Mobile to Cloud Offloading Techniques

Literature Survey

Siddhi Vilas Sawant, Dept. of Electrical and Computer Engineering Colorado State University Fort Collins, United States siddhi.sawant@colostate.edu

***Abstract*--Mobiles have insufficient resources, like data storage, battery, bandwidth which leads to latency in runtime of applications. The problems could be overcome to a certain extend if mobile to cloud offloading techniques are adopted. These techniques allow the users to offload their data or operations on the cloud to store or the run the application without using the battery and data from the mobile devices. There are couple of clouds available in the sky. Thus, it becomes quintessential to select an ideal cloud for fast and reliable execution of an application. The selection procedure for offloading depends upon various parameters like availability, security, cost and network connections etc. In this literature survey, few of offloading and security techniques are been discussed.**

keywords: Mobile Cloud Computing, offloading techniques, tradeoffs, security and privacy

I.INTRODUCTION

There has been phenomenal development in research of hardware and software architecture, designing and popularity of mobiles. However, the upgrade in the hardware comes with consumption of battery from mobile phones. [1] Features like face recognition and video require high computation power for execution. [2] Thus, consuming a lot of battery and resources. This is not acceptable, to the users as they want excellent performance with minimum usage of battery and shorter execution time of the processes. [2] However, execution time, battery and power consumption on mobile applications can be minimized by using Mobile-to-cloud offloading techniques. Cloud has abundant computation services that can be used for the processing of the applications. Thus, applications using excessive resources can be migrated on third party server to avoid the hindrance in the efficiency and heating of mobile in the process. This is done by migrating applications or tasks in the application depending on different criteria on a suitable cloud.

Cloud offloading basically means, to transfer those processes to a third part which require a lot of computational power and consuming battery from the devices while running. As mobile devices have limited amount of battery cloud offloading technique can be used. There has been advance research in the field of cloud computing that has led to growth of couple of clouds. [4] Hence, selection of an ideal cloud from couple of clouds to offload is necessary to improve the efficiency of mobile applications. Multiple criteria's need to be taken into consideration before selection. Factors like availability, security, confidentiality, speed and capacity play vital role in choosing a cloud for offloading for a given program. These factors are calculated using various techniques to ensure user gets the results in stipulated amount of time with optimum usage battery.

The paper is divided as follows II. Offloading methods. Further the methods are divided into three sub-categories
II a. Offloading techniques to minimize the battery and latency.
II b. Security techniques to ensure the security of offloaded data. II c. Scheduling techniques to maximize the profit margin mobile service providers (MSP) and to offload a task on right Virtual Machine. III. Conclusion

II.OFFLOADING METHODS

To carry out the offloaded task with minimum delay execution time it is quintessential to select an optimal cloud among a certain class of clouds. When only one factor is considered in selecting a cloud it becomes an easy task. The traditional algorithms of server selection on cloud are limited only on one criterion. But, couple of criteria need to be considered when making final decision of offloading data onto a cloud.

The optimal cloud is selected using the following figure.

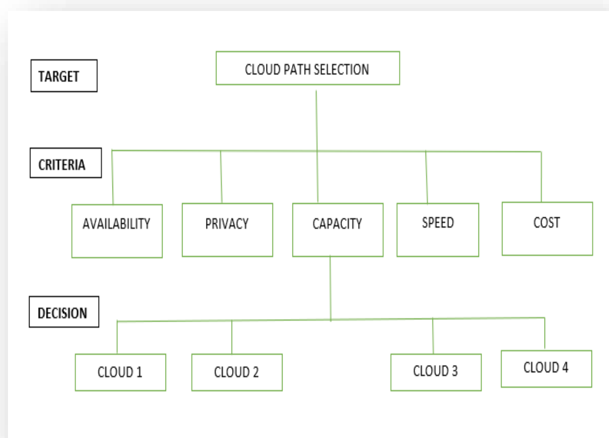


Fig. 1. Hierarchical Structure of Cloud-Path Selection [1]

In figure 1. Three hierarchies are listed. What one wants to do and what the object is mentioned in the first level i.e. target hierarchy. The second level is criteria hierarchy which mentions the criteria to be mentioned for cloud-path selection. The third level is called the decision hierarchy meaning here based on the analysis in criteria hierarchy one of the cloud is chosen.

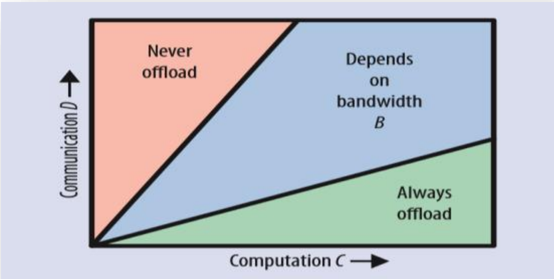


Fig. 2. Offloading is beneficial if D/C is low [6]

II.A. Offloading techniques

Here, [8] cloud selection is done using fuzzy set theory. As it is difficult, to make an exact decision due to lack of simplicity and certainty of the environment needed by Analytic Hierarchy Process [7-8].

Zadeh [9] introduced fuzzy set theory to determine uncertainty in making decision problems. It has a membership function µ whose value ranges from 0 to 1 [9]. Like AHP is divided into six steps FAHP is too divided into six steps as follows.

1. Construct the ranked structure based on different criteria and form pairwise comparison matrix and express the comparison matrix. [4]
2. Check consistency of the comparison matrix as follows
   The consistency of matrix is checked as follows

   sij=0.5, sij+sji=1, $\frac{1}{sij}-1=(\frac{1}{sik}-1)\ x\ (\frac{1}{ski}-1)$
3. Calculate positive fuzzy matrix to transform scores of pairwise comparison into fuzzy variables having values ranging from 0 to 1. [4]
4. Calculate fuzzy weights of decision elements. [4]
5. Take the geometric mean by combining the decision of all decision makers.
6. Calculate the consistency index(CI) to obtain ranking of the criteria. [4]
   CR=$\frac{CI}{RI}$
   CR= Consistency Ratio, RI= Random Consistency Index

Evaluation matrix for cloud path [4] selection is obtained and a cloud with highest performance must be used to offload the data. In this case cloud 3 is a suitable cloud to offload the corresponding application.

Evaluation matrix for the alternative clouds for cloud selection. [4]

Cloud	Avail	Privacy	Capacity	Speed	Cost	Result
Cloud1	0	0	0.01528	0.0696	0.0134	0.0983
Cloud2	0.037	0.012	0.0306	0.0696	0.0067	0.1562
Cloud3	0.2597	0	0	0.2436	0.0134	0.5167
Cloud4	0.074	0.0122	0.1071	0.0348	0	0.2283

While it will always not be, efficient to due offloading on cloud as it may consume more battery to upload a task on cloud due to network issues. To ensure that a user selects best cloud and minimum battery is consumed in the process MAUI profiler is being used. MAUI profiles the code contained in the functions that are used by applications and other component known as MAUI profiler profiles the corresponding data requirements of the functions. [5] Along with that the profiler checks the network connections and feeds this information to the solver to determine where to perform the task either locally or on the remote location. [5] Then the Proxy serializes function calls so that they can run at appropriate locations. [5]

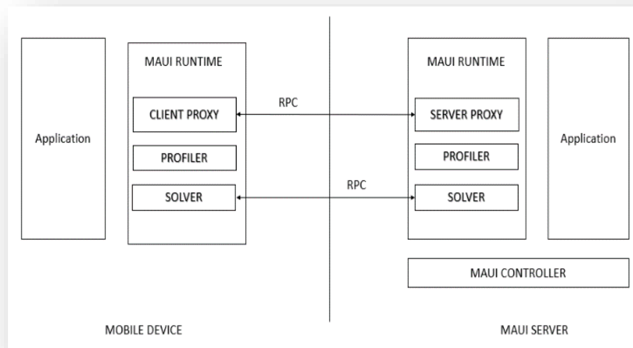


Fig.3 MAUI Architecture [5]

MAUI is lightweight and can be easily handled by a developer and it remains seamless to the user. This is done by using a developer tag a function as 'remote-able'. This ensures that the functions which access local peripherals cannot run remotely. [4] The solver in the MAUI decides whether to upload the task or not. The second advantage of the MAUI is that it adapts to the varying environment. The solver depending upon the factors like network connectivity, battery etc. decides whether to offload the task or not. If the conditions are not in the favor then solver tries to do a tradeoff between battery, speed and cost.

In this method [10] two processes are used – AHP and fuzzy TOPSIS. For multi criteria decision problem, AHP is used to determine the relative importance of a set of alternatives.

The results of the pairwise comparison on N criteria can be expressed in an evaluation matrix as

$$A=\begin{matrix} a11 & a12\ldots\ldots & a1n \\ a21 & a22\ldots\ldots & a2n \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ an1 & an2\ldots\ldots & ann \end{matrix}$$

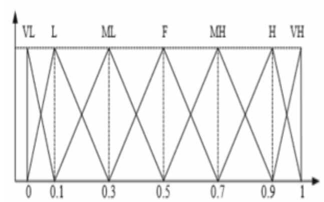
Here aii=1 and aji=1/aij

Fuzzy TOPSIS is used to obtain the final ranking of the clouds. The process steps can be outlined as follows.
Establish a decision matrix for the ranking:
The structure of the matrix can be expressed as follows.

$$X=\begin{matrix} x11 & x12\ldots\ldots & x1n \\ x21 & x22\ldots\ldots & x2n \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ xm1 & xm2\ldots\ldots & xmn \end{matrix}$$

n= No. of criteria; m= No. of alt.

1. Calculated the weighted normalized decision matrix
   Vij= xij X wj i= 1,2,……………….m ; j=1,2,……………………………….n
   where wj represents the weight of the jth criterion, which is obtained from the AHP method.



Linguistic values	Fuzzy ranges
Very Low	(0,0,0.1)
Low	(0,0.1,0.3)
Medium	(0.1,0.3,0.5)
Fair	(0.3,0.5,0.7)
High	(0.5,0.7,0.9)
Very High	(0.7,0.9,1.1)

Fig.3. Membership functions of linguistic values [10]

2. Determine the positive-ideal (A+) and negative-ideal solutions (A−), respectively:
   A+ = {v1+, v2+, ……………….vn+} = {(max vij l i € I), (min vij l i € I')}
   A- = {v1-, v2-, ……………….vn-} = {(min vij l i € I), (max vij l i € I')}
3. Calculate the distance of each alternative from A+ and A− using the Euclidean distance:
   Di + = $\sum_{j=1}^{N} d(vij, vj+)$ i=1,2,…………m
   Di - = $\sum_{j=1}^{N} d(vij, vj-)$ i=1,2,…………m
4. Calculate the relative closeness of the ideal solution
   Ci*= $\frac{Di^-}{Di^+ + Di^-}$

Rank the alternatives according to Ci* in descending order.
By following the above steps, a best cloud meeting all the criteria will be selected. Thus, processing the offloaded data within the given set of parameters. [10]

For implementing optimization techniques in mobile devices, changes must be made in application software architecture and complex OS which becomes very challenging. Apart from that to ensure that minimum energy is consumed in offloading it is necessary to make correct calculations of response time and offloaded energy. These two factors are greatly dependent on the wireless network quality and wireless interface in use which varies a lot. Like whether to go for 3G, consumes less power but slower data transfer rates or 4G, higher data rates but consumes more power. The trade-off between them vary across the location. Hence, middleware use is becoming popular for conducting optimization techniques and energy-efficient methods without changing the architecture or OS of devices. As it provides flexibility and standardization.

In [11] a middleware framework is proposed which provides energy-efficiency and performance robustness. By utilizing various resources like -type and status of available wireless networks, the capabilities of cloud servers and the application communication/computation intensiveness. It analyzes data for the network type, app type, cloud capabilities and network conditions by using an unsupervised Q-learning machine learning technique to select an optimal cloud and network type and to decide what and when to offload. It is proved that there has been 30% improvement in battery life and 25% better response time as compared to the state-of-the-art fuzzy logic based offloading approach. [11]

In ThinkAir technique, method level offloading is adopted. Basically, methods migrated are from the smartphone to ThinkAir server [12]. Thus, processes at mobile side are reduced. At the server side, to handle computation offloading smartphone virtualization is done. The smartphone acts like a thin client displaying the received results and saving the battery in process [12]. Through task distribution support over virtual machine images scalability and elasticity is obtained on the server side in ThinkAir. Reduction in execution time and energy consumption of applications is attained through parallel processing in ThinkAir.

In [13] delayed offloading queuing model is used. Instead of transmitting the offloaded data immediately the data to be offloaded is delayed when the Wi-Fi networks are unavailable. The delay window depends on two factors i.e. first the expiry of deadline and second the availability of Wi-Fi network. If one the mentioned factor holds true, then data is no more delayed.

To react to the continuous changes in the variables and foster an interference model based on the comparison of multiple dependent variables a system based on fuzzy logic is adopted [15]. A fuzzy logic system consists of fuzzifier- to transform crisp sets to fuzzy sets. Rules are created based on basic fuzzy operations, targeting the variable to control, reasoning engine to make inference based on set of rules and defuzzifer to map the fuzzy sets output to crisp sets.

In this method each input parameter like mobile, cloud etc. are considered as crisp sets and then transformed to respective fuzzy sets. To identify the crisp set a linguistic value is created and then translated into a possible finite number of Fuzzy sets. A membership function is associated with FS whose values are extracted from CS and arranged in intervals. [14]. As in fuzzy logic system infers a decision based on degree of truth to a specific criterion. By this method of inference of FLS it can be used to quantify how the execution of components can be segregated between local and remote processing. The center of mass formula is applied to compute grade of truth to the decision.

Most of the times the important factor cost is been overlooked in mobile to cloud computing. To keep the data center operation cost low, it is necessary to use the server to its optimum level. Hence CMcloud method is been adopted. CMcloud makes optimum use of the server to cut down the cost as much as possible.

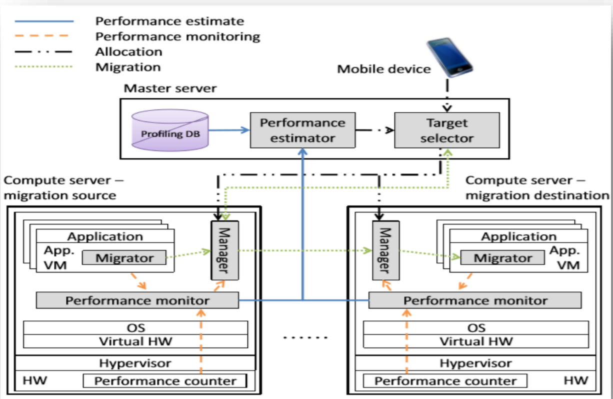


Fig.4. CMcloud Architecture [15]

First when the offloading request is been made the performance estimator collects the estimated performance from ever server. Based on this information target selectors finds out the most suitable server which has the adequate resources for processing the offloaded data. Using a random server will not fetch the desired results of optimum utilization of the resources. Hence target selector is been used. Its job is to find a server whose resources are just adequate to satisfy the needs of the offloaded data without compensating on the results. If the resources required for computation of the offloaded data are more than the available resources in the servers then new server is added. Cost is cut done by offloading as many applications on the server as possible. 84 % improvement in the datacenter throughput is observed when CMcloud is used. 83% reduction in the number of service failures in comparison with static light load scheme while stilling improving the throughput by 31%. [15]. Reduction in cost, improvement in the datacenter throughput is possible because of the effective utilization of the server.

In [16] Clone Cloud system is proposed. This system will partially offload applications running on mobile devices onto the clouds specifically on the virtual clones present on cloud surface automatically. It is proved in [16] that this method helps to reduce the runtime of task and minimize the battery usage.

Computation offloading means to run a code of mobile application fully or partly on the cloud server. It can be done in two ways

Coarse Grained Offloading- Here, the entire application code is offloaded on the server i.e. cloud. In this method the code is not revised thus no burden on the programming side but as the entire application is offloaded on the cloud server there is a transmission overhead. This overhead will in turn increases the runtime of applications. [17]

Fine grained offloading method –The apart of the application which consumes a lot of power is offloaded on the cloud. In this method, those parts are offloaded on the cloud which are energy hungry and not the entire application or task. [17] Thus, saving the application and device from transmission overhead but increasing the programming complexities. Hence depending upon the needs of the user one of these methods can be used.

In [18] a framework is proposed known as CDroid (cloud android). This framework combines the mobile device operating system to its cloud server. The data traffic of the device is controlled effectively by the cloud. Along with that, for future use to access the same data traffic the cloud simultaneously uploads code and data into its cache. This method reduces the energy and data traffic in mobiles.

As battery of mobile devices is retained by offloading computational intensive task on the cloud it is necessary that energy is saved while computing this task on the cloud. In [19] GEM Cloud method is proposed. In this method, the energy is saved by using scattered mobile devices to process task in parallel [19]. Server client protocol refer fig. 4 is used.

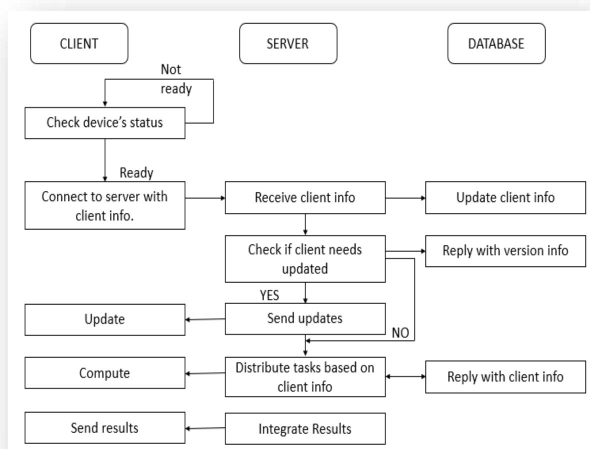


Fig. 6 Flowchart of server-client protocol [19]

This protocol divides the user input task on the client to be processed. If the size of computational task is small, then the server will reduce the load on each client thus decreasing the turnaround time for the task.

II.B. SECURITY METHODS

Along with different techniques to offload the data with minimum latency and battery the other factor that needs to be consider is security of offloaded data. To prevent unauthorized access to offloaded data different techniques can be implemented. In this paper few of the security techniques are discussed

A. Non-Biometric Authentication

In this [20] method distribution authorization token is generated for a single access to resources. It consists of two parts- Identity management system(IMS) generates a token and sends it to user and cloud. Then cloud generates one more token and sends it to the user. The token send by IMS and to user to/by cloud are stored in cloud database. When the user logins in the server by passing the two tokens obtained from IMS and cloud the cloud compare them with these tokens within the database. To authenticate the user or not.

B. Biometric Authentication

In this [21] method, biometric input is considered as fingerprint image which is captured by the mobile phone camera used as biometric sensor. The captured image is stored in the cloud database whenever the user wants to access or offload data it must go through validation process. It is done by scanning the fingerprint. If the user's fingerprint matches with the fingerprint in the database, access is proved otherwise not.

To improve the key generation process in traditional fingerprint image processing which is done by extracting minutiae points from the image. In [22] a new distance based key generation algorithm is proposed where Centroid C for the given fingerprint image is identified. By using this algorithm key can be generated. It simplifies the process of generating the crypto keys as compared to the traditional algorithm. This improved version of algorithm can be used for authentication of user to access services on cloud.

II.C SCHEDULING METHODS

Offloading the task on cloud using these techniques will achieve the goal of uploading the task on cloud but directing the uploaded data to the desired Virtual Machine in stipulated time to achieve the goals of minimum latency, power and is also important. Some of these goals can be achieved if proper scheduling algorithms are used.

Here [23] an improved version of FCM that is IGAFCM algorithm is proposed. This algorithm consists of three parts – 1) Resource clustering- To reduce the screening data volume, large number of resources are decomposed. 2) Resource Scheduling- By using algorithm of fast searching the Top- N results are chosen. 3) Score Matching- Based on the demands and resource performance the best candidate from the queue is selected by the dispatcher and return to the user. [23] As compared to the traditional MIN- MIN algorithm the proposed algorithm increases the efficiency in the scheduling process as it improves the demand hit factor.

The IGAFCM has a good convergence required for re-clustering of resources that in the process has more speed in comparison with the traditional FCM algorithm. The reason it works better than the FCM because genetic algorithm finds the approximate center for global search instead of finding the local optimal solutions.

Other way to improve the capabilities of mobile devices to handle traffic done by mobile service provider (MSP) can be improved by integrating the cloud radio access network (C-RAN) with the mobile edge cloud computing (MEC) technology [24]. But the lot of power is consumed, and profits obtained are less. In order to solve this problem in [24] joint scheduling of network resources in C-RAN and computation resources in MEC is done. This objective is attained by – implementing the extended version of Lyapunov technique and by formulating resource scheduling issue as a stochastic problem. As in real world dynamic situation- the job requests won't be of fixed lengths and finished within the stipulated time frame as assumed by the Lyapunov technique. Hence an extended version of Lyapunov technique is implemented and new algorithm- VariedLen is proposed. For job requests varying in lengths this algorithm makes online decisions in consecutive time. The algorithm maintains low congestion and strong system stability and can give (1/v) diminishing gap for the MSP for time average profit.

One of the important features of cloud computing are virtual machines which are built as per the client requests. Jobs are processed and executed randomly in virtualization. The disadvantage is that the task which have long run time end up using the VM for longer time for processing and in turn the small tasks have to wait for a long time. A solution is presented in [25] to tackle this problem. If the current task has more MIPS as compared to the VM MIPS then a new VM is create on the same physical machine.

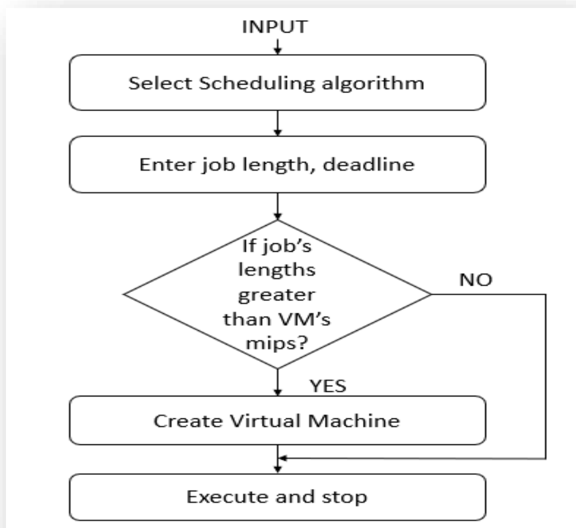


Fig. 7 Proposed System Overview [25]

In fig.7 First input is taken, and user selects the algorithm. Depending upon the type of algorithm the inputs varies - Shortest job first then Job have shortest execution time is executed first. In Earliest Deadline First task having highest priority is executed. Length as well as priority is considered in Credit Based scheduling algorithm. Second, the length of job is entered but for EDF deadline Is entered. Third, here the system has three virtual machines with 200, 500, 1000 MIPS. If the entered job has high MIPS then a new VM is created. Fourth, the task is executed on desired VM. Thus, all task gets a calculated amount of time to process and are not starve based on their data input or priority.

Most of the algorithms fragment a single task and disturb it to multiple devices on cloud to minimize the runtime and latency. But when the load on the cloud is high these techniques do not give the desired results. In order to address this issue in [26] a new approach is proposed- Round-Wise Pro-active Scheduling. On the cloud platform, ROSAC divides and disturbs the task based on their computing power and complexity thus decreasing the average completion time and increasing the resource utilization of all the offloaded task [26] First, from the received task Ti the cloud retrieves the asymptotic complexity Ci. This information is used and precomputed with its task profile and determines T'i ⊂ Ti. Here the estimated runtime of all the Ti's are same and only they are considered for computation [26]. Further the Ti are divided into subtasks and distributed depending upon the computing power of the devices in their allocated resource pool. The same procedure is repeated for the remaining portion of task. Thus, addressing multiple tasks.

III.CONCLUSION

It can be concluded that mobile to cloud offloading has a great scope increasing the battery life of the mobile devices by outsourcing the task on the third-party server i.e. cloud. These servers not only consume less power but also process the task in less time and procure the correct results. As these servers have abundant resources to process the task, the execution time required to run the same task on the mobile increases the run time and consumes a lot of battery. Along with that the advantages of offloading are- lower bandwidth requirements, data charges and usage. But there few limitations to this technique. First, the three factors that act as a barrier to adoption of MCC are privacy, security and trust. Second, environmental uncertainty- due to changes in environment access the resources of won't be always feasible and reliable. Lastly, adoption of this technique in day to day by people. Hence, techniques that not all provide reliable connection and minimum battery usage but also which ensures the user that the uploaded data is in safe hand by using a high end encrypting algorithm and easy to use and adopt should be implemented.

IV.REFERENCES

1. B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama and R. Buyya, "mCloud: A Context-Aware Offloading Framework for Heterogeneous Mobile Cloud," in IEEE Transactions on Services Computing, vol. 10, no. 5, pp. 797-810, 1 Sept.-Oct. 2017.
2. D. Shibin and G. J. W. Kathrine, "A comprehensive overview on secure offloading in mobile cloud computing," 2017 4th International Conference on Electronics and Communication Systems (ICECS), Coimbatore, 2017, pp. 121-124. doi:10.1109/ECS.2017.8067851
3. Z. Zhang and S. Li, "A Survey of Computational Offloading in Mobile Cloud Computing," 2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), Oxford, 2016, pp. 81-82. doi:10.1109/MobileCloud.2016.15
4. C. Singla and S. Kaushal, "Cloud path selection using Fuzzy Analytic Hierarchy Process for offloading in Mobile Cloud Computing," 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), Chandigarh, 2015, pp. 1-5. doi:10.1109/RAECS.2015.7453370
5. R. Chandra, S. Hodges, A. Badam and J. Huang, "Offloading to Improve the Battery Life of Mobile Devices," in IEEE Pervasive Computing, vol. 15, no. 4, pp. 5-9, Oct.-Dec. 2016. doi:10.1109/MPRV.2016.77
6. K.Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" Computer, vol. 43, 2010.
7. C. S. Yu, "A GP-AHP method for solving group decisionmaking fuzzy AHP problems", Computers and Operations Research, vol. 29, (2002), pp. 1969–2001.
8. Saaty, T. L. (1980). The analytic hierarchy process. New York: McGraw-Hill.
9. L. A. Zadeh, "Fuzzy sets", Information and Control, vol. 8, (1965), pp. . 338–35
10. H. Wu, Q. Wang and K. Wolter, "Methods of cloud-path selection for offloading in mobile cloud computing systems," *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, Taipei, 2012, pp. 443-448. doi: 10.1109/CloudCom.2012.6427587
11. A. Khune, S. Pasricha, "Mobile Network-Aware Middleware Framework for Energy Efficient Cloud Offloading of Smartphone Applications", to appear, IEEE Consumer Electronics, 2017.
12. S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobilecodeoffloading,"inINFOCOM,2012ProceedingsIEEE. IEEE, 2012, pp. 945–953
13. F. Mehmeti and T. Spyropoulos, "Is it worth to be patient? analysis and optimization of delayed mobile data offloading," in INFOCOM 2014, IEEE International Conference on Computer Communications, 27 April-2 May, 2014, Toronto, Canada. IEEE, 2014.

14. D. Chae *et al*., "CMcloud: Cloud Platform for Cost-Effective Offloading of Mobile Applications," *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Chicago, IL, 2014, pp. 434-444. doi: 10.1109/CCGrid.2014.75
15. B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: elastic execution between mobile device and cloud,” in Proceedings of the sixth conference on Computer systems, ser. EuroSys ’11. New York, NY, USA: ACM, 2011, pp. 301–314.
16. “Adaptive Code Offloading for Mobile Cloud Applications: Exploiting Fuzzy Sets and Evidence-based Learning “,Huber Flore and Satish Narayana Srirama
17. X.Ma,Y.Cui and I.Stojmenovic “Energy optimizations for mobile terminals via computation offloading,”in Parallel Distributed and Grid Computing (PDGC),pp. 236-241,Solan,6-8 Dec. 2012.
18. MV. Barbera, S.Kosta, A.Mei, VC. Perta,and J.Stefa, “Mobile Offloading in the Wild: Findings and Lessons Learned Through a Real – life Experiment with a New Cloud-aware System,”in IEEE INFOCOM, pp 2355-2363, Toronto,ON, 27April- 2 May 2014.
19. He Ba Wendi Heinzelman,C.Antoine Janssen and J.Shi “Mobile Computing-A green computing resource”Wireless Communications and Networking Conference (WCNC), pp 4451-4456, 7-10 April 2013.
20. Azeem Ahmad, Muhammad Mustafa Hassan, Abdul Aziz,” A Multi-Token Authorization Strategy for Secure Mobile Cloud Computing” 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering
21. IehabALRassan, HananAlShaher,” Securing Mobile Cloud Using Finger Print Authentication” International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.6, November 2013
22. R Ranjan, Sanjay Kumar Singh,” Improved and Innovative Key Generation Algorithms for Biometric Cryptosystems” 2013 3rd IEEE International Advance Computing Conference (IACC)
23. V. Meena, M. HariPrasath, V. Kalpana, K. ArunKumar and J. SenthilKumar, "Optimal resource reservation for offloaded tasks in mobile cloud computing," *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, 2017, pp. 677-682.
24. W. Hong-Qiang, L. Xiao-Yong, F. Bin-Xing and W. Yi-Ping, "Resource Scheduling Based on Improved FCM Algorithm for Mobile Cloud Computing," *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, Wuhan, 2016, pp. 128-132.
25. P. V. Kapse and R. C. Dharmik, "An effective approach of creation of virtual machine in cloud computing," *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, 2017, pp. 145-147. doi: 10.1109/I-SMAC.2017.8058326
26. S. Noor and R. Hasan, "ROSAC: A Round-Wise Fair Scheduling Approach for Mobile Clouds Based on Task Asymptotic Complexity," *2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, San Francisco, CA, 2017, pp. 60-65.