New cloud offloading algorithm for better energy consumption and process time

Rakan Aldmour[1], Sufian Yousef[1], and Mohammad Yaghi[1]

Anglia Ruskin University, Faculty of Science and Technology

Chelmsford, UK

{rakan.aldmour@pgr.anglia.ac.uk,sufian.yousef@anglia.ac.uk,mohammad.yaghi@hotmail.com}

Abstract. Offloading in cloud computing is a way to execute big files in short times due to the available processing resources on core computers. However in some cases it is vital to execute the file locally on the node if the file size is less than a threshold size. There is trade off in this issue due to the limited power of the node, therefore, in this paper a novel algorithm is proposed where the file size in each case is measured and then a decision is taken to either execute the file on the node or to send the file to be processed in the core cloud. The main reason is to save time of the execution of the file. However, the second and important reason, is to save the limited node energy in some large file, the power consumption of the node will be very high. The measurement of the file size and the execution time and the power consumption for the node and the core cloud is measured to represent an input to the execution decision

Keywords: Offloading, Power Consumption, Execution Time, Cloud Computing, Mobile Cloud Computing,

1 Introduction

The first start of the offloading technique has occurred in early of the 1970s and the purpose was for load balancing amongst servers. Bowen, et al., defined offloading as the technique to enhance the quality, efficiency or performance of computation task by sending the heavy tasks to the machines which have a massive powerful computation ability greater than the local machine [1]. There are some of the computation forms such as memory, executing time, storage, energy consumption, and processing. These forms could be one or more at the same time. The offloading

task needs a medium to perform it; this medium is a communication between local machine and remote machine. This shows that the communication is the primary player to do the offloading task. On the contrary, local processing and communication consume the energy. As a result of this, there is a trade-off between handling the task locally and the offloading. It was recognized that energy saving is an important factor. The greatest power consumer in the mobile device is the network interface and the processor. So the offloading should be aware of the communication and the computation cost. According to Karthik et al., offloading can be performed for several reasons in general [2]. Firstly, to reduce the processing time and increasing the response time for the application. For instance Wolski et al., designed an algorithm for the offloading decision that aims to support code offloading and this helped in the selection of the wireless medium and this based on network bandwidth [3]. Secondly, offloading could increase the performance of an application because the results are better when handled on a powerful machine than handled on a limited computation power. Thirdly, the offloading is used for the load balancing between servers. Fourthly, the offloading technique helped the limited devices, for instance, smartphones to save energy Stephen et al., [4]. The offloading technique has not only one benefit but as well might provide more than one value at the same time. For instance, it might offer low latency as well as save energy.

2 Background

A lot of applications are highly computation intensive to execute on a mobile device and if the users want to perform such tasks, these applications must be on a cloud. Other applications such as gaming, voice recognition, could be performed on the mobile device but this will consume energy Karthil and Yung [5].
Amiya, et al., proposed models which help smartphones to be able to estimate the energy cost of task offloading. The validation of their models was done by using five mobile devices form different brands. The results showed that the new models accurately estimated the energy needed to offload tasks [6].

Hao and Daniel proposed system called Jade, which decreases down to 39% of power consumption for smartphone application besides enhancing the application performance. Jade is capable to offload tasks from an Android device to any Android or non-Android device. Multi-level offloading offers suitable way to transfer workloads to the suitable server depends on the energy requirement and the performance [7].

Altamimi, al., compared the energy weight for downloading and uploading a video file form and to multimedia cloud computing with the energy cost for the same video file on the smartphone, excused by using FTP and HTTP Internet protocols with WIFI and 3G network interfaces. The results illustrated that MCC offer for smartphones several kinds of functionalities as well conserve mobile devices energy from 30% to 70% [8].

Majid and Kshirasagar, developed decision engine on Android devices and an Amazon EC2 and S3 clouds to handle experiments to decide the cost of system parameters which is used by the decision engine. To convert video type, the offloading decision engine task decides whether to execute locally on the mobile device or to offload to the cloud [9].
Rajkumar and Tiago proposed an interface called Aneka interface which is contained on Mobile Client Library and helps to encapsulate the process such as sending a message, serializing and desterilizing message and gathering their responses of communication from mobile to cloud. There are two different mobile applications evaluated for the proposed architecture in terms of application execution time and the battery life of the mobile device [10].

Yaser et al., proposed a cloudlet-based application to reduce the network delay and power consumption for most common applications for instance multimedia applications. The experiment results show that using the new model helps to reduce the communication latency and power consumption of the smartphones [11].
Henri, et al., proposed a model called Cuckoo, for Android application to take a dynamic decision to offload a part of a task to the cloud or perform it locally. The cuckoo model could integrate with an advancement tool to perform both of them local and remote. The offloading decision is executed depends on the local resources such as memory, device type `and wireless medium [12].

Magurawalage et al. proposed intermediate layer architecture of cloudlet between cloudlet provider and mobile devices. They proposed a decision-making algorithm to decide whether to do offloading to a cloudlet or a clone. The new algorithm considered the energy consumption for task execution and the network status while satisfying certain task response time constraints including a data caching procedure at cloudlets for improving the all MCC performance [13].

Muhammad Shiraz et al. performed analysis of computational frameworks in respect with offloading in mobile cloud computing (MCC). The paper analysed the resource-intensive nature of the latest existing computational offloading techniques and he realised that 31.6% extra energy is consumed, 39% extra time is required and 13241.1 KB of data are sent for offloading [14].

3 Proposed Algorithm

Rich media needs an excellent mobile devices to handle the intensive computing tasks in a better way. However, mobile devices have limited resources and battery life, and this resource is one of the biggest obstacles for smartphone improvements. Offloading is one of the best suitable solutions to overcome battery life restriction and to reduce the energy consumption of smartphone. This technique supports mobile devices to perform the computations by sending the heavy tasks from the smartphone to the cloud computing. Our new system supports mobile users to send the task from

mobile to cloud taking into account the file size: i.e the file size will be measured in advance.

The established new model contains a decision engine to examine the feasibility of the offloading, in order to save mobile phone energy. New model investigates whether or not a mobile phone preserves energy by offloading to cloud computing. This technique depends on the file size. In this case, the model has two parameters to check. Firstly, it evaluates the time required to finish the task locally on the mobile and remotely on the cloud. Secondly, it evaluates the power consumption of doing the computations on the mobile or on the cloud. After finishing this process, the decision engine decides whether to offload or not.

After the decision engine determines the suitable solution based on the two parameters which are time needed to finish a task and power consumption, the mobile user starts doing the intensive computation on the smartphone, otherwise decides to perform offloading on the cloud. In this situation, decision engine has to check queuing and add the queuing latency to the total delay on the server and send an acknowledgment to users which includes the time required, power consumption, and queuing.

In this novel model, it is important to measure the power consumption and the time required for some of the tasks. As power consumption varies from 3G, 4G and WLAN interface networks, it is required to measure the weight of uploading and downloading files to or from cloud computing using different types of the network interface. This type of interface supplies the 4th parameters to the algorithm.

The mobile phones access the cloud through the Internet and the mobile phone applications which are connected to the cloud. A network interfaces such as (3G, 4G, and WLAN), where each of these network interfaces has its particular features. As a result, each of these types of network interfaces consumes an unbalanced level of power and offers different throughput. The new system checks all of these network interface and figures out the best method to save energy and to perform in a fast way.

The mobile phone consists of software and hardware. The hardware part includes RAM, HDD, CPU, and Wireless Network Interface (WNI). Software part contains the applications and the operating systems. The power is consumed in a mobile phone by some of the hardware parts, which are operated by the operating system which depends on the applications needs. As a result, the consuming of power depends on the application type, because each service provided requires specific hardware. Internet protocols such as (HTTP and FTP) and network interface are the main issues that impact the power consumption of these applications.

4 Experiments and Results

To test the execution time and power consumption on locally and by offloading, we have used Android Studio and Java programming language. The table below shows the results for different file size:

Table1: Execution time and power consumption for local and offloading

File Size	Local Node		Offloading	
	Execution Time(Minutes)	Power Consumption	Execution Time(Minutes)	Power Consumption
1 MB	1.94	.7%	4.16	. 8%
2 MB	2.79	1.1%	4.07	.1%
5 MB	6.69	2.3%	5.83	1.3%
7 MB	9.87	4.8%	6.33	1.9%
9 MB	14.05	6.01%	7.2	2.5%

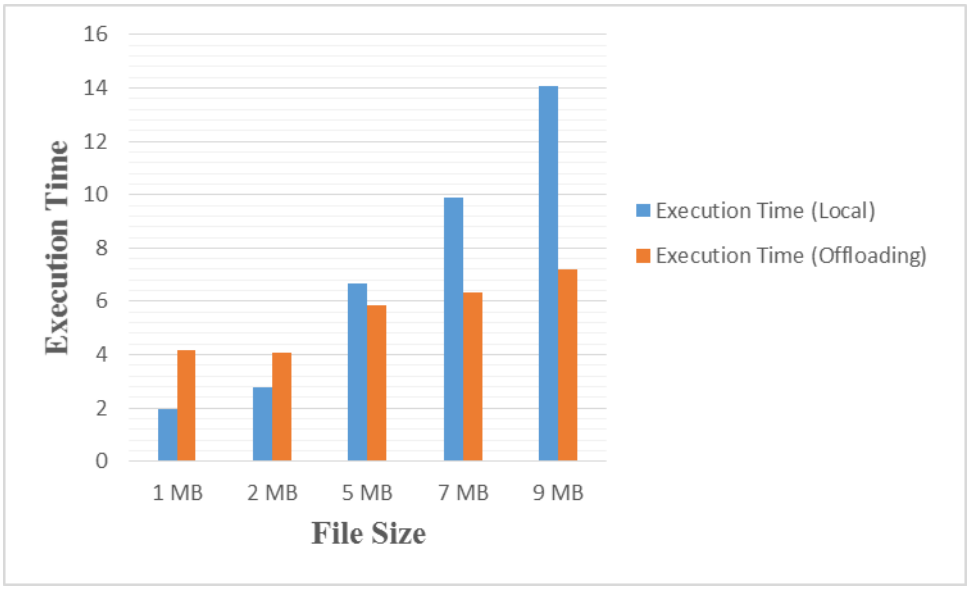

Figure 1: Execution time for local and offloading:

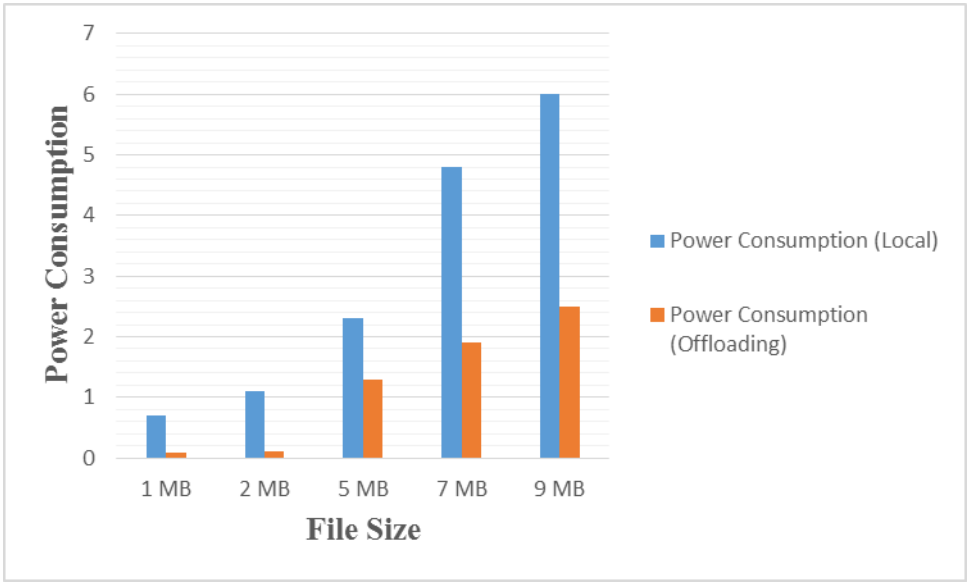

Figure 2: Power consumption for local and offloading

The first experiment of the algorithm in figure 1 has considered the relation between the file size and the execution time either in the local node or in the offloading cloud core. The results show that the more the file size is, the more the execution time will be. The file size was tested between 1MB and 9 MB. From the results it can be seen that the increasing in the file size put high pressure on the execution time of the node, while much less pressure on the offloading core computer. Which is a proof of the hypothesis, for example at file size = 9MB the execution time is doubled in the local load.

In Figure 2 the power consumption is more than double at file size =9 MB. Therefore it is very paramount to use this algorithm to take a good decision to save time and power.

5. Conclusion:

A new algorithm to compute the file size, the duration of execution and the power consumption has been established and tested. The experiments included different size file and a comparison was performed. After each test, it can be decided either to execute the file locally or to execute the file by the offloading to a cloud core unit. From the results, it is noted that this is a big jump in the execution time and the power consumption for big files but it is noted that in these circumstances it is vital to

execute files on the cloud core by offloading. There is in this case a big reduction in the power consumption. For further research topic, it is suggested that other quality parameters have to be involved such as queuing and utilization.

Acknowledgment:

The authors of this work than the UK-India Educational and Research Partnership for sponsoring this research under the collaboration UKIERI Project between Anglia Ruskin University in the UK and The ABV-Indian Institute of Information Technology and Management in Gwalior, India.

References

1. Zhou, B., Dastjerdi, A.V., Calheiros, R.N., Srirama, S.N. and Buyya, R., 2015, June. A context sensitive offloading scheme for mobile cloud computing service. In Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on (pp. 869-876). IEEE.
2. K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, \A Survey of Computation Offloading for Mobile Systems," Mobile Networks and Applications, vol. 18, no. 1, pp. 129{140, 2013. 31, 33
3. Wolski, R., Gurun, S., Krintz, C. and Nurmi, D., 2008, April. Using bandwidth data to make computation offloading decisions. In Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on (pp. 1-8). IEEE.
4. Gao, B., He, L., Liu, L., Li, K. and Jarvis, S.A., 2012, September. From mobiles to clouds: Developing energy-aware offloading strategies for workflows. In Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing (pp. 139-146). IEEE Computer Society.
5. Kumar, K. and Lu, Y.H., 2010. Cloud computing for mobile users: Can offloading computation save energy?. Computer, (4), pp.51-56.
6. Altamimi, M., Abdrabou, A., Naik, K. and Nayak, A., 2015. Energy cost models of smartphones for task offloading to the cloud. Emerging Topics in Computing, IEEE Transactions on, 3(3), pp.384-398.
7. Qian, H. and Andresen, D., 2015, June. Reducing mobile device energy consumption with computation offloading. In Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on (pp. 1-8). IEEE.
8. Altamimi, M., Palit, R., Naik, K. and Nayak, A., 2012, June. Energy-as-a-Service (EaaS): On the efficacy of multimedia cloud computing to save smartphone energy. In Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on (pp. 764-771). IEEE.
9. Altamimi, M. and Naik, K., 2014, June. A Practical Task Offloading Decision Engine for Mobile Devices to Use Energy-as-a-Service (EaaS). In Services (SERVICES), 2014 IEEE World Congress on (pp. 452-453). IEEE.

10. Justino, T. and Buyya, R., 2014, October. Outsourcing resource-intensive tasks from mobile apps to clouds: Android and aneka integration. In Cloud Computing in Emerging Markets (CCEM), 2014 IEEE International Conference on (pp. 1-8). IEEE.
11. Jararweh, Y., Ababneh, F., Khreishah, A. and Dosari, F., 2014. Scalable cloudlet-based mobile computing model. Procedia Computer Science, 34, pp.434-441.

12. R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: A Computation Offloading Framework for Smartphones," in Mobile Computing, Applications, and Services, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, M. Gris and G. Yang, Eds. Springer Berlin Heidelberg, 2012, vol. 76, pp. 59–79. 34, 35

13. Magurawalage, C.M.S., Yang, K., Hu, L. and Zhang, J., 2014. Energy-efficient and network-aware offloading algorithm for mobile cloud computing.Computer Networks, 74, pp.22-33.
14. Shiraz, M., Sookhak, M., Gani, A. and Shah, S.A.A., 2015. A study on the critical analysis of computational offloading frameworks for mobile cloud computing. Journal of Network and Computer Applications, 47, pp.47-60.
15. Shiraz, M., Gani, A., Shamim, A., Khan, S. and Ahmad, R.W., 2015. Energy efficient computational offloading framework for mobile cloud computing.Journal of Grid Computing, 13(1), pp.1-18.