Towards Efficient Computation Offloading for Resource-Deficient Mobile Devices

Collinson Colin M. Agbesi[1], Katsriku Apietu Ferdinand[2], Jamal Abdulai-Deen[3] and Kofi Sarpong Adu-Manu[4]

Abstract*:* Mobile and smart computing devices have become ubiquitous and the preferred computing device for their daily tasks and business activities. Despite the phenomenal increase in the computational power of these devices, they still are unable to cope with the exceptionally high demand for resources by current applications [1], [10], [11]. It has been suggested that, resource demanding applications and intensive computations from mobile devices should be offloaded to other systems with higher resource capacities in the cloud [3], [4], [5]. Mobile cloud computing is a form of cloud computing that seeks to enhance the capacity and capabilities of mobile and smart computing devices by enabling mobile devices to offload some computational tasks to the cloud for processing[4] [6], [8], [25]. The study also presented an energy model for computation offloading.

Keywords: *Cloud Computing, Mobile Cloud Computing, Computation Offloading*

1. INTRODUCTION

Recent years has witnessed a phenomenal growth and proliferation in the use of mobile and smart computing devices and applications. Devices such as smartphones, tablets, and eBook readers enable users to connect to the internet instantly, giving them access to vast amounts of information and cloud resources. This opens up the possibility of these mobile devices using productivity tools to perform tasks remotely [3], [15], [23]. Despite the growing trend in the use of mobile and smart computing technologies, they still lack the required computational capacity as a well as resources for effective adoption in computational intensive applications [5], [16], [24]. Mobile cloud computing is a variant of cloud computing which addresses the challenges faced by mobile and smart computing devices in running resource intensive applications [5], [13], [26]. Mobile cloud computing seeks to enhance the capacity and capabilities of mobile and smart computing devices enabling them to execute the many resource demanding applications available for mobile users remotely [2], [20], [21]. Under the mobile cloud computing paradigm, computationally intensive and resource demanding mobile applications such as 3D graphics, face recognition, voice recognition, games, videos, malware protection and data manipulation are transferred to other systems with higher resource capacities for efficient processing in the cloud. The goal of mobile cloud computing is to enable mobile and smart computing devices to run and execute the plethora of resource hungry mobile applications which greatly consumes processing power, energy and battery life of mobile devices by offloading computations to other systems with higher resource capacities in the cloud [17], [23], [25].

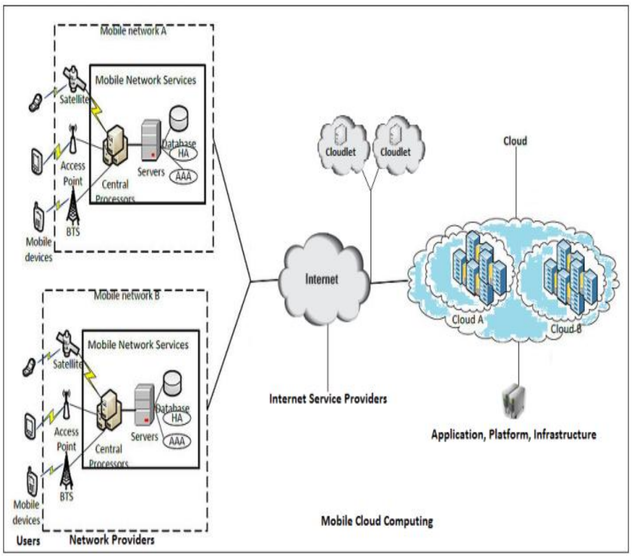

Figure 1: Mobile Cloud Architecture [23]

1.1 Problem Statement

The computational power and capacity of mobile and smart computing devices have increased over the years. This has allowed mobile and smart computing devices to run resource demanding applications [1], [3], [16], [20]. In spite of the increase in computational capacity such as processor, memory, storage, display and battery life; mobile and smart computing devices are still very much limited regarding the kind of applications that they can run and execute [7], [9], [17], [18]. Also the open nature of mobile applications market has encouraged the development of applications by millions of enthusiasts over the world. Unfortunately, many of these applications are resource intensive and put a lot of strain on the mobile device [12], [16], [21]. In many instances the mobile and smart computing devices are unable to run these applications. This is making mobile and smart computing devices unattractive to users in terms of applications that these systems can run and execute [1], [2], [15], [20]. The limitations in computational power, battery life, memory, storage capacities and other resources has led to an emergence of several techniques to enable mobile devices process and execute the many rich and resource demanding applications available for mobile and smart computing devices [7], [14], [19], [25].

2. RELATED WORK

Mobile computational offloading is an aspect of mobile cloud computing that focuses on transferring computations from native applications running on resource limited mobile devices to other powerful systems such as cloud servers for processing and execution [7], [10], [13], [18]. Computation offloading is sending very intensive and resource demanding and complex computations to resourceful servers with higher computational capacities for efficient and effective execution and processing which would otherwise be impossible for mobile and smart computing devices to execute locally [16], [18], [20], [24].

Previous works have identified different mechanisms to solve the seamless execution of offloaded computational tasks from resource limited mobile devices to other systems with higher computational power and resources in the cloud. Some of these researches have proposed sending computationally intensive tasks to nearby surrogates with higher computational resources [9], [10], [25]. Some other research works have proposed the complete cloning or mirroring of a mobile device together with its operating system and applications and sending the image to a cloud server located at a remote place for processing and the results sent back to the mobile devices [10], [22], [26]. Also other research works have proposed a game based approach to computation offloading in which the resource capacities of servers are determine in the cloud and intensive computations are sent to available and resourceful servers to execute.

Chun & Maniatis [6] have proposed the encapsulation of a mobile device software stack including operating environment and applications into a virtual machine image and offloading it to a more powerful system in the cloud for execution. Orsini et. al [18] have proposed a context-aware mechanism for computation offloading in which mobile devices decide when and how to offload based on their state and resource capacities. Also, Cong et. al [8] presented a computation offloading as a services (COSMOS) to provide a leverage for all mobile device requests and reduce monetary cost to the cloud provider. Some studies have provided alternative means of computation offloading but lack the fundamental models to evaluate energy consumed during the computational offloading process.

Hence there is still much work to be done in order to find a good resource efficient computational offloading scheme for mobile and smart computing devices which have variable resource capacities and computational power [2], [15], [16], [20], [22], [25].

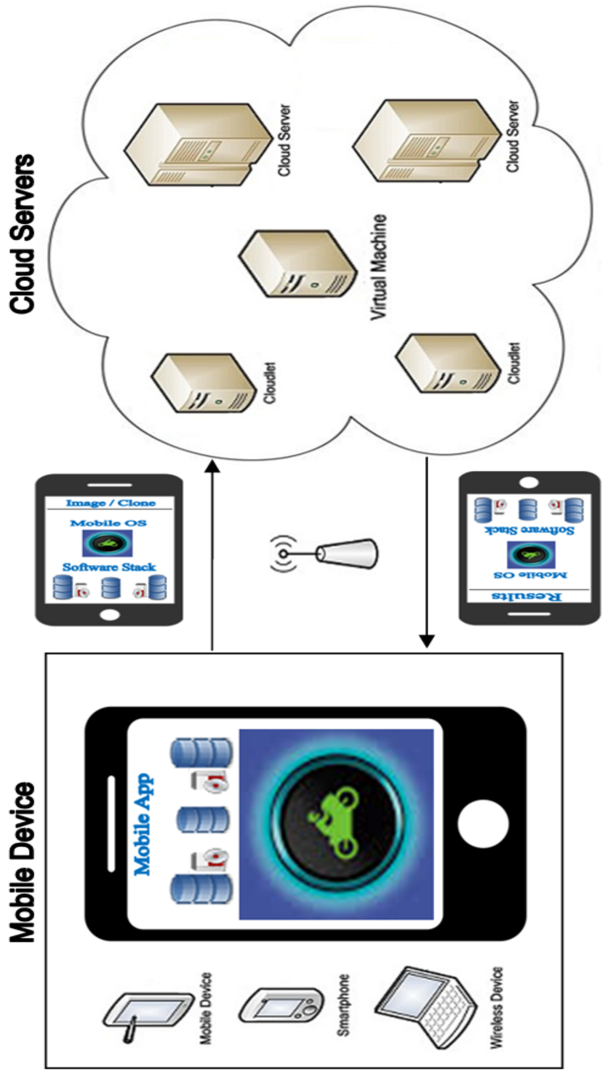

Figure 2: Traditional Computation Offloading Scheme

The traditional computation offloading scheme has presented a means of offloading computations in which the computational tasks to be executed, operative system and software stack of the mobile device is mirrored or cloned and the image send to a server in the cloud for processing.

3. PROPOSED COMPUTATION OFFLOADING FRAMEWORK

In a resource efficient computation offloading scheme, the mobile device, assesses every computational task to be executed to determine whether it can execute the task using its local resources or it cannot. The mobile device compares the volume of task and computations expected to be perform on the tasks to its resource capacities and capabilities. The resource manager determines the available resource capacity of the mobile device. Base on the available resource capacity of mobile devices, a decision is made whether or not to execute the computational task locally or to partition it and send the intensive part to a remote server in the cloud for processing. Then the partitioner divides the computational task or mobile application to be executed into user interaction and intensive tasks.

The user interaction task is then assigned to the local execution manager to execute locally whiles the computationally intensive tasks are sent to the offload manager. The offload manager then transfers the intensive tasks to other powerful server systems with higher and scalable resource capacities in the cloud. The result of the computation in the cloud is sent back to the mobile device. Also, studies have shown that, traditional computation offloading schemes are usually resource efficient when the offloading decision is done accurately and proportionately in terms of the resource availability of mobile devices. Hence the resource manager and partitioner must make the best resource efficient decisions for computational tasks to be executed based on the mobile device.

3.1 Algorithm of Computation Offloading

1. Start:
2. Initialize Mobile Device
3. Run Mobile Application
4. If Mobile Application is Not Intensive
5. Run Application Locally
6. Else if Mobile Application is Intensive

7. Determine Resource Capacity of Mobile Device
8. If Mobile Device is Capable
9. Run Application Locally
10. Else if Mobile Device is Not Capable
11. If Wireless Access is Not Available
12. Force Local Execution or End Application
13. Else if Wireless Access is Available
14. Send intensive mobile application to Servers in the Cloud
15. Return Results to Mobile Device
16. End:

As stated in the algorithm, without the mobile device being switched on, the computation offloading process cannot be started by the mobile device. Also without any computational tasks, the mobile device cannot initialize any computation offloading process. Therefore, the mobile device should be switched on and there should be a computational task in order to follow the computational offloading algorithm.

The initialization involves loading of the operative system and other essential component systems on the mobile device. After the initialization, the mobile application to be executed is run by the mobile device. If the mobile application is not intensive, then it is executed locally else if the mobile application is intensive, depending on the resource availability of the mobile device and also if wireless connectivity is available on the mobile device, the mobile application is divided into less intensive and intensive parts. Then the less intensive user interaction part is executed locally by the mobile device whiles the computationally intensive part is sent to a cloud server system for remote execution and the results is sent back to the mobile device.

The computation offloading process is performed whenever there is a computational activity or task to be performed by the mobile device which may or may not be intensive for the mobile device to execute.

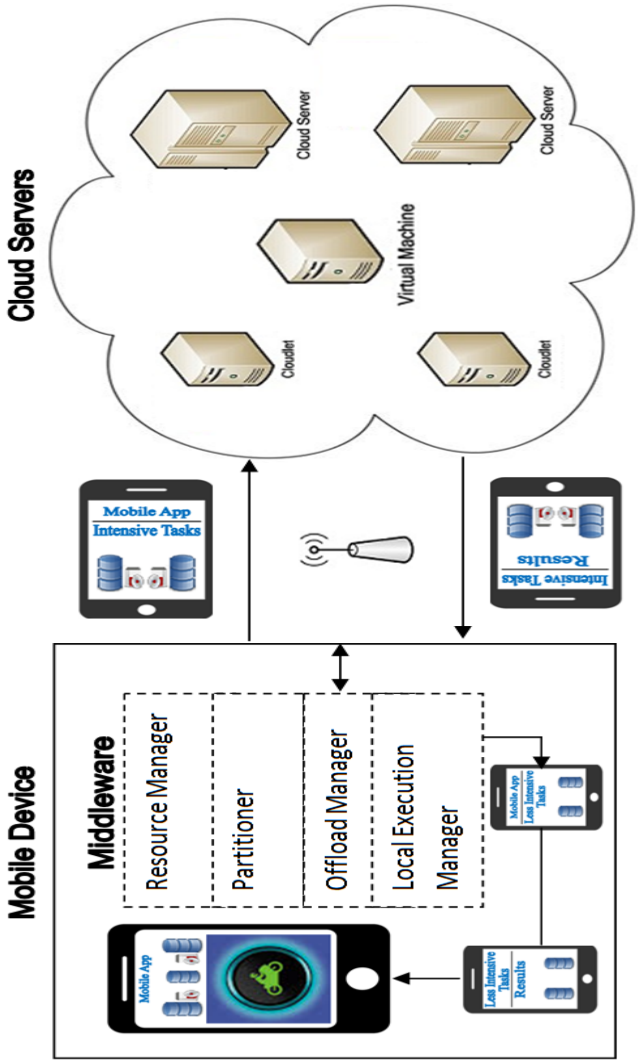

Figure 3: Proposed Computation Offloading Framework

3.2 Flowchart of Computation Offloading

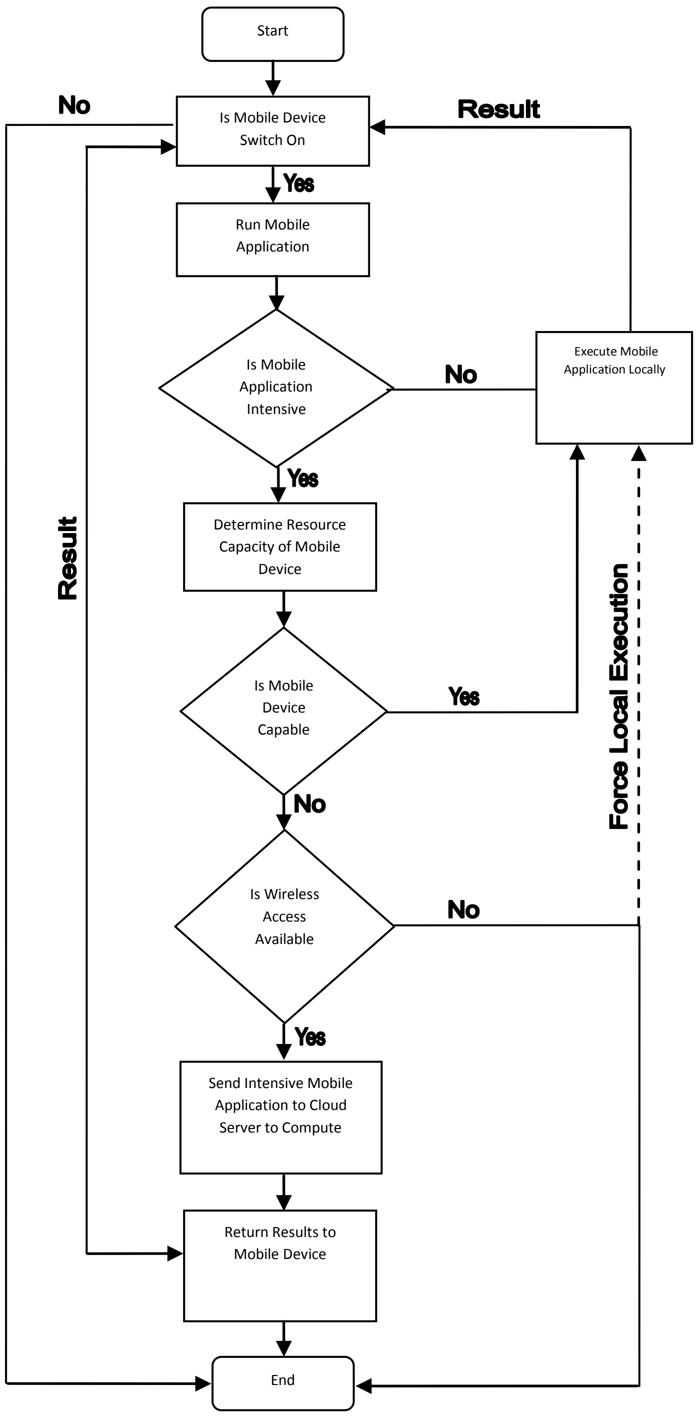

Figure 4: Flowchart of Proposed Computation Offloading Scheme

From the flowchart diagram, the mobile device is switched on and its operative system initialized together with other component systems. If the mobile device is not powered on, then the computation offloading process is halted. If the mobile device is powered on, then the mobile application to be executed is run by the mobile device. If the mobile device detects that the mobile application is not intensive, the mobile device will then execute the mobile application locally using its own resources.

But if it the mobile device detects that the mobile application is intensive or it cannot execute the computational task, it will determine its resource availability. If the mobile device can strain execute the computational task or mobile application, depending on the situation, it might attempt to execute it if it is still efficient in the worst case.

Otherwise if there is a wireless connectivity capability on the mobile device, then the computational task or mobile application is divided into intensive and less intensive tasks.

The less intensive tasks are then sent to the mobile device for processing locally whiles the computationally intensive parts are sent to the offload manager who then forwards the computationally intensive tasks to a server in the cloud for remote execution and after the remote processing the results is sent back to the mobile device.

3.3 Mathematical Model of Computation Offloading

The study presented a mathematical model for computation offloading. Also a model for calculating the energy and time cost of computation offloading is presented.

3.3.1 Computation Offloading Decisions

Let MA denote mobile application
Let MD denote the mobile device

Let S denote cloud servers
Let the intensive part of the mobile application be denoted MA_i
Also

Let MA_u denote the user interaction part of mobile application (MA)

Then we can assign the less user interaction part of mobile application to mobile device for processing locally as follows;

MA_u assign to MD

Then we can assign the intensive part of mobile application to mobile device for processing remotely as follows;

MA_i assign to S

3.3.2 Energy Cost of Computation

Let EC denote energy consumption
Let ECC denote energy cost of computation
Let MAC denote intensive parts of mobile application in the cloud
Let MAL denote intensive parts of mobile application locally executed on mobile device
Let EC_{MAC} denote energy consumption of intensive mobile applications in the cloud
Let EC_{MAL} denote energy consumption of less intensive mobile application locally executed
Let ETT_{MAi} be the energy used to transmit intensive mobile application to the cloud
Let ETE_{MAi} be the energy used to execute the intensive mobile application in the cloud
Let ETR_{MAi} be the energy used to receive the results of executing intensive part of mobile application in the cloud
Then

$$EC_{MAi} = ETT_{MAi} + ETE_{MAi} + ETR_{MAi} \qquad (1)$$

$$EC_{MAi} = EC_{MAC}$$

Likewise

$$EC_{MAu} = ETE_{MAu} \qquad (2)$$

$$EC_{MAu} = EC_{MAL}$$

Then the total energy cost of computation is given by Energy Cost of Computation (ECCMA) where

$$ECC_{MA} = EC_{MAC} + EC_{MAL} \qquad (3)$$

$$ECC_{MA} = EC_{MAi} + EC_{MAu}$$

Therefore

$$ECC_{MA} = ETT_{MAi} + ETE_{MAi} + ETR_{MAi} + ETE_{MAu} \qquad (4)$$

3.3.3 Time Cost of Computation

Let TC denote time consumption
Let TCC denote time cost of computation
Let MAC denote intensive parts of mobile application in the cloud
Let MAL denote intensive parts of mobile application locally executed on mobile device
Let TC_{MAC} denote time consumption of intensive mobile applications in the cloud
Let TC_{MAL} denote time consumption of less intensive mobile application locally executed
Let TTT_{MAi} be the time used to transmit intensive mobile application to the cloud
Let TTE_{MAi} be the time used to execute the intensive mobile application in the cloud

Let TTR_{MAi} be the time used to receive the results of executing intensive part of mobile application in the cloud

Then

$$TC_{MAi} = TTT_{MAi} + TTE_{MAi} + TTR_{MAi} \quad (1)$$

$$TC_{MAi} = TC_{MAC}$$

Likewise

$$TC_{MAu} = TTE_{MAu} \quad (2)$$

$$TC_{MAu} = TC_{MAL}$$

Then the total energy cost of computation is given by Energy Cost of Computation (ECCMA) where

$$TCC_{MA} = TC_{MAC} + TC_{MAL} \quad (3)$$

$$TCC_{MA} = TC_{MAi} + TC_{MAu}$$

Therefore

$$TCC_{MA} = TTT_{MAi} + TTE_{MAi} + TTR_{MAi} + TTE_{MAu} \quad (4)$$

4. METHODOLOGY

4.1 Experimental Setup

The study setup an energy test bed using a De Lorenzo DL3155AL power supply and De Lorenzo DL10060 power meter and two digital meters connected in parallel to the power supply and in series to the mobile device respectively. The test bed was used to measure the energy consumed during the processing of tasks by the mobile device client.

The study also developed a mobile application for performing intensive computations and processing of large volumes of data files in comma separated values (CSV) format. The study used a Samsung Galaxy Mega phone and a laptop which acts as a server to process and migrate very large volumes of data files in comma separated values (CSV) format into a database which was developed for the mobile phone and laptop systems to help collate, process and transfer huge volumes of data in a flat file format for further computations.

Different sizes of data files from thousands of records approximately 128KB up to about ten million (10,000,000) records approximately 500MB in size were passed to the application to process and the results were logged for both mobile phone and laptop systems. The very low computation involved data of size between 1000 to 5000, the low computation involved data of size between 10000 to 50000 , the high computation involved data of size between 600000 to 1000000 and the very high computation involved data of size between 2000000 to 10000000. The study also tried to determine the time taken to execute various computational tasks during the experiment.

The results of the demonstration were exported into Microsoft Excel for analysis and plotting some useful graphs to help the study discussions. It was observed during the experiment that by offloading intensive applications from mobile and smart computing devices to other systems with higher resource capacities, a great amount of resource efficiency is achieved.

Figure 5: Diagram of Energy Emulation Test Bed

5. RESULTS PRESENTATION & DISCUSSION

5.1 Result Presentation: Below are the results of the study experiment

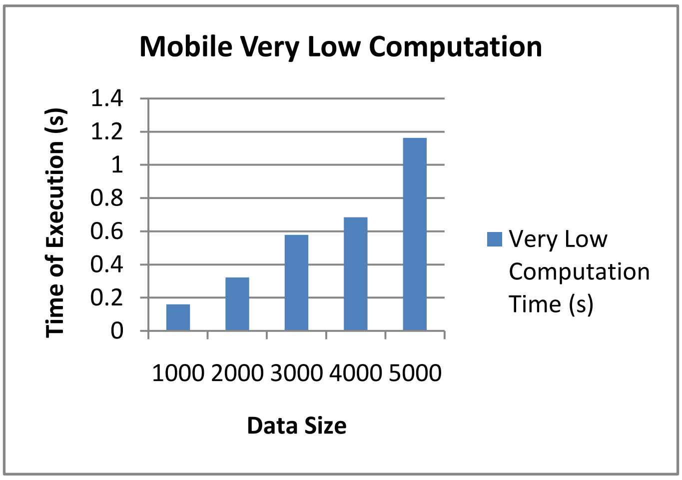

Figure 6a: Mobile Very Low Computation

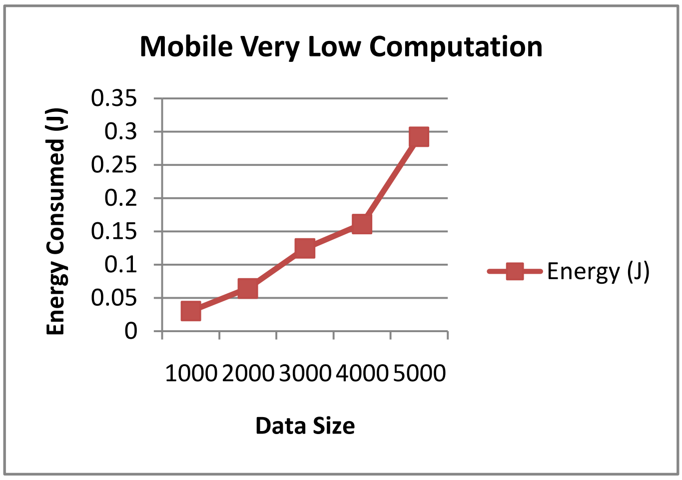

Figure 6b: Mobile Very Low Energy Consumed

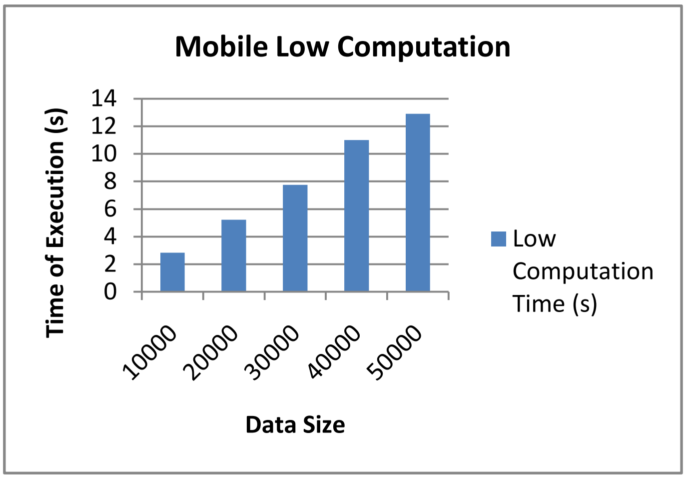

Figure 7a: Mobile Low Computation

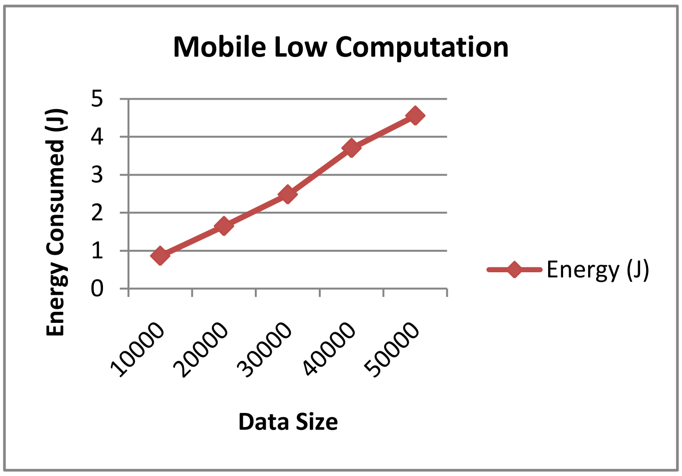

Figure 7b: Mobile Low Energy Consumed

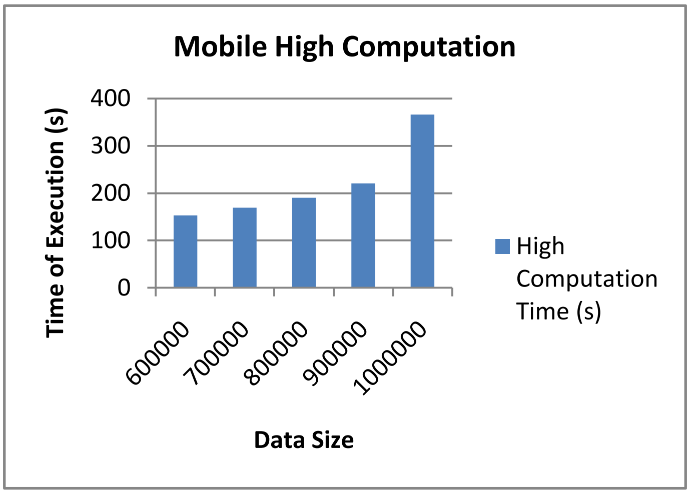

Figure 8a: Mobile High Computation

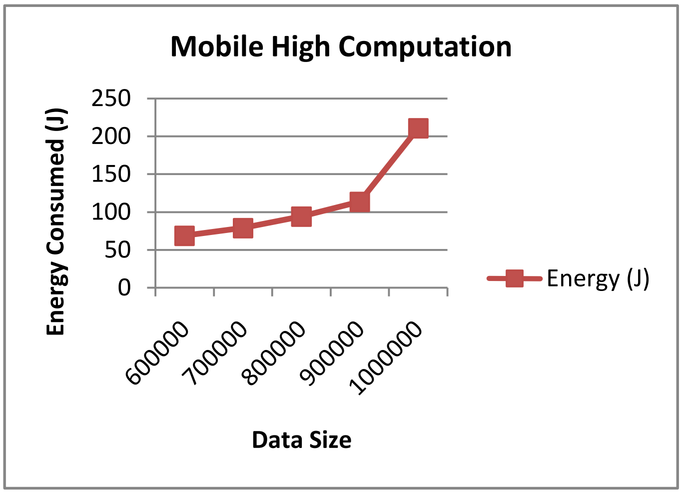

Figure 8b: Mobile High Energy Consumed

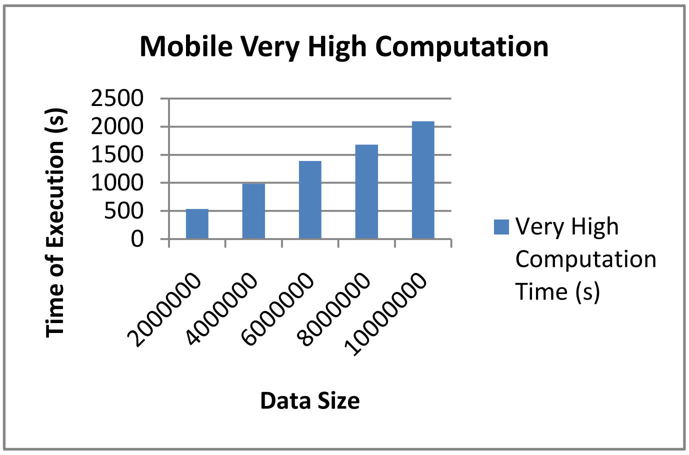

Figure 9a: Mobile Very High Computation

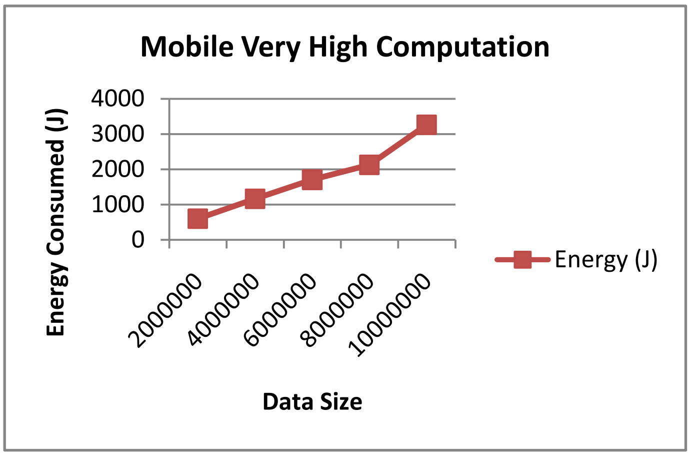

Figure 9b: Mobile Very High Energy Consumed

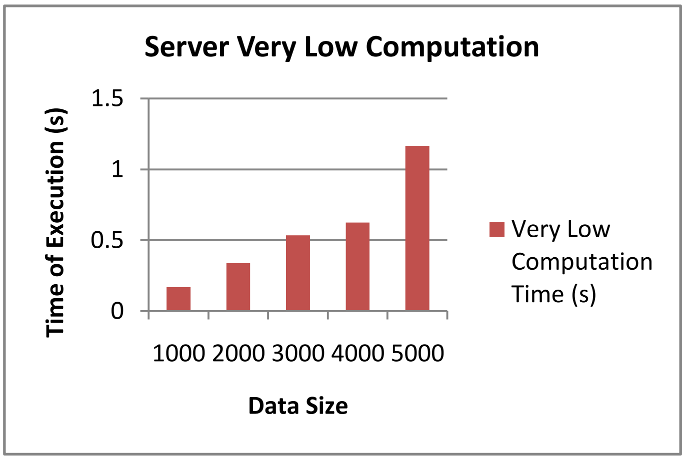

Figure 10: Server Very Low Computation

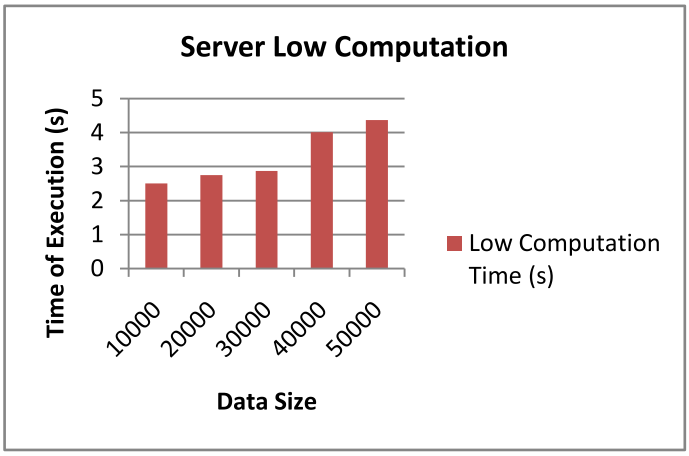

Figure 11: Server Low Computation

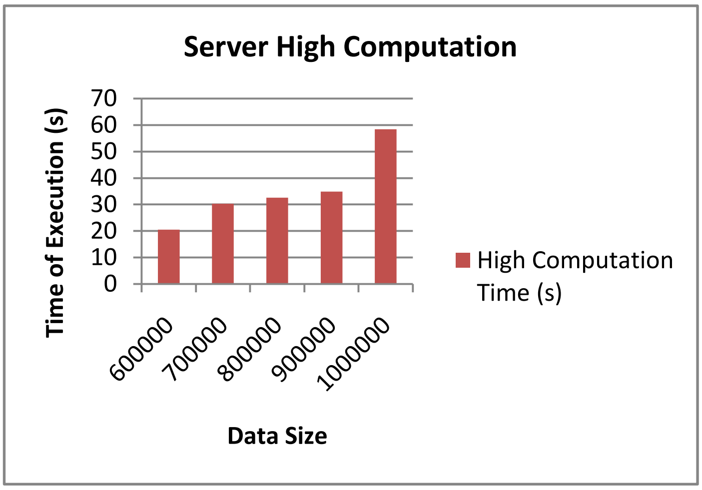

Figure 12: Server High Computation

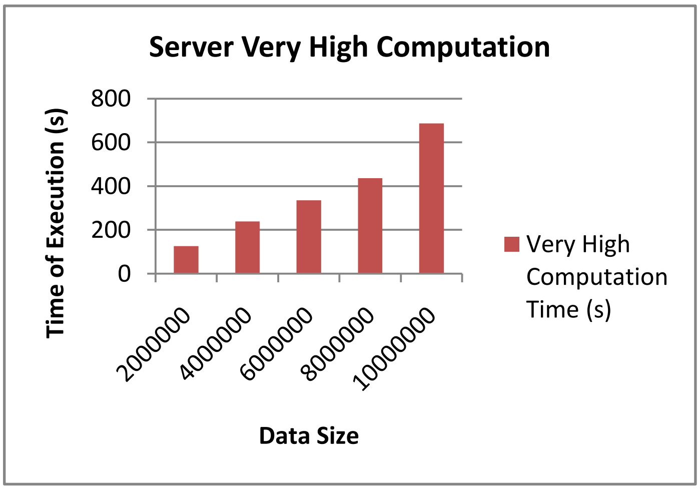

Figure 13: Server Very High Computation

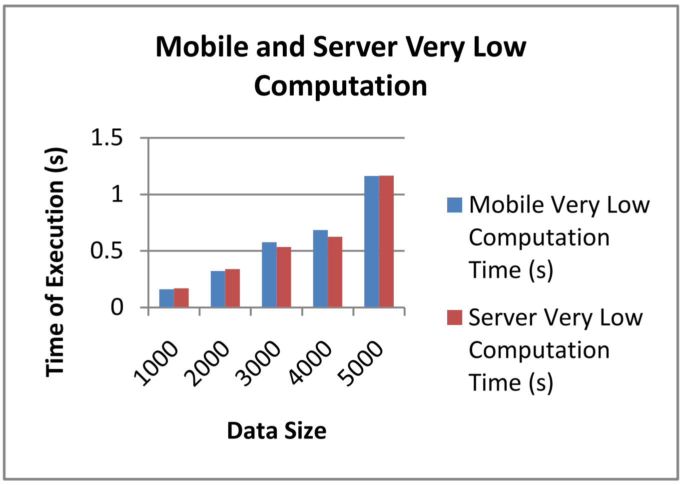

Figure 14: Mobile & Server Very Low Computation

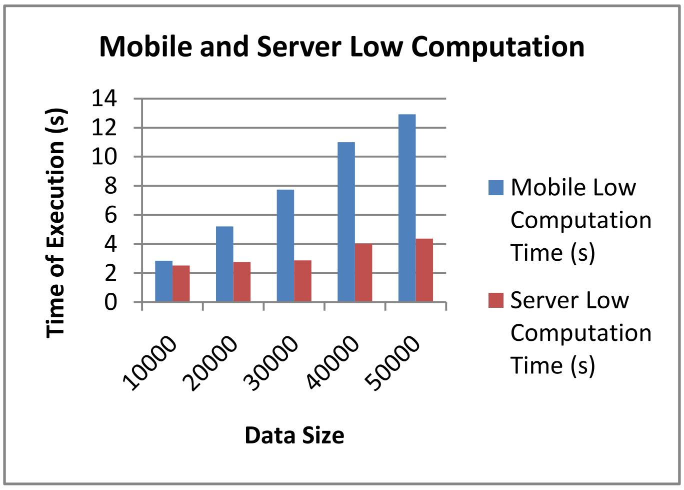

Figure 15: Mobile & Server Low Computation

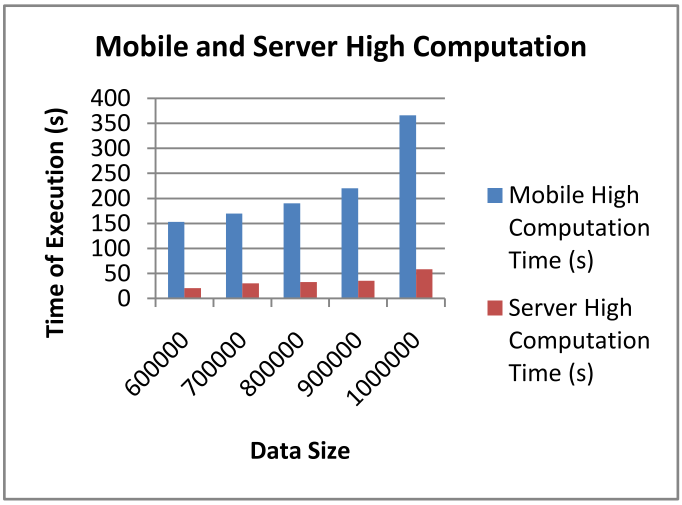

Figure 16: Mobile & Server High Computation

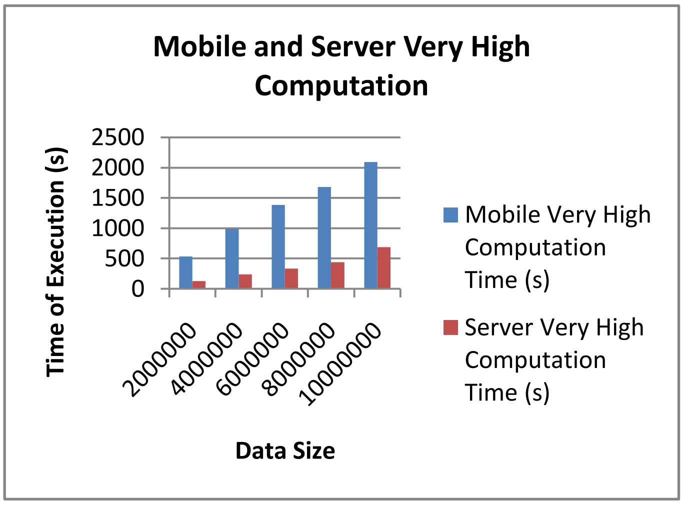

Figure 17: Mobile & Server Very High Computation

5.2 Result Discussion

From the above figures 6a, 7a, 8a, and 9a; the results shows that the time of execution increases with respect to the data size. As the size of data increases, there is a corresponding increase in the computational time. Figure 6a and 7a shows that as the size of data increases, the computational time also increases proportionally compared to the size of data. Also from figure 8a and 9a the size of the computational task to be performed increases, the time taken to execute the tasks also increases astronomically at a higher rate compared to the computational tasks hence the need to offload these tasks for efficient execution by other systems with higher resource capacity.

Likewise figures 6b, 7b, 8b and 9b; shows the energy consumption of the mobile computation during the experiment. It can be observed that, the amount of energy needed to execute a task increases with respect to the size of data and computational task to be executed. Figure 6b and 7b shows that, the amount of energy needed to execute a task increases proportionally with respect to the computational task to be executed. Also, figure 8b and 9b above shows that, a large amount of energy is needed by the mobile device to execute intensive and resource demanding tasks. As the size of data and computation increases, a lot of energy is needed to execute these resources demanding and intensive tasks hence the need to offload intensive computations for efficient processing.

Also, figure 10, 11, 12 and 13 shows that; the time of execution increases proportionally with respect to the data size. As the task to be performed increases, the amount of time needed to perform the computation also increases with respect to the computational task. The server system presented has more computational resources thus memory, storage and processor power and is able to execute tasks faster and efficiently than the mobile device client. Therefore, given any intensive computational tasks, the server system is able to execute computational tasks within the shortest possible time using a small amount of energy as compared to the mobile device client.

Lastly, from figures 14, 15, 16 and 17; it can be observed that for low computational tasks, there is a little difference between the execution time for both mobile and server systems. In some cases, it is beneficial to run and execute low computation tasks on mobile devices than to offload to other systems with higher capacities. Some mobile devices are capable of running and executing low computational tasks faster and efficiently and hence there is no need to offload some low computational tasks. Again, from figure 17, it can be seen that with very high computational tasks, the server performance is better than that of the mobile device. The time taken to execute tasks by the mobile device client is much higher compared to that of the server systems. In the case of moderately high computations, it is sometimes beneficial and efficient to offload some computational tasks to other systems with higher capacities than enable resource constrained mobile devices to execute the tasks. Also from figure 14, it was discovered that, for very huge and complex computational tasks, it is far more efficient to offload to other systems with higher resource capacities than to use resource limited mobile devices. It was also observed during the experiment that, very high computations which involve big data files of sizes above 1 million records are very intensive and stressful for mobile device clients to process and migrate. On the other hand, very low computations which involve the processing of data files below 10000 were efficiently and quickly processed on mobile phone clients than their server counterparts.

6. CONCLUSION

Generally, mobile and smart computing devices are resource limited and handicapped [13], [15], [20], [24]. During the study, it was observed from the experiment that, most mobile devices are unable to run and execute highly intensive computational tasks. Computation offloading is a key benefit of mobile cloud computing which focuses on transferring very intensive computations from resource limited mobile and smart computing devices to other powerful systems such as cloud servers for efficient and timely processing and execution [8], [12], [13], [19], [22], [25].

It was discovered during the research experiment that, offloading or transferring intensive computations from mobile devices to other systems with higher resource capacities in the cloud is far more efficient than processing of these intensive applications locally on mobile and smart computing devices. A great amount of time and energy resource is saved when resource demanding applications are transferred to cloud servers for execution.

Also, the energy model presented in the study provides a means of evaluating the energy consumed during computation offloading in mobile cloud computing. It provides a way of assessing how much energy is used during the computational offloading process by resource limited mobile and smart computing devices.

Computation offloading provides a high degree of time and energy resource savings and optimizations than local processing of tasks on mobile and smart computing devices [10], [16], [21], [24]. A great amount of resources such as time, memory, storage and energy is saved when intensive mobile applications are transferred from mobile phones to other systems with higher resource capacities in the cloud [7], [11], [19], [26].

REFERENCES

[1]Arockiam L., Monikandan S., Parthasarathy G., (2011) "*Cloud Computing: A Survey*" International Journal of Internet Computing (IJIC), ISSN No: 2231 – 6965, Vol1, Issue-2.

[2]Conley et. al (2015), "*Achieving Cost-Efficient, Data-intensive Computing in the Cloud*", Association of Computing Machinery (ACM), 2015.

[3]Cuervo, E., et. al, (2010), "*MAUI: Making Smartphones last longer with code offload*", Proceedings of 8th International Conference on MobiSystems, pp. 49-62

[4]Chen, X., (2015), "*Decentralised Computation Offloading Game for Mobile Cloud Computing*", Proceedings of Parallel Distributed Systems, IEEE Transactions, Vol. 26, No. 4, pp. 974-983

[5]Cong, S., et. al, (2014), "*COSMOS: Computation Offloading as a Service for Mobile Devices*", School of Computer Science, Georgia Institute of Technology, USA

[6]Chun, B. & Maniatis P., (2009), "*Augmented Smartphone Applications Through Clone Cloud Execution*", Proceedings of the 12th Workshop on Hot Topics in Operating Systems, pp. 1-8

[7]Deng, S., et. al, (2015), "*Computation offloading for service workflow in mobile cloud computing*", IEEE Transactions Manuscript

[8]Dudeja, M. & Soni, K., (2014), "*Offloading Schemes in Mobile Cloud*", International Journal of Computer Applications, Vol 96, No. 8.

[9]Fangming, L., et. al, (2013), "*Gearing Resource-Poor Mobile Devices with Powerful Clouds: Architectures, Challenges and Applications*", IEEE, Wireless Communications.

[10] Giurgiu, I., et. al, (2009), "*Calling the Cloud: Enabling mobile phones as interfaces to cloud applications*", Proceedings of 10th ACM/USENIX, International conference, pp. 1-20

[11] Hoang, T. D., et. al, (2013), "*A survey of mobile cloud computing: Architectures, applications and approaches*", Wireless Communications and Mobile Computing, Vol 13, Issue 18, pages 1587-1611

[12] Juntunen, A., et. al, (2012), "*Mobile Computation Offloading - Factors Affecting Technology Evolution*", Proceedings of 11th ICMB.

[13] Karthik, K. et. al, (2012), "*A survey of computation offloading for mobile systems*", Mobile Network Application, DOI 10.1007/s11036-012-0368-0

[14] Kemp, R. et. al, (2010), "*Cuckoo: A computation offloading framework for smartphones*" Proc. 2nd International Conference of Mobile Computing, Application, Services, CA, USA

[15] Kovachev, D. & Klamma, R. (2012), "*Framework for Computation Offloading in Mobile Cloud Computing*", Department of Information Systems and Databases, RWTH Aachen University.

[16] Khan, R. A., et. al, (2014), "*A survey of mobile cloud computing application models*", IEEE Communications Survey.

[17] Niroshinie, F., et. al, (2013), "*Mobile Cloud Computing: A Survey*", Journal of Future Generation Computer Systems

[18] Orsini, G., et. al, (2015), "*Context-Aware computation offloading for mobile cloud computing: Requirement Analysis, Survey and Design Guideline*", 12th International Conference on Mobile Systems and Pervasive Computing, MobiSPC 2015.

[19] Prasad, R. P., et. al, (2012), *"Mobile Cloud Computing: Implications and Challenges"*, Journal of Information Engineering and Applications, Vol. 2, No. 7

[20] Pragaladan, R. & Leelavathi, M., (2014), "*A Study of Mobile Cloud Computing and Challenges*", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 7

[21] Srinivasa, R. V., Nageswara, R. N. K., and Ekusuma K. (2009) "*Cloud Computing: An Overview*" Journal of Theoretical and Applied Information Technology.

[22] Suradkar, J. A. & Bharati, R.D., (2016), "*Computation offloading: Overview, Frameworks and Challenges*", International Journal of Computer Applications, Vol 134, No 6

[23] Satyanarayan, M., et. al, (2009), "*VM-based Cloudlets in mobile computing*", IEEE Pervasive Computing Proceedings

[24] Shravanthi, C. & Guruprasad, H.S., (2014), "*Mobile Cloud Computing as Future for Mobile Applications*", International Journal of Research in Engineering and Technology.

[25] Wolski, S., et. al, (2008), "*Using Bandwidth data to make computation offloading decisions*", Proceedings of IEEE, IPDPS, pp. 1-8.

[26] Xiaopeng, F. and Cao, J., (2011), "*A Survey of mobile cloud computing*", ZTE Communications, Vol. 9, No. 1

Authors Background

Collinson Colin M. Agbesi is a graduate student in Computer Science at the University of Ghana, Legon.

Katsriku A. Ferdinand is the Head of Computer Science Department at the University of Ghana, Legon.

Jamal Abdulai-Deen is a Senior Lecturer at the Department of Computer Science, University of Ghana, Legon.

Kofi Adu-Manu Sarpong is a research student undertaking PhD in Computer Science at the University of Ghana, Legon.

Correspondence: University of Ghana, Legon, Computer Science Department or Koforidua Technical University, Computer Science Department, Koforidua. **Email**: koliny3k@gmail.com