IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-6, June 2019 E-ISSN: 2347-2693

Energy Aware Offloading in Cloud Assisted Mobile Environment

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

[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

Corresponding Author: nanarya1@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i6.203208 | Available online at: www.ijcseonline.org

Accepted: 12/Jun/2019, Published: 30/Jun/2019

Abstract— In the last one decade, the utilization of mobile technology has increased tremendously. The development in the mobile technology is leading to the innovations in the smart phones. Now, smart phones are not only used for communication but also capable to support number of complex applications, mostly requires high computational power. Although these smart phones have several resource issues related to power capacity, data transfer, storage capacity, security and limited processing power etc. Power capacity is one of the most crucial issue of the current smart phones. However, mostly mobile devices emphasize on many power management functions to save the power but the overall saving is much less than actual requirement. In this situation, Mobile Cloud Computing provides good option of task offloading where any intensive computations are migrated to the remote servers or cloud. Mobile Cloud Computing is a new network paradigm which merged the features of mobile computing and cloud computing to overcome the issues of mobile devices. It can also be a power saving paradigm which needs serious attention. Thus, this paper proposed a novel approach for energy aware task offloading for mobile cloud environment.

Keywords— Mobile Technology, Power Capacity, Mobile Cloud Computing, Task Offloading, Cloud Computing, Energy Aware Task Offloading.

I. INTRODUCTION

Cloud Computing is a popular trend in the field of networking that provides the facility to move the computing and data away from laptops and other portable devices [24]. The maturity of this technology is leading many enterprises to shifting their IT infrastructure to cloud. It offers many benefits in terms of accessibility of data, minimum cost, reliability, portability and unlimited storage [20]. In addition of the technology of cloud computing, Mobile Cloud Computing is an emerging trend which is the integration of mobile computing and cloud computing technology to make the constrained devices such as mobile phones resource full. It offers many features over mobile computing. It is a framework where resources are located remotely to the mobile devices. By Juniper study, from 2009 to 2015, the users of mobile applications have increased rapidly approximate 88% every year [15]. Now a days, mobile phones have enough capability to execute high range of complex applications, which requires more processing power and high energy consumption like video call, browsing, GPS, uploading, downloading and online shopping etc. [6]. Smart phones with limited processor capacity and limited memory can restricts the actual use of the device. Another drawback is power capacity of mobile devices. Power capacity is the most desirable feature in mobile devices. But mostly times it becomes the constraint to process mobile applications. It also restricts the best possible utilization of mobile devices. Lithium-ion batteries are capable to supply only few hours of energy. Charging points are not available every time especially when the person is moving. Thus, energy conservation is an important perspective in terms of resources as well as green environment. However, manufacturers are providing various solutions to conserve the energy such as low brightness, turning off the mobile data connection etc. But these options are not sufficient to resolve the issue completely.

To resolve these issues, the process of task offloading or computational offloading provides good solution by offloading the resources demanding tasks to the resource rich environment of remote cloud server [1],[2]. This process is very helpful to boost up the computation capacity as well as to conserve the energy of mobile devices. Offloading can be static or dynamic. In static offloading, the decision-making parameters of offloading such as size of applications, available memory and storage, type of applications etc. are defined at the time of development whereas in dynamic offloading, these parameters are defined at the time of execution [3],[16],[22]. The decision of offloading also depends on what components needs to be migrated, how and where components are migrated.

Several research works exist to save the energy for mobile cloud computing environment. Mostly frameworks have not considered performance which is also an important aspect for mobile applications. Traditional frameworks used adaptive algorithms to offload the serious computations on cloud. Some research works has not focused on the transfer cost of data at runtime. This paper proposed a new approach for dynamic task offloading to save the energy and to augment the performance. The main objective of this work is to achieve the best solution to offload the most intensive parts of the mobile applications with proper utilization of resources.

This paper is classified into following sections. Section II contain the related work for computational offloading in mobile cloud environment. Section III explains the methodology of proposed work with architecture and features. Section IV is discussion of results. Finally, Section V concludes the work with future scope.

II. RELATED WORK

Mobile Cloud Computing provides the framework to extend the assets for the mobile device over the communication network [11]. Authors discussed in [9],[26] the various advantages of the integrated use of mobile computing technology and the cloud computing technology. These mobile devices have some constraints related to energy, available bandwidth, less memory capacity, security and quality of services etc. [9],[18],[26]. Several research works have been done to conserve the energy by increasing the processing power and management of the display [8],[17],[22]. However, these changes require various modifications in the hardware and thus, the cost increases.

Computation offloading is the technique that provides the advantage to resolve the challenges of resource poverty by transferring the complex and large computations from mobile to cloud server. Many offloading approaches and frameworks [4],[5],[7],[13],[15],[16],[19],[23],[25] are in existence which emphasized on energy optimization. However, energy can save by shifting the computation to the cloud. But the computational application must consume limited energy while transferring because sometimes the overhead of sending the files on network is too high [10],[21]. MAUI model [5] designed for power saving especially for the games' applications in cloud environment. It optimizes 27% energy for the gaming application. But the drawback is the model used only saved data to the decision making of offloading. One of the approach for the task offloading is proposed in [16]. It works for high complexity problems and saves around 93% execution time and 96% energy which shows the better energy conservation and performance. Another model is µcloud [4] that works for decoupled application partitions. But at a time, it can compute single partition only. EECOF framework [15] proposed for the processing of large applications by using the cloud service models with the run-time migration of minimum partitions of the application. It conserves energy around 68% by compressing the size of data transmission by 84%. A dynamic offloading framework proposed [12] for adaptive partitioning to select the cloud category like edge cloud for task offloading. The goal of this model is to reduce the overall cost while offloading the tasks. Compss [13] framework is designed for the high-performance applications in a distributed environment. It supports the applications of minimum energy consumption. It is a transparent model that provides the facility to write the android code for applications. Then, these applications are transferred for the computation on cloud. This framework used merge sort application to validate the power saving. One of the model is MobiCloud [14] designed for ad-hoc networks and service-oriented architecture. It treats every node of the network as service node and create multiple replicas of the nodes on the cloud. Another framework [25] proposed for the offloading risk evaluation. It works on the several parameters related to offloading game theory. Matrix multiplication application is used for the validation of offloading decision making. Authors in [7] proposed a dynamic offloading algorithm to analysis the computations and transfers only most intensive methods to cloud to save the power. The algorithm has used quick sort application and word count to achieve the algorithm efficiency.

III. METHODOLOGY

The objective of this proposed work is to explore the optimum energy aware method to offload applications with intensive computation and resource requirements by introducing novel parameters. The proposed work also supports the concept of partitioning to equally partitioned the intensive task across mobile and cloud environments to optimize the overall energy. The decision for selection and offloading of tasks from mobile to the cloud depends on their calculated energy. The work employs cloud service models IaaS (Infrastructure-as-a-Service) and SaaS (Software-as-a-Service) to optimize the overall energy. The approach considered in the work used the concept of benchmarking to estimate the value of consumed energy. The decision that allotted computational task is to compute on mobile device or cloud depends on the calculated value of consumed energy. The methodology includes experimental validation of the work through a simulated experiment that includes a setup of android mobile device Xiaomi note 4 with snapdragon 625 octa core processor, battery of 4100 mAh and RAM of 4 GB and a cloud simulator on a virtual server.

The cloud environment has been simulated by a machine running docker to provide virtual environment.

The architecture of the proposed work is shown in Figure 1. It made up of two components local device (mobile) and remote server (cloud).

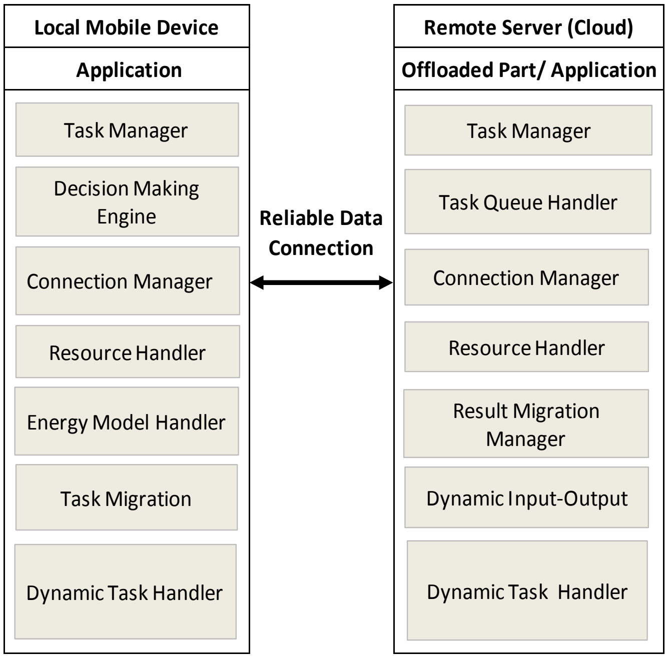

Figure 1. Architecture of Proposed Work

The mobile device or smart phone can be device like Android, Windows, iOS etc. The mobile device environment is made up of several sub components. The first component is Task Manager responsible to handle the computing tasks requested by the user. It manages the process of benchmarking to calculate the value of required energy. The second sub-component is Decision Making Engine is responsible to take the appropriate decision for offloading by using the estimated values of input parameters. It also decides the actual execution location of the allotted complex task. It also communicates with other subcomponents to take the required decisions. The next component is Connection Manager which manages the network connections and control part to the cloud. It handles the request and manages the required cloud resources on demand. Another sub-component is Resource Handler which manages and allots the various resources to the computation and offloading process as per request. The other subcomponent is Energy Model Handler is responsible to calculate the consumed energy for a given particular task. It also communicates with decision making engine to take the optimum decision for offloading. Task Migration Handler is responsible to examines the input tasks and evaluates these tasks can be subdivided into sub tasks which can be further distributed across the mobile and cloud server. Once decisions are taken, then the tasks are migrated to appropriate locations for computation and the received results are merged together to provide the final outcome to the user. The last sub-component is Dynamic Task Handler which manages multiple parallel activities like read and transmit operations.

The remote server (cloud) environment is resource full environment also consists of several subcomponents. The first subcomponent of cloud environment is Task Manager manages the activity of task migration and it is also responsible for the various input and output functions. Another component is Task Queue Handler responsible to receive the tasks from the mobile device and scheduled them for execution. It can also demand the resources for execution on the cloud if there is a need. Connection Manager on the cloud server is responsible to handle the connections across distributed remote locations and these connections are transparent to the other execution engines. The next subcomponent of cloud is Resource Handler which monitors the utilization of resources demanded by other operations. Another subcomponent is Result Migration Manager handles the obtained results for the offloaded task on cloud and then return these results back to the mobile device. Dynamic Input-Output is the component of cloud application handles all the input-output files on behalf of the cloud application. There is corresponding Dynamic Task Handler on the cloud which manages multiple parallel activities like read and transmit operations done by the cloud.

Research experiment has carried out by executing the computational task on mobile environment at first stage and on later stage, the proposed work has employed for offloading the intensive task partially or completely on cloud server. The computational task of size 10^7 is selected for experiment. Benchmarking has performed for a selected size of task to calculate the average values of consumed energy, energy required for data transmission etc. By using these values, benchmarked coefficients are estimated for local and cloud environment to take the decision of offloading. Selection of the task is also an important decision. In the work, sorting algorithms (merge sort and quick sort) is selected as intensive tasks because these algorithms are quite efficient as compared to other sorting algorithms. Another reason is these algorithms works on the divide and conquer strategy and can be partitioned into multiple sub problems. The size of sub problems depends on the computational capacity of the local mobile device. Then, these subproblems executed simultaneously on local and remote environments and then the results from respective environments are merged to achieve final outcome. Hence the process of partitioning augments the overall energy efficiency of the task.

Energy consumption is the important parameter to take the decision for offloading. It has calculated for both mobile and cloud environment. The total energy required for execution on mobile device:

E_{mob} = consumed energy in data reading + consumed energy in writing data + consumed energy in activated execution state.

The total energy for execution on cloud server:

E_{cd} = E_{mob} + consumed energy in data transmission on network.

The total energy involved in partitioning:

E_{part} = E_{cd} + consumed energy to partition the tasks + consumed energy for combining the results.

Now, these above parameters E_{mob}, E_{cd}, and E_{part} has calculated for the allotted sorting task. The environment with less energy requirement would prefer for execution. If the value of energy E_{mob} is less than E_{cd} and E_{part} then the task executes on mobile. If the value of energy E_{cd} is less than E_{mob} and E_{part} then the task executes on cloud server environment. Otherwise, partition the tasks into T_x and T_y. Then, execute the T_x on mobile and T_y on cloud and combine the results. The partitioning approach used in the work balances the tasks to mobile and cloud server environment for simultaneous execution. By proposed partitioning algorithm, $T_x = N*BC_{mob} / (BC_{mob}+BC_{cd})$ and $T_Y = N\text{-}T_x$. Here, BC_{cd} is benchmark coefficients for cloud environment and BC_{mob} is benchmark coefficients for mobile environment and the final outcome is the combination of results achieved from T_x and T_y. It is also possible that one of these sets may be empty. The proposed approach works on parallel read and transmit operations during the execution to optimize the energy and performance enhancement.

IV. RESULTS AND DISCUSSION

The proposed work evaluated the computational task for both mobile and cloud environment by carried out different experiments. The experiment has done for sorting applications like merge sort and quick sort of different data size. The consumed energy is estimated initially for the mobile device and then for cloud server by proposed work.

Initial experiment has done for merge sort which is a popular sorting algorithm. Figure 2 presents the energy consumption in mJ unit for different size of data for the application of merge sort. As shown by figure, the energy consumption is increasing along with the increment in data size. Merge sort is a problem of low time complexity. Thus, the results are good for small data size also on the mobile itself. But as data size increases, the overhead of sending the data to cloud through network is balanced by the achieved increased efficiency in computation at cloud server. The average size for which the offloading on cloud is beneficial at around 3500000. Hence, there is an advantage in parallel execution on mobile and cloud.

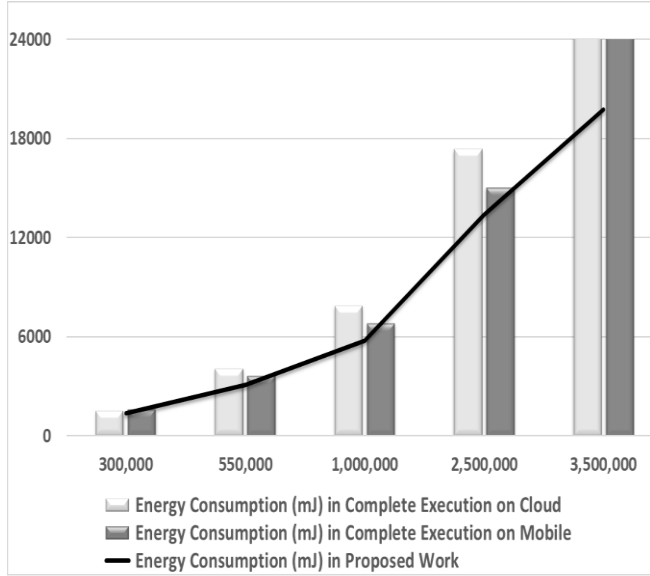

Figure 2. Size vs Energy Graph for Merge Sort Application

In Figure 3, an energy consumption is 16.1% to 20.3% when proposed framework is compared with complete execution on mobile device environment and 10.3 % to 26.7% in the case of comparison of complete execution on cloud and proposed work (Partial offloaded execution on cloud) for different data size.

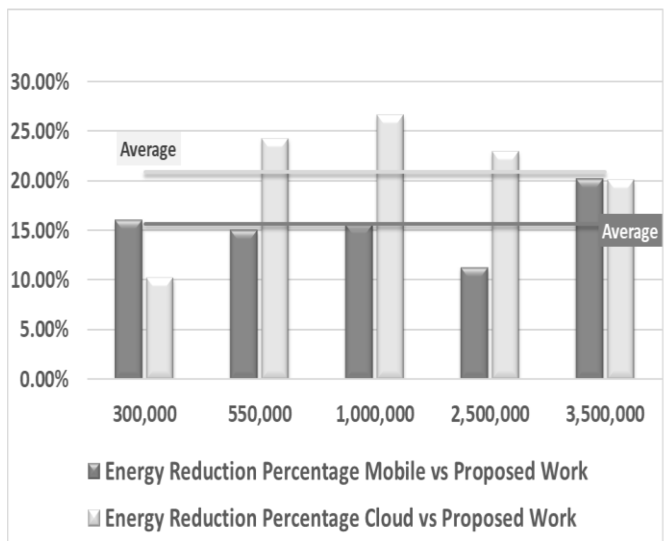

Figure 3. Energy Reduction Percentage Graph for Merge Sort

The next experiment has performed for the quick sort as it is one of the most efficient sequential sorting algorithm and good candidate for partitioning. It is the application of average time complexity in its worst case. As Figure 4 shows, energy consumption also increases with the computation size. It is also clear that the energy consumption is high for large data sets on mobile and lowest for proposed framework.

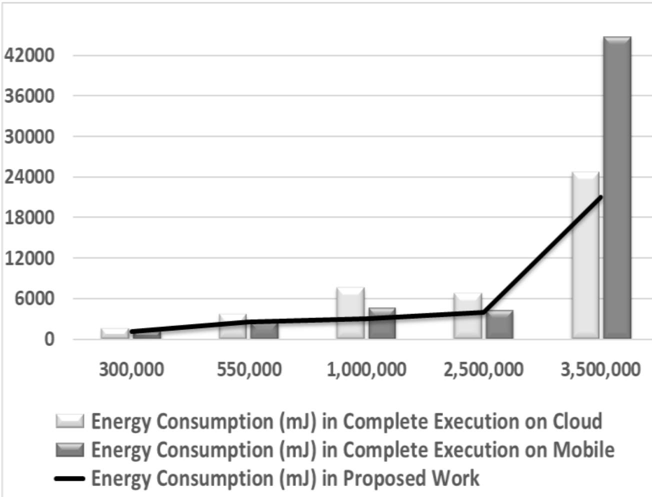

Figure 4. Size vs Energy Graph for Quick Sort Application

Figure 5 shows the comparison of energy advantage. It is clear that there is an advantage at around 1% to 53% in the case of comparison of proposed work with mobile execution and around 14% to 59% when proposed framework is compared with complete execution of computation on cloud.

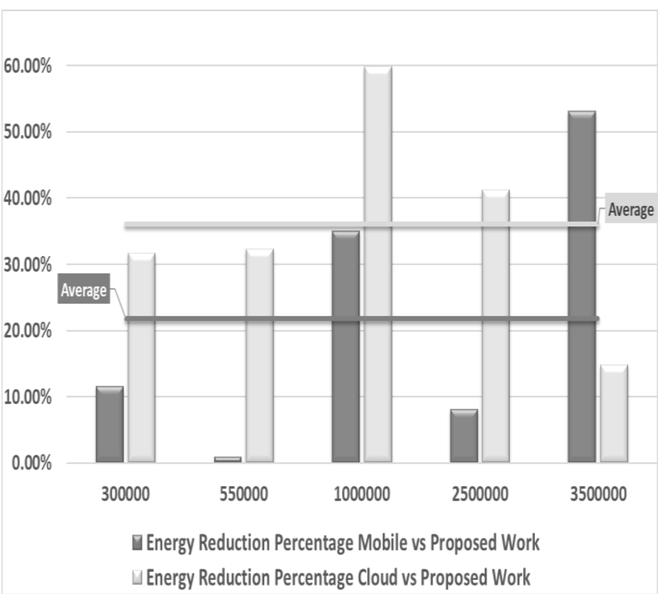

Figure 5. Energy Reduction Percentage Graph for Quick Sort

It has been experimentally verified that addition of partitioning parameter and simultaneous execution of computation task leads to a significant reduction in energy consumption and this is an efficient way of task offloading as compared to conventional method of offloading. The proposed work shows the proper offloading of computational components from mobile to cloud. Hence, this approach is useful to achieve the energy optimization for mobile device as well as better performance.

V. Conclusion and Future Scope

Traditional research works have focused to speed up the processing and conserve the energy for mobile devices. In the paper, the author has proposed a novel approach to offload the intensive tasks by increasing the performance and energy efficiency. The approach focused on the process of benchmarking of the intensive task before execution. According to the estimated value for energy, the decision has taken for offloading. The proposed work also used the concept of partitioning to divide the tasks across mobile and cloud to achieve the efficiency. To maintain the synchronization, parallel read and transmit operations are performed. The experimental results prove that the proposed work provides better results for performance and energy efficiency. The experimental work has carried out by the execution of computational application such as sorting (merge sort and quick sort). The achieved efficiency for smaller data size is not large enough but as the data size increases, the efficiency also increased because actual overhead of sending the files of small size on network is high. It is balanced against the efficiency achieved in offloading of large data. The achieved efficiency is around 20% for the merge sort application and 53% for the quick sort. Thus, the combination of task partitioning and energy parameter leads to an augmented energy efficiency and performance. However, more research work is possible for enhancements. For large computations, the decision of dynamic offloading can be on periodic basis. Computational offloading can be explored for peer networks and edge clouds.

References

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

[2] A. Gani, E. Ahmed, R. Buyya, Saeid Abolfazli, Z. Sanaei, "*Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges*", IEEE Communications Surveys & Tutorials, Vol. **16**, Issue. **1**, **2013**.

[3] A.P. Miettinen, J.K. Nurminen, "*Energy Efficiency of Mobile Clients in Cloud Computing*", In the Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, **USA**, **2010**.

[4] B.S. Lee, E. Leonardi, G. Goh, M. Kirchberg, V. March, Y. Gu, "*μcloud: Towards a New Paradigm of Rich Mobile Applications*", Procedia Computer Science, Vol. **5**, **2011**, pp. **618–624**.

[5] Cuervo, Eduardo, "*MAUI: Making Smartphones Last Longer with Code Offload*", In the Proceedings of the 8th

international conference on Mobile systems, applications, and services, ACM, **2010**.

[6] D. Tayade, "*Mobile Cloud Computing: Issues, Security, Advantages, Trends*", International Journal of Computer Science and Information Technology, Vol. **5**, pp. **6635-6639**, **2014**.

[7] Elgendy, A. Ibrahim, M.E. kawkagy, A. Keshk, "*Improving the Performance of Mobile Applications using Cloud Computing*", 9th International Conference on Informatics and Systems, IEEE, **2014**.

[8] G. Badshah, J.M. Zain, M.F. Zolkipli, M. Ali, *"Mobile Cloud Computing & Mobile Battery Augmentation Techniques: A Survey*", IEEE, **2014**.

[9] G.H. Forman, J. Zahorjan, "*The Challenges of Mobile Computing*", IEEE Xplore Digital Library, **1994**, pp. **38-47**.

[10] Kumar, Karthik, Y.H. Lu, "*Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?*", Computer **43.4**, **2010**, pp. **51-56**.

[11] Liu, Leslie, R. Moulic, D. Shea, "*Cloud Service Portal for Mobile Device Management*", e-Business Engineering (ICEBE), IEEE 7th International Conference on, IEEE, **2010**.

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

[13] Lordan, Francesc, R.M. Badia, "*Compss-mobile: Parallel Programming for Mobile Cloud Computing*", Journal of Grid Computing **15.3**, **2017**, pp. **357-378**.

[14] M. Ali, G. Badshah, "*Mobile Cloud Computing & Mobile's Battery Efficiency Approaches: A Review*", Journal of Theoretical and Applied Information Technology **79.1**, **2015**, pp. **153-175**.

[15] M. Shiraz, A. Gani, S. Khan, "*Energy Efficient Computational Offloading Framework for Mobile Cloud Computing*", Journal of Grid Computing **13.1**, **2015**, pp. **1-18**.

[16] N. Arya, S. Choudhary, T. Sunil, "*An Energy Efficient Task Offloading for Mobile Cloud Environment*", International Journal of Computer Science and Network, Vol. **8**, Issue. **3**, **2019**, pp. **306-310**.

[17] Paulson, L. Dailey, "*Low-power chips for high-powered handhelds*", Computer 1, **2003**, pp. **21-23**.

[18] Rahimi, M. Reza, N. Venkatasubramanian, A.V. Vasilakos, "*MuSIC: Mobility-Aware Optimal Service Allocation in Mobile Cloud Computing*", IEEE Sixth International Conference on Cloud Computing, **2013**.

[19] Rudenko, Alexey, "*Saving Portable Computer Battery Power through Remote Process Execution*", ACM SIGMOBILE Mobile Computing and Communications Review 2.1, **1998**, pp. **19-26**.

[20] S. Kathuria, "*A Survey on Security Provided by Multi-Clouds in Cloud Computing*", International Journal of Scientific Research in Network Security and Communication, Vol. **6**, Issue. **1**, pp. **23-27**, **2018**.

[21] S.M. Saad, S.C. Nandedkar, "*Energy Efficient Mobile Cloud Computing*", International Journal of Computer Science and Information Technologies, Vol. **5(6)**, pp. **7837-7840**, **2014**.

[22] Saraf, D.H. Gawali, "*IoT Based Smart Irrigation Monitoring and Controlling System*", Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2nd IEEE International Conference, **2017**.

[23] Smailagic, Asim, M. Ettus, "*System Design and Power Optimization for Mobile Computers*", VLSI, Proceedings of IEEE, Computer Society Annual Symposium on IEEE, **2002**.

[24] V.S. Varnika, "*Cloud Computing Advantages and Challenges for Developing Nations*", International Journal of Scientific Research in Computer Sciences and Engineering, Vol. **6**, Issue. **3**, pp. **51-55**, **2018**.

[25] Wu, Huijun, D. 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**.

[26] Y.G. Patil, P.S. Deshmukh, "*A Review: Mobile Cloud Computing: Its Challenges and Security*", International Journal of Scientific Research in Network Security and Communication, Vol. **6**, Issue. **1**, pp. **11-13**, **2018**.

Authors Profile

Ms. Nancy Arya pursed Bachelor of Science from University of Rajasthan in year 2007, Master of Computer Applications from Rajasthan Technical University in year 2010 and Master in Technology in Computer Science Enginering from Jagannath University in 2013 . She is currently pursuing Ph.D. in Computer Science & Engineering. She has presented many papers in national and international conferences. She has published around 15 research papers in reputed national and international journals including Scopus indexed journal and conferences including IEEE, Springer and Elsevier and it's also available online. Her main research work focuses on Cloud Computing, Mobile Cloud Computing, Algorithms, Network Security, Cloud Security and Privacy. She has 5 years of Teaching Experience, 1.5 years of Industry Experience and 3 years of Research Experience.

Dr. Sunita Choudhary pursed Bachelor of Science, Master of Computer Application and Ph.D. from Banasthali Vidyapith. She has worked as Assistant and Associate Professor in Banasthali Vidyapith from 2003 to 2016. Currently she is working as Guest Faculty in Rajasthan University. She has 16 year of Teaching Experience and 7 years of Research Guide Experience.

Dr. S Taruna is an Associate Professor in Department of Computer Science in JK Lakshmipat University. Her research areas are Adhoc and Sensor Network, Cloud Network and Data Mining. She has presented many papers in national and international conferenc around 25 papers in various reputed journals. She is also reviewer and committee member of various international journals and conferences. She has 20 years of teaching, research and administrative experience.