowchart of Proposed Architecture

A. ADVANTAGES AND DISADVANTAGES OF PROPOSED SOLUTION

Pros	Cons
• Increase the lifetime of battery of devices • Performance improvement • Capital investment is low initially. • Storage of data and easier disaster recovery • Privacy of the clients data from cloud vendor itself has be considered before data offloading.[9] • Trust of surrogate and authentication of user is also another issue tackled before offloading data onto the clouds.	• Heterogeneity of varied mobile devices like black berry, smart phones. • Scalability of varied sized applications • Access right to different users[10]

Tab 2:Pros and Cons of Proposed Solutions

B. PROPOSED PSEUDOCODE

Algorithm 1: Whether to offload or not

Calculate the latency for different sized applications and store it.
Calculate the latency for each datacenter and store it.

Input: cpu_utilization, datasize, latency
Output:offloading decision
Assumption:Static offload, no packet loss,static user,reliable connection

```
Offload_Algo ( )
{
        for each datasize i
        {
                if (cpu_utilization_i>min_cpu && latency_i<min)
                        {
                          Mobile_offload=true;
                          return true;
                        }
                        else
                        {
                          Mobile_offload=false;
                          return false;
                        }
        }
}
```

Proposed algorithm is implemented in Google app engine and offload decision has been got that there is no need to offload small sized application

Algorithm 2: Selecting the Best Hop to offload

Input:latency of fixed sized data,available datacenters
Output:Besthop
Assumption: no packet loss,static user,reliable connection

```
Best_Hop()
{
        if (Offload_Algo ( ))
        {
        for each datacenter H
                if (latency_H <min && distance_H<mind)
                        {
                        min = latency_H
                        mind= distance_H
                        }
        done
        }
Return chosen datacenter H with min latency.
}
```

Proposed algorithm is implemented in EC2 instances and performance has been increased

VI. MOBILE CLOUD COMPUTING PLATFORMS USED

Overheads for offloading are related mainly to data size of offload as well as network latency [11].For this research work, google cloud platform has been used for deploying varied sized applications and after that runs over sandbox which is provided by GAE.The simulation is done for cpu utlization and latency of varied sized applications. Amazon EC2 web service helps developers to build failure proof apps.

A. AMAZON EC2 VS GOOGLE APP ENGINE

EC2 is Iaas offering while GAE is Paas offering.

	GAE	Amazon EC2
Pros	• Free service •Billing only for CPU usage	• Vendor-lock s/m not available to make job to move code to other box easy

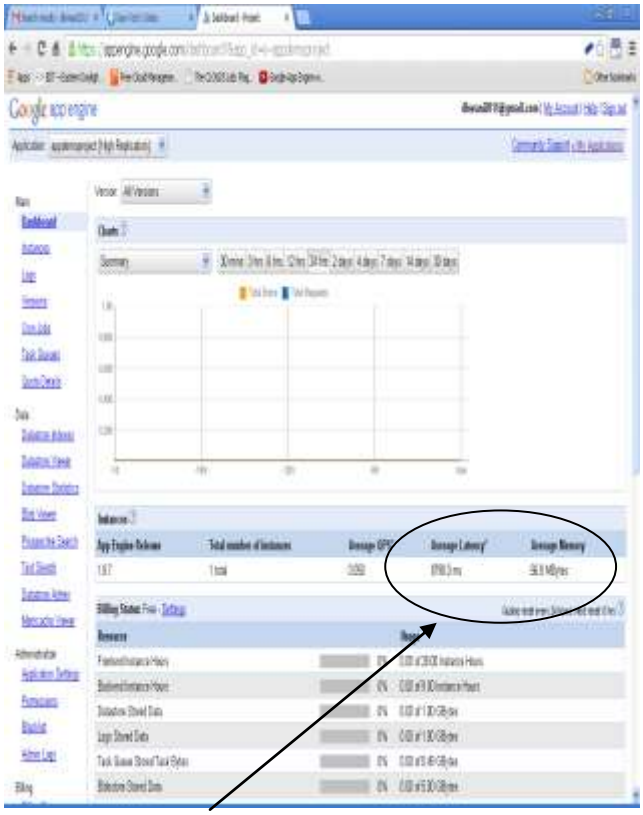
		• Written in the simple programming languages like dot Net, C#, MVC • Backend support of Amazon team
Cons	• Lack of support for .NET/C#, only PHP, Java, Python are supported • Lot of restrictions of data access, more tedious task for managing.	•Priced for different bandwidth •Scaling is tough, •even if one server fails, there is breakdown. •Requires admin to monitor instances

VII. EXPERIMENTAL ANALYSIS AND IMPLEMENTATION DETAILS

LATENCY CALCULATIONS FOR NON MOBILE DEVICES

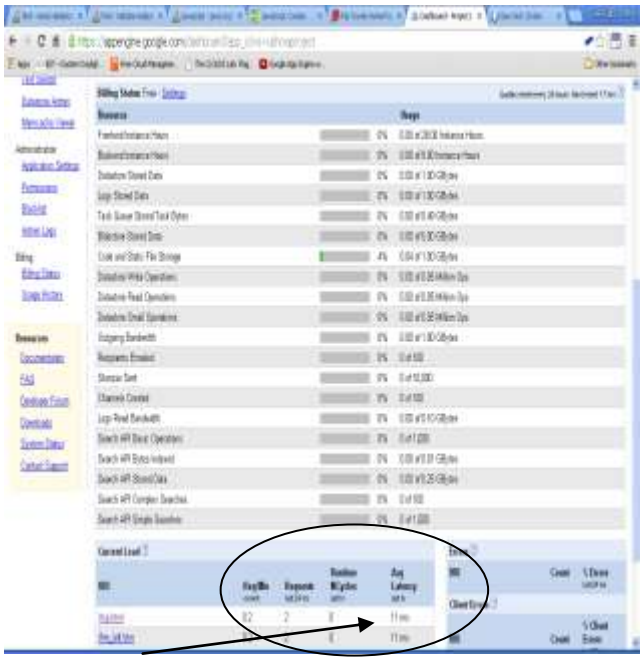
While offloading different tasks on,energy can be saved for large tasks[19].

A. LATENCY CALCULATIONS OF PHP APP (HEAVY CODE STORAGE ON CLOUD) IN PC (NON MOBILE)



Latency: 8798.0ms

B. LATENCY CALCULATIONS OF HTML APP (LIGHT CODE STORAGE ON CLOUD) IN PC



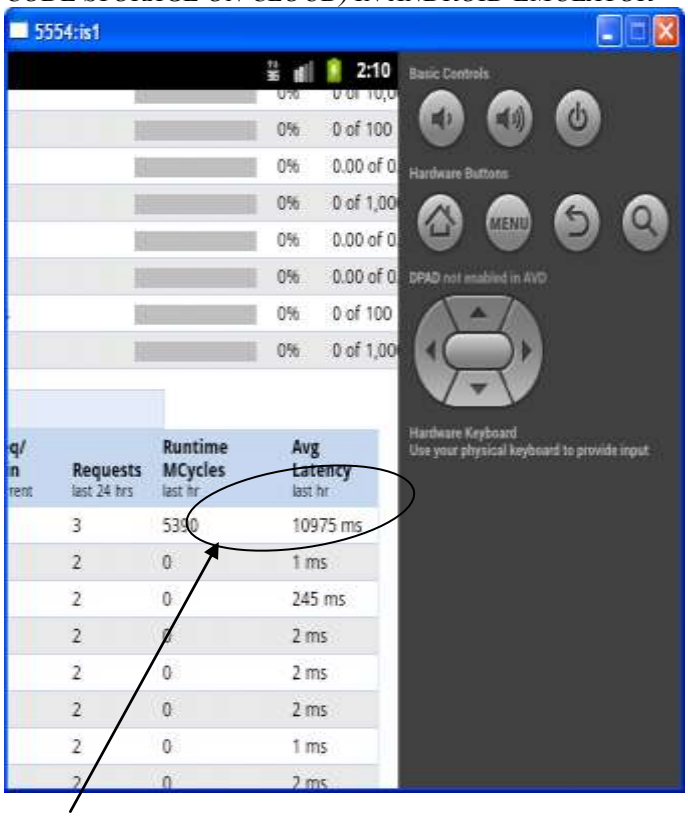
Latency: 11ms only!!

Non mobile devices shows greater performance as compared to the mobile devices ,but considering mobility as factor mobile devices are more favored

LATENCY CALCULATIONS THROUGH VARIED MOBILE DEVICES

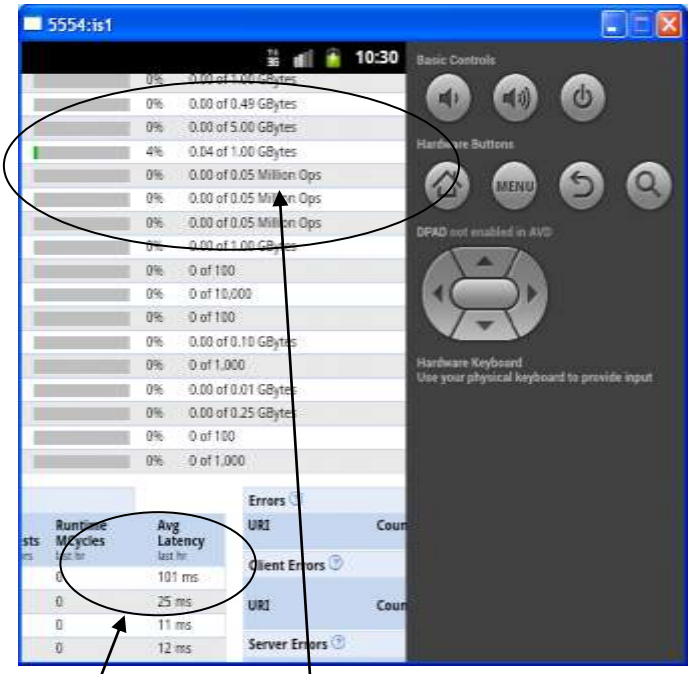
ANDROID EMULATOR V2.3.3

A. LATENCY CALCULATIONS OF PHP APP (HEAVY CODE STORAGE ON CLOUD) IN ANDROID EMULATOR



10975 ms

B. LATENCY CALCULATIONS OF HTML APP (LIGHT WEIGHT CODE STORAGE ON CLOUD) IN ANDROID V2.3.3

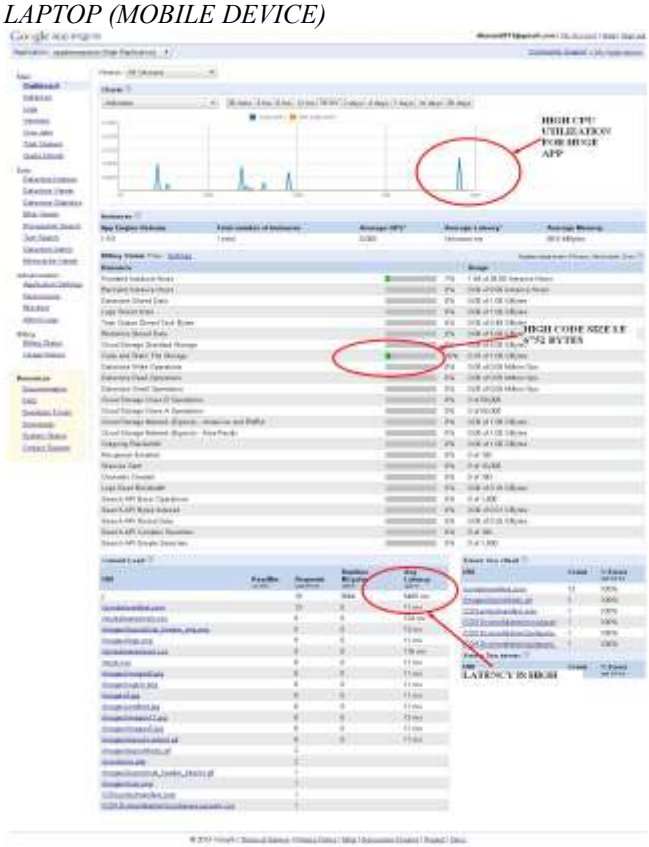


Latency: 101 ms only!!!

Code size on cloud is less (4% of 1GB)

LAPTOP

A. LATENCY CALCULATIONS, CPU UTILIZATION OF HEAVY WEIGHT CODE STORAGE ON CLOUD IN LAPTOP (MOBILE DEVICE)



B. LATENCY CALCULATIONS, CPU UTILIZATION OF LIGHT WEIGHT CODE STORAGE ON CLOUD IN LAPTOP (MOBILE DEVICE)

OFFLOADING THROUGH WI-FI

Wi-Fi	PHP app(heavy coded-6752 bytes)Latency(in ms)	Java APP(medium coded-2631 bytes)	App-2598 bytes	HTML APP(Light coded-388 bytes)) Latency (in ms)
Android emulator	10975ms 580ms 230ms	4000 ms 3000 ms 200 ms	3950ms 3000ms 150 ms	101 ms 49ms 25 ms
From PC	8798ms 4000ms 20ms	78ms 78 ms	60 ms 50 ms	11 ms 5ms 1ms
From Laptop	10660ms 5409ms	2792 ms 2700 ms	2858ms 2500ms	42 ms 32 ms

Tab 3: Latency comparisons of various apps (Wi-Fi)

C. GRAPHICAL RESULTS SHOWING DIFFERENT SIZED APPS WI-FI OFFLOAD

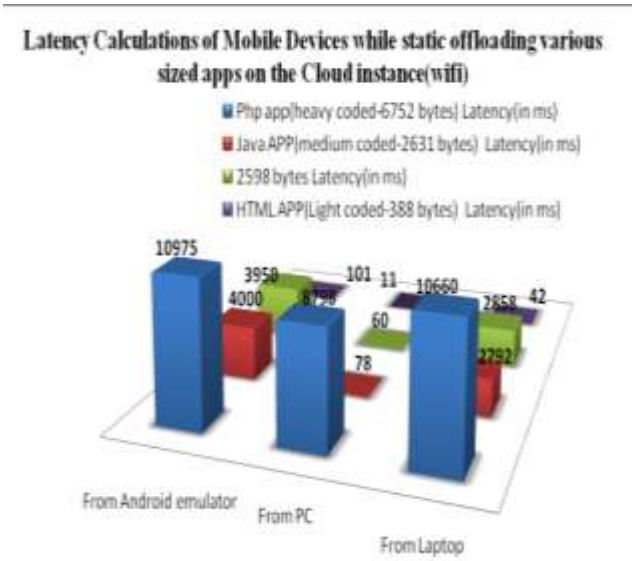
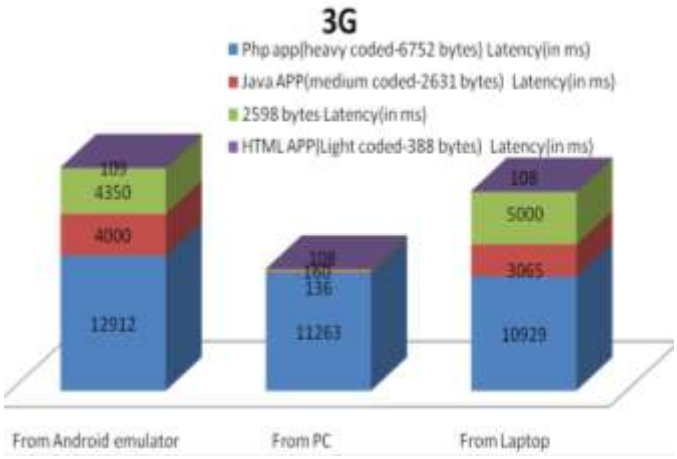


Fig 4:Showing Wi-Fi offload of varied sized apps.

STATIC OFFLOADING THROUGH 3G

D. GRAPHICAL RESULTS SHOWING 3G OFFLOAD



OFFLOADING THROUGH 3G

3G	PHP app(heavy coded-6752 bytes)	Java APP(medium coded-2631 bytes)	2598 bytes	HTML APP(Light coded-388 bytes)
	Latency(in ms)	Latency(in ms)	Latency(in ms)	Latency(in ms)
From Android emulator	12912 11000	4000 3900	4350 4350	109 109
From PC	11263 10000	136 136	160 160	108 108
From Laptop	10929 10000	3065 3065	5000 5000	108 108

Fig 5:Showing 3G offload of varied sized apps.

Simulation results show that latency is high for 3G.Such long latencies makes the code offload expensive.Hence this shows that 3G gives less performnace as compared to Wi-Fi.

VIII. PERFORMANCE IMPROVEMENT BY OPTIMIZING LATENCY THROUGH CLOUDLET CREATION IN EC2 INSTANCE

Here same sized data has been offloaded in ec2 intance and it has been found that cpu utilization is decreased and the performance has increased by creating cloudlet architecture.

IX. CREATING A CLUSTER OF CLOUDLETS THEREBY INCREASING PERFORMANCE CONSIDERING MOBILITY OF USER

Cloudlets are created in the different regions considering user's mobility as factor and nearby cloudlet is selected for offloading data of same size whichever is nearest to the user and has decreased latency.

Network Latency Tests

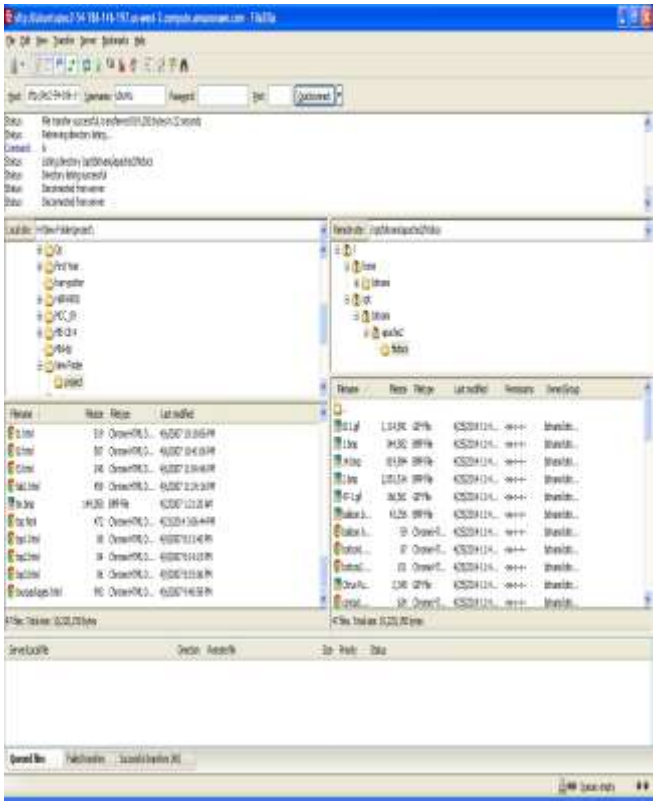
Service	Location	Time (secs)	# of Samples	Min ms	Max ms	Std Dev	Median ms	Avg ms
Amazon EC2	OR, US	1.09	2	367	723	46.19%	723	545
Amazon EC2	CA, US	1.06	2	370	694	43.06%	694	532
Google AppEngine		1.42	1	1418	1418	0%	1418	1418
Amazon EC2	VA, US	1.02	2	343	675	46.12%	675	509

Calculating the latency for all regions, it can be shown that California is nearer to the mobile user.

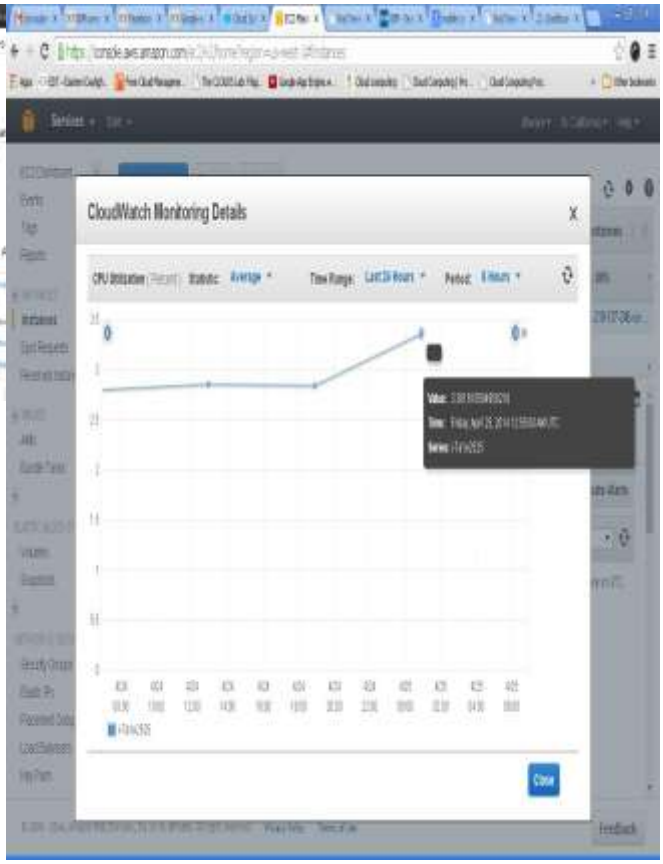
A. APP AT NORTH CALIFORNIA REGION

Launch instance for authenticated user.

Connecting to the instance with secure shell SFTP client for higher security

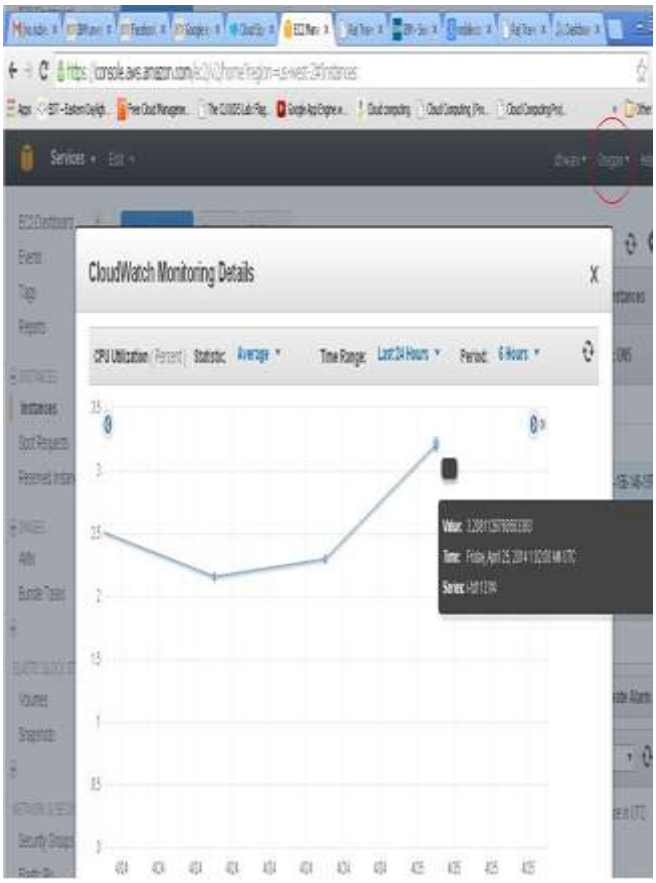


Offloading same sized app at different regions depending on distance.Calculate the CPU utilization for all regions.The region in which the utilization is comparatively less is chosen as the best hop to offload the data depending on users mobility.



CPU Utilization is 3.36 in instance at North California region

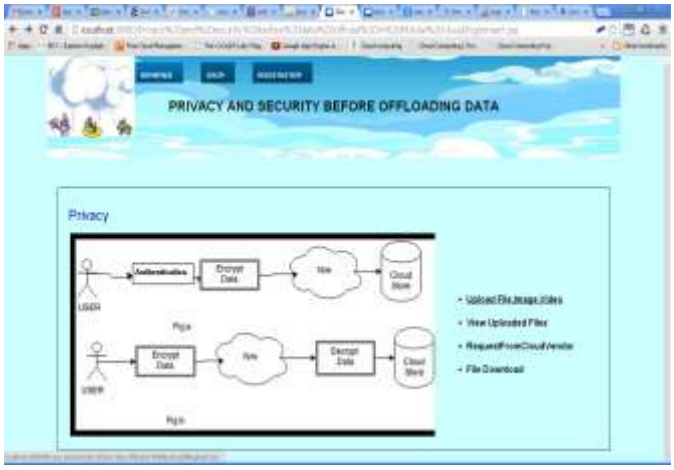
B. APP AT OREGON REGION (MUCH NEARER REGION)



CPU Utilization is 3.27 in instance at Oregon region.
Hence it can be shown that the performance can be increased nearly by 0.09%, hence offloading the data to much nearer regions thereby decreasing latency.

X. VIRTUAL SYSTEM SHOWING PRIVACY BEFORE DATA OFFLOADING

A virtual system in is created showing privacy of data from the cloud vendor itself. Cloud vendor like Amazon can have the data only after requesting the decrypt key from the user. Here a 3 users system is designed→Mobile User, cloud service provider and Cloud Vendor, where user will upload file using unique secret key. If the Cloud Vendor wants to have the data, he can do so only by having the key shared by mobile user.



CONCLUSION AND FUTURE WORK

Cloud computing is known to be a promising solution for mobile users due to mobility,resource constrained,etc.[21] Hence Cloud Computing can be used for running mobile users applications.But not all applications are beneficial when migrated to the cloud.For example the heavy applications when migrated to cloud consumes more CPU utilization of device and more latency as compared to the light weighted applications, hence for the smart phones,it is much better to offload heavy apps.Hence smart phones can last longer through the offload. Overheads of offloading namely performance, privacy of data, storage, battery lifetime of mobile devices has been tackled off through cloud computing. Heterogeneity on different platforms and scalability of applications needs to be take care off in the future.

ACKNOWLEDGEMENT

First of all I would like to thank Almighty for all his kind blessings without which my work has been knock-off. I would like to appreciate the faculty members of LJIET for providing me great support. Last but not the least; I would honor to thanks my parents, sister and friends for their encouragement and confidence on my ability.

REFERENCES

[1] http://en.wikipedia.org/wiki/Cyber_foraging
[2] Dhwani Sanghavi,Prof Jignesh Vania,"*Generating Future Systems through Mobile Cloud Computing and Approaches to Cyber Foraging*",IJETTCS,Vol 2,Issue 5,p-124-129,Sep-Oct, 2014
[3] Kafaie, S.; Kashefi, O.; Sharifi, M., "Augmented Mobile Devices through Cyber Foraging," *Parallel and Distributed Computing (ISPDC), 2011 10th International Symposium on*, vol., no., pp.1451.52, 6-8 July 2011, IEEE
[4] Malawski, M.; Kuźniar, M.; Wójcik, P.; Bubak, M., "How to Use Google App Engine for Free Computing," *Internet Computing, IEEE* , vol.17, no.1, pp.50,59, Jan.-Feb. 2013
doi: 10.1109/MIC.2011.143
[5] http://aws.amazon.com/ec2/
[6] http://en.wikipedia.org/wiki/Mobile_cloud_computing
[7] Kristensen, M.D.; Kjaergaard, M.B.; Toftkjaer, T.; Bhattacharya, S.; Nurmi, P., "Improving pervasive positioning through three-tier cyber foraging," *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on* , vol., no., pp.135,140, 21-25 March 2011
[8] Sharifi, Mohsen; Kafaie, Somayeh; Kashefi, Omid, "*A Survey and Taxonomy of Cyber Foraging of Mobile Devices*," Communications Surveys *& Tutorials, IEEE* , vol.14, no.4, pp.1232,1243, Fourth Quarter 2012
[9] K. Kumar and L. Yung-Hsiang, "*Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?*" IEEE Computer , vol.43, no.4, pp.51-56, April 2010. doi: 10.1109/MC.2010.98
[10] Hong Zhao, "Data Security and Privacy Protection Issues in Cloud Computing," *Computer Science and Electronics Engineering (ICCSEE),*

2012 International Conference on vol.1, no., pp.647,651, 23-25 March 2012,IEEE
[11] Li, Kai Bu, Xuan Liu, Bin Xiao, "*ENDA: embracing network inconsistency for dynamic application offloading in mobile cloud computing"*, Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing (2013), pp. 39-44, doi:10.1145/2491266.2491274 Jiwei
[12] Qi, Han and Gani, Abdullah," *Research on mobile cloud computing: Review, trend and perspectives*", in DICTAP, by IEEE, 2012
[13] Mahadev Satyanarayanan; Bahl, P.; Caceres, R.; Davies, N., "*The Case for VM-Based Cloudlets in Mobile Computing,*" Pervasive Computing*, IEEE* , vol.8, no.4, pp.14,23, Oct.-Dec. 2009doi: 10.1109/MPRV.2009.82
[14] Cuervo, Eduardo and Balasubramanian, Aruna and Cho, Dae-ki and Wolman, Alec and Saroiu, Stefan and Chandra, Ranveer and Bahl, Paramvir," *MAUI: making smartphones last longer with code offload*", 2010
[15] Satyanarayanan, M. Lewis, G. ; Morris, E. ; Simanta, S. ; Boleng, J. ; Kiryong Ha,"*The Role of Cloudlets in Hostile Environments*",2013, IEEE CS
[16] Shiraz, M.; Gani, A.; Khokhar, R.H.; Buyya, R., "*A Review on Distributed Application Processing Frameworks in Smart Mobile Devices for Mobile Cloud Computing*", *Communications Surveys & Tutorials, IEEE* , vol.15, no.3, pp.1294,1313, Third Quarter 2013
[17] Xiao Ma; Yong Cui; Lian Wang; Stojmenovic, I., "Energy optimizations for mobile terminals via computation offloading," Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on , vol., no., pp.236,241, 6-8 Dec. 2012S
[18] Namboodiri, V.; Ghose, T., "To cloud or not to cloud: A mobile device perspective on energy consumption of applications," World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a , vol., no., pp.1,9, 25-28 June 2012
[19] Kristensen, M.D.; Bouvin, N.O.,"*Using Wi-Fi to save energy via P2P remote execution*," Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on, vol., no., pp.123, 128, March 29 2010-April 2 2010
[20] www.slideshare.net/shreyashsrs66/google-app-engine-27050013
[21] Qi, Han and Gani, Abdullah," *Research on mobile cloud computing: Review, trend and perspectives*", in DICTAP, by IEEE, 2012

Dhwani Sanghavi has received Bachelor of Engg degree in 2012 from L.J. Inst. of Engg. And Tech affiliated with Gujarat Technological University. She is pursuing her Masters from L. J. Inst. of Engg and Tech affiliated with Gujarat Technological University.

Prof. Jignesh Vania is currently the faculty of Computer Engg. Department in L. J. Inst. of Engg.And Tech affiliated with Gujarat Technological University.

