IJCSN - International Journal of Computer Science and Network, Volume 8, Issue 3, June 2019
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

An Energy Efficient Task Offloading for Mobile Cloud Environment

[1]Nancy Arya; [2] Sunita Choudhary; [3] S. Taruna

[1] Department of Computer Science, Banasthali Vidyapith
Rajasthan, India

[2] Department of Computer Science, Banasthali Vidyapith
Rajasthan, India

[3] Department of Computer Science, JK Lakshmipat University
Rajasthan, India

Abstract - In the last few years, the technology of mobile computing has gained tremendous popularity and changed users mind for computing. However, smart phones are constrained computing devices which are facing number of issues related to resources such as memory, storage, computation power and shortened energy. To overcome these constraints, offloading provides natural solution for mobile cloud environment by migrating the intensive problems to cloud servers. However, conventional frameworks of offloading lack in considering the dynamic execution time as well as they have not focused on extra overhead of runtime migration. This paper proposed a new approach for runtime offloading to achieve better performance and energy optimization.

Keywords - *Mobile Computing, Offloading, Runtime Offloading, Energy Optimization.*

1. Introduction

Mobile Cloud Computing technology enables the integration of cloud computing environment with smart mobile phones to make these devices asset full in terms of resources such as storage, power and memory. It is a framework that allows to move the applications from smart phones to cloud servers. According to Juniper Report, the mobile applications have increased around 88% every year from 2009 to 2015 [20]. Now, mobile devices or smart phones have capability to run large number of complex applications, mostly needs high computing power and high energy consumption such as video call, internet banking, uploading and downloading, use of GPS etc. [8]. As per study conducted in (2005), power capacity is the most required element for mobile devices such as smart phones and laptops [18]. It is a constraint in the development and execution of mobile applications.

Charging points are not always available especially when the person is on the move. Sometimes, to conserve battery, users cannot use the device completely which hinders the best possible use of smart phones. Energy saving is also important from the perspective of green environment. However, phone manufacturers provide different options to save the battery like reduce the brightness of phones, turning off the network connection etc. But these solutions cannot solve the issue completely.

To overcome these drawbacks, computational offloading is a better solution which shifts the intensive codes for remote execution on cloud which is resource-full environment [1,2]. This technique is helpful to speed up the computation power as well as to optimize the energy consumption of limited power devices. But, in offloading, the decision that which component needs to be offload plays major role. It can depend on number of parameters such as type and size of applications, storage capacity and available bandwidth [5]. If these parameters of decision are defined at development time then it is known as static offloading and if these decision parameters are defined at run time known as dynamic or runtime offloading.

Many offloading frameworks exist to optimize the energy as well as the performance for mobile cloud computing [6,7,13,14,17,18,20-22]. However, every framework has its own benefits and limitations. Mostly frameworks have not considered dynamic execution time which is a major aspect for real time applications. Some of the frameworks has not focused on cost to transfer the data at runtime. This paper proposed a new technique for runtime offloading to optimize the energy and to improve the performance. The main goal of this technique is to find the optimum solution to offload intensive parts of the applications with proper use of assets. However, this works on the decision that which parts should offloaded, how and where they offloaded. The results of this proposed work compared with existing framework [20] and shows better energy efficiency as compared to existing framework.

This paper is classified into following sections. Section II is Related Work for task offloading in Mobile Cloud Computing. Section III describes Proposed Work with features. Section IV presents discussion of Results and Performance Evaluation. Finally, Section V presents the Conclusion of the work.

2. Related Work

Mobile Cloud Computing provides the facility to extend the resources for the mobile device over the network and integrates the advantages of mobile computing and cloud computing [12]. Author presents in [9] the several features of the combined use of mobile computing and the cloud computing. These smart phones have some issues such as limited power, bandwidth, limited memory capacity and issues related to quality of services etc. [9,16]. Many approaches has been designed to save the energy by improving the computation power and proper controlling of the display and storage [10,15,19]. But these modifications need lot of changes in the hardware and increases the cost.

Task offloading provides good option to resolve the issues of asset poverty by shifting the complex computations from mobile device to remote cloud server. Many task offloading models [7,14,17,18,20,21] exist which focused on the power saving and energy optimization. Energy can conserve by transferring the computation to the remote server. But the most important, the application should consume less energy while migrating [11,19]. One of the model is MAUI [7]. It works on the task offloading to save the overall energy for the applications of mobile games in cloud-based environment. It saves around 27% energy for the mobile games application. But it works on the saved data of offloading decisions. It does not consider any new independent task. One of the effective model for the offloading decision is proposed in [3] works on size of files. It gives the benefit for energy efficiency, only if the file sizes are more than 4 MB. The µcloud model [6] is energy aware framework supports decoupled self-contained application modules. However, this model can compute only a single partition at a time as well as it provides no security for the exchanged data between modules. One of the energy aware framework EECOF proposed in [20] works for the computation of complex applications by utilizing the services of cloud models with the runtime transfer of less number of parts of the application. This framework used the problem of sorting and multiplication of matrices to perform the experiment and to verify the results. The model has compared with existing frameworks and save energy by 69% by reducing the data transmission size approximate 84%.

MobiCloud [4] is one of the model that supports adhoc networks for the service oriented architecture by considering each node as a part of service node over the network and create copies of these nodes in the cloud. A runtime offloading model has proposed in [13] based on adaptive partitioning for choosing the category of cloud such as central clouds and edge cloud for offloading. The objective of this model to reduce the execution cost while offloading. One of the framework is Compss [14] proposed for the applications of high performance in a distributed environment which consumes minimum energy. It supports a transparent model to write the android code for frequently using applications. Then, these applications are migrated for the processing on cloud. One of the framework is proposed in [22] works for the risk evaluation for offloading of tasks. It validates the different parameters of offloading on game theory. Multiplication of matrices problem is used for the offloading decision.

3. Proposed Work

The proposed work used the existing offloading framework proposed in [20] as a benchmark. The existing framework [20] has been focused only on energy consumption and data transmission size. But it has not taken into account of extra overhead of a given task at runtime migration as well as the dynamic execution time which plays an important role for the applications of real time.

The proposed work defines the technique which takes the decision of offloading on the basis of runtime and energy requirements parameters of a task. The proposed work is an attempt to optimize the energy and execution time during offloading by introducing new parameters of offloading decision. The objective of the work is proper selection and offloading of intensive tasks from mobile to the cloud server on the basis of estimated values of energy and time. It works on IaaS (Infrastructure-as-a-Service) and SaaS (Software-as-a-Service) to optimize the energy. The particular technique considered in this proposed work where a benchmarking is performed and estimate the values for the required execution time and energy to achieve the better results. It takes the decision that given task is to either compute on local device or offload to the remote server. The proposed work is validated by experimental setup involving an android phone Xiaomi note 4 with processor of snapdragon 625 octa core, 4100 mAh battery and 4 GB of RAM and the host intel core i5 system of 8 GB of RAM running operating system ubuntu and the cloud simulator docker to provide virtualization.

IJCSN - International Journal of Computer Science and Network, Volume 8, Issue 3, June 2019
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

The overall architecture of the proposed work is shown in Fig. 1. It consists of two components - a mobile device (client) and cloud server.

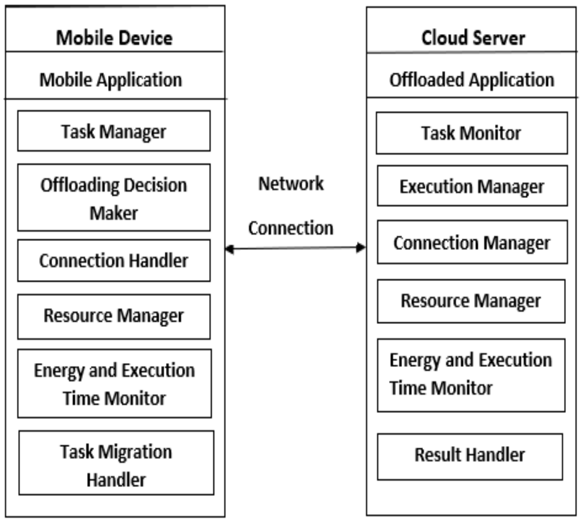

Fig. 1 Architecture of proposed work

The mobile device can be any smart phone like Android, iOS etc. These mobile devices are portable and constrained devices. However, these devices are capable to run large number of applications. The mobile device environment consists of another several components. The first important component is Task Manager which handles the computing tasks requested by the client. It is also responsible for benchmarking to estimate the values of required time and energy. The second component is Offloading Decision Maker which is responsible to take the offloading decision on the basis of input parameters and decide the execution location of the computational task. It also interacts with other subcomponents to make appropriate decisions for offloading. The next component is Connection Handler which handles the network connections and provides the control to the cloud environment. Another subcomponent is Resource Manager which provides the required resources to the computation and offloading operations. The next subcomponent is Energy and Execution Time Monitor. It is responsible to estimates the required energy and execution time for a given task. It also interacts with offloading decision maker to take the optimum offloading decision. The last subcomponent is Task Migration Handler which handles the process which is initiated after offloading decision.

The cloud server environment, consists of several remotely located computing resources which is available on demand. The cloud server environment also consists of various subcomponents. The first important subcomponent of cloud server environment is Task Monitor which manages the process of task migration and it also manages the various input output functions. The second component is Execution Manager responsible to receive the computational tasks from the mobile client and scheduled them for execution appropriately. It can also request for additional resources for execution on the cloud server if there is a requirement for it. Connection Manager on the cloud server manages all the connections between distributed remote locations. This network connectivity is transparent to the other components. Another subcomponent of cloud environment is Resource Manager responsible to monitor the usage of resources required by different operations. There is corresponding Energy and Execution Time Monitor on the cloud as well which estimates the actual consumption of energy and execution time for a given offloaded task. However, energy is not a constraint on cloud side environment. The final subcomponent is Result Handler manages the generated results for a given offloaded task on cloud and then returns back to the mobile client.

Experiment has carried out by executing the task on local environment at initial stage and then the proposed work has employed for offloading the task on cloud. The proposed technique takes the computational task of size 10^n where n=7, then the benchmarking has performed for the given task and on the basis of results, average values are estimated for required computing time, transmission time, time for uploading and downloading, required energy, energy used for transmission. Benchmarking coefficients are also calculated for local and cloud environment to take the decision for execution either on mobile device or migrates to the cloud. Selection of the computational task is also an important parameter. The criteria of selection for the proposed technique is time complexity. Matrix multiplication is selected as an experimental task because the complexity of matrix multiplication is high $O(n^3)$.

Execution time is the first important parameter to take the task offloading decision. It has calculated for both environments local and cloud. The total time required for execution on local device:

T_{loc} = *time required in reading + time required in writing to memory + time required to executing the task on local device.*

The total execution time to offload the task on cloud:

T_{cld} = T_{loc} + *time required in data transmission + time required for execution on the cloud + time required to return the results.*

In case, execution time is same for local and cloud environment then the environment which needs less energy is consider.

IJCSN - International Journal of Computer Science and Network, Volume 8, Issue 3, June 2019
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

The next important parameter is the required energy. The total required energy for execution on local device:
E_{loc} = *energy required in data reading + energy required in writing data to memory + energy required in execution state.*
The total required energy for execution on cloud:
$E_{cld} = E_{loc}$ + *energy required in data transmission.*

Now, these above parameters T_{loc}, T_{cld}, E_{loc}, and E_{cld} has calculated for the experimental task of matrix multiplication. If the value of T_{loc} is less than or equal to T_{cld} and energy E_{loc} is less than E_{cld} then the task executes on local environment otherwise on cloud. The proposed technique utilized the concept of parallel read and transmit operations to provide augmented energy efficiency and performance during the execution of task.

4. Results and Performance Evaluation

The proposed work validates the application components for both of the environment mobile and cloud by carried out the experiments. The experiment has performed for matrix multiplication application for the different size of data. The value of execution time and energy are estimated for the mobile environment and cloud by proposed work. Fig. 2 shows the increment in the value of execution time with the increment in size of data. The figure shows the percentage of reduction in the execution time for offloading by proposed technique on cloud environment as compared to mobile environment. It is clear that proposed technique provides better results for the small size computation also such as 150*150 and 250*250 as the matrix multiplication is the application of high complexity O(n^3). In case of large matrix 1000*1000, achieved benefit is around 93.1% for execution time.

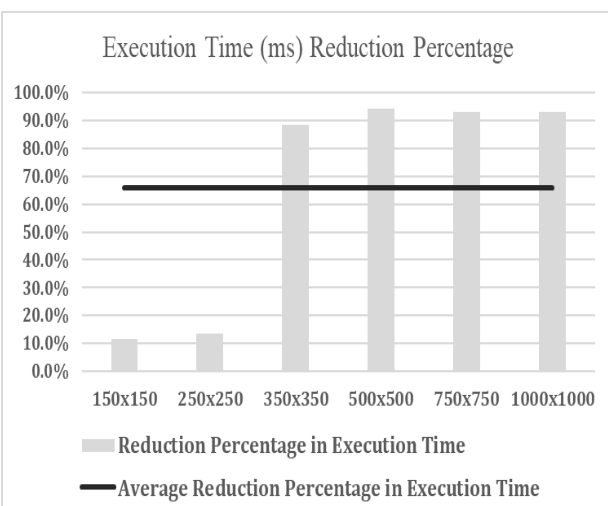

Fig. 2 Execution time reduction graph.

According to Fig. 3, the energy increases exponentially with the increment in data size. The difference of energy increases initially for the data sets of small size and then saturates at approximate 94.5% on the matrix of 500*500 which shows excellent gain. After then for 750*750 and 1000*1000, it has reached on optimum value at around 96.2%.

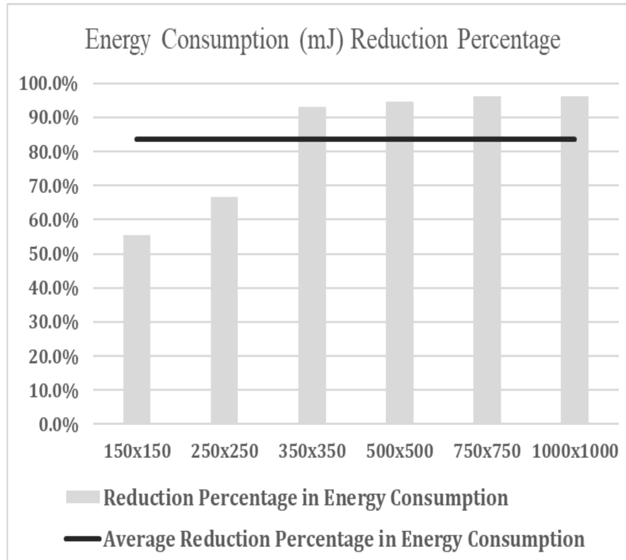

Fig. 3 Energy consumption reduction graph.

Thus, the complex computation such as matrix multiplication gives the immediate advantage of offloading for the small size of matrices also. It is clear that the proposed work shows proper migration of computational components from local device to cloud server. It can save the energy of mobile device as well as augment the performance.

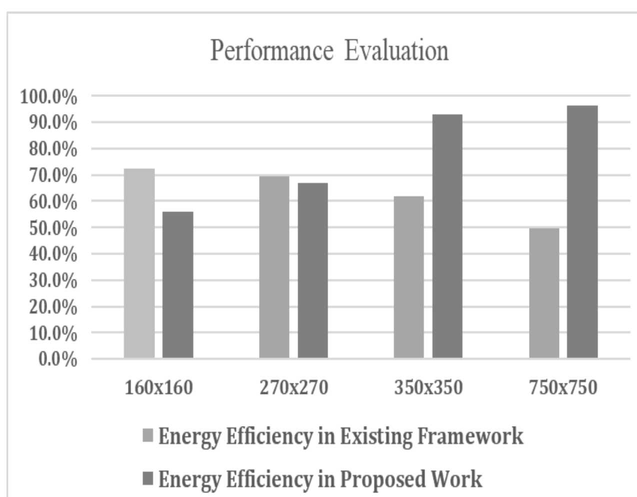

Fig. 4 Performance evaluation graph.

The proposed technique provides the better results for the execution time and energy efficiency by offloading the computation to the remote server. However, when this efficiency compared with earlier framework [20], the results are shown by the chart in above Fig. 4.

Thus, it is clear that the proposed technique provides efficient results for the complex computational tasks. As Fig. 4 shows the comparison of results with existing framework [20], for small size matrices, the difference is small or in few cases the computation consumes high energy in proposed technique but the size of more than 300x300 matrix, the efficiency of proposed work has been clearly better than the efficiency of existing framework.

5. Conclusion

Traditional models have worked to speed up the execution and save the energy for local mobile devices. It is very important for the resource aspects. However, every model has its own benefits and limitations. In the paper, the author has proposed a novel technique for offloading the computational tasks by improving the performance and efficiency of energy. This work used the concept of benchmarking of the computational task before actual execution of the task and on the basis of the estimated value for time and energy, the final decision has taken for offloading. The decision depends on less energy consumption. The results of experiment clearly shows that the proposed work gives better results for energy efficiency and performance. The work has carried out by the execution of complex application of high complexity such as matrix multiplication. For the large matrix size, achieved efficiency around 93.1% for execution time and 96.2% for energy which proves the higher efficiency. The results of the proposed work are then compared to the results of existing framework and shows better efficiency in energy optimization. Thus, it is verified that including the criteria of execution time with energy parameter provides an efficient way as compared to conventional methods which worked only for energy parameter only.

References

[1] Abdullah Gani, and Han Qi, “Research on Mobile Cloud Computing: Review, Trend and Perspectives”, Digital Information and Communication Technology and it's Applications (DICTAP), Second International Conference, 2012, pp. 195-202.

[2] Abdullah Gani, Ejaz Ahmed, Rajkumar Buyya, Saeid Abolfazli, and Zohreh Sanaei, "Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges", IEEE, 2013.

[3] Aldmour, Rakan, and Yousef, "New cloud offloading algorithm for better energy consumption and process time", International Journal of System Assurance Engineering and Management 8.2, 2017, pp. 730-733.

[4] Mushtaq Ali, and Gran Badshah, "Mobile cloud computing & mobile's battery efficiency approaches: A Review", Journal of Theoretical and Applied Information Technology 79.1, 2015, pp. 153-175.

[5] Antti P. Miettinen, and Jukka K. Nurminen, "Energy efficiency of mobile clients in cloud computing", 2011.

[6] Bu Sung Lee, Erwin Leonardi, George Goh, Markus Kirchberg, Verdi March, and Yan Gu, “µcloud: towards a new paradigm of rich mobile applications”, Procedia Computer Science, Vol. 5, 2011, pp. 618–624.

[7] Cuervo, and Eduardo, "MAUI: making smartphones last longer with code offload", Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM, 2010.

[8] Dhammapal Tayade, "Mobile Cloud Computing: Issues, Security, Advantages, Trends”, IJCSIT, Vol. 5, 6635-6639, ISSN: 0975-9646, 2014.

[9] Forman, George H., and John Zahorjan, "The challenges of mobile computing”, 1994, pp. 38-47.

[10] Gran Badshah, Jasni Mohamed Zain, Mohammad Fadli Zolkipli, and Mushtaq Ali, "Mobile Cloud Computing & Mobile Battery Augmentation Techniques: A Survey", IEEE, 2014.

[11] Kumar, Karthik, and Yung-Hsiang Lu, "Cloud computing for mobile users: Can offloading computation save energy?", Computer 43.4 2010, pp. 51-56.

[12] Liu, Leslie, Randy Moulic, and Dennis Shea, "Cloud service portal for mobile device management", e-Business Engineering (ICEBE), IEEE 7th International Conference on, IEEE, 2010.

[13] Liu, Xing, Songtao Guo, and Yuanyuan Yang, "Task Offloading with Execution Cost Minimization in Heterogeneous Mobile Cloud Computing", International Conference on Mobile Ad-Hoc and Sensor Networks. Springer, Singapore, 2017.

[14] Lordan, Francesc, and Rosa M. Badia, "Compss-mobile: Parallel programming for mobile cloud computing", Journal of Grid Computing 15.3, 2017, pp. 357-378.

[15] Paulson and Linda Dailey, "Low-power chips for highpowered handhelds", Computer 1, 2003, pp. 21-23.

[16] Rahimi, M. Reza, Nalini Venkatasubramanian, and Athanasios V. Vasilakos, "MuSIC: Mobility-aware optimal service allocation in mobile cloud computing", 2013 IEEE Sixth International Conference on Cloud Computing, 2013.

[17] Rudenko, and Alexey, "Saving portable computer battery power through remote process execution", ACM SIGMOBILE Mobile Computing and Communications Review 2.1, 1998, pp. 19-26.

[18] Saad, S., and S. Nandedkar. "Energy Efficient Mobile Cloud Computing." International Journal of Computer Science and Information Technologies, 2014.

[19] Saraf, Shweta B., and Dhanashri H. Gawali "IoT based smart irrigation monitoring and controlling system", Recent Trends in Electronics, Information & Communication

Technology (RTEICT), 2nd IEEE International Conference, 2017.

[20] Muhammad Shiraz, Abdullah Gani, and Suleman Khan, "Energy efficient computational offloading framework for mobile cloud computing", Journal of Grid Computing 13.1, 2015, pp. 1-18.

[21] Smailagic, Asim, and Matthew Ettus, "System design and power optimization for mobile computers", VLSI, Proceedings of IEEE, Computer Society Annual Symposium on IEEE, 2002.

[22] Wu, Huijun, and Dijiang Huang, "Modeling multi-factor multi-site risk-based offloading for mobile cloud computing", 10th International Conference on Network and Service Management and Workshop, IEEE, 2014.

First Author Nancy Arya is a Ph.D. scholar in Computer Science & Engineering at Banasthali Vidyapith, Rajasthan. She has completed her M.Tech. in Computer Science and Engineering from Jagannath University, Jaipur in 2013 and MCA from Rajasthan Technical University, Kota in 2010. She has 4 years of teaching experience and 2 years of industry. She has published several national and international papers and awarded by many awards. Her research interests includes Networking and Cloud Computing.