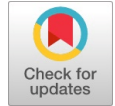


Secured and Dynamic Decentralized Computational Offloading Framework for Mobile Cloud Computing



Shanthi AL, Ramesh V

Abstract: Mobile Cloud Computing is represented as an advanced mobile computing technique by integrating with resource rich servers of various clouds and networks towards infinite mobility, computation, functionality and storage in favour to overcome the restrictions of mobile devices. Mobile Cloud Computing is a platform where storing and processing of data are performed away from mobile devices and processed in the cloud and bringing back results to mobile device to improve the capabilities of mobile devices. The main objective of this study is to minimize response time and to improve the battery performance of mobile devices. In this research, an intelligent secure and dynamic decentralized framework is proposed which will provide accurate decision for execution environment for the application either local or at cloud using SENN classifier and DCNN Model. Modified Fuzzy C Means Clustering algorithm is put forward to create alike clusters for the profiler parameters information collected by smonitor. Moreover, the proposed framework provides more security by encrypting the input image data while transferred to cloud server using blowfish technique can protect the application data from threats. The test results in simulation environment proved that SDCNN framework achieved significant performance by minimizing the consumption of energy and execution time by offloading computation intensive task to clouds.

Keywords: Mobile Clouds, Offloading, Decision Making, Decentralized resource allocation, Clustering, Energy Consumption.

I. INTRODUCTION

Nowadays, mobile devices are improved in terms of speed, sensors and acute screen enable to amble resource rich or computation intensive applications such as object tracking, face recognition, web surfing, speech and object recognition, video streaming, augmented reality and interactive games. So, mobile device computing has become more practicable than the traditional approach. But still mobile devices have few restrictions like limited battery life, processing power and in terms of storage becomes the main challenges for mobile computing technology. In order to evolve a solution for these limitations, led the researchers to the formation of Mobile Cloud Computing, amalgamation of mobile and cloud computing. In MCC, smart phones are connected to high computational power server to increase the capabilities of mobile devices.

In MCC, computation offloading is a method transferring and executing computation intensive applications from mobile devices to server in order to minimize processing time and to improve battery life. Recent research in Mobile Cloud Computing mainly focused on either full offload or partial offload of the applications from local devices to cloud without supporting decentralized resource allocation mechanism. Most of the existing frameworks are not consider RAM size in decision making process for execution environment. RAM is one of the main resources to execute application in mobile devices. Moreover, Security concern are not much focused before while migrating images to clouds. So, this work mainly focuses to propose a Secured, Dynamic and Decentralized model to overcome the constraints mentioned above and also aim to minimize response time and energy consumption. This research proposed a dynamic and novel framework that used decentralized resource allocation for offloading the execution of computation intensive tasks to clouds. **The main objective and contributions of this work are stated below:**

This work proposes a Secured, Dynamic and Decentralized novel framework that offloads compute intensive tasks to clouds to reduce the overheads in transmission. Decentralized model is formulated for decentralized resource allocation which deallocates the allocated resources to achieve better response time.

Optimized model is put forward to predict the offloading decision using SENN classifier dynamically at runtime based on profiler parameters such as average response time, battery charge, bandwidth and signal strength.

Security process is added to the proposed framework to protect the image data while transferring to clouds using blowfish algorithm.

A. MCC Advantages Extended Battery life

Migrating and processing the resource intensive mobile applications to cloud improved the battery life of mobile devices.

Data Storage

By integrating mobile devices with cloud server, users can enable to store their data in cloud encrypted manner and possible to access their data anywhere and anytime

Increased Processing Power

Resource rich, computation intensive and background processes of a mobile devices are offloaded to cloud to improve the computing power of a mobile device.

Manuscript published on 30 September 2019.

*Correspondence Author(s)

Shanthi AL., Research Scholar, SCSVMV, Kanchipuram

Ramesh V., Asst Professor, SCSVMV, Kanchipuram

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license [http://creativecommons.org/licenses/by-nc-nd/4.0/](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Reliability

Executing applications on the server machine is more reliable than running it on the local machines as there are many copies of the processed data are stored in cloud.

Ease of Integration

Mobile users possible to access multiple resources and services into one in the form with the help of cloud resources. **The main objective of MCC are as follows:**

Improved Performance

The main aim of the framework is to improve the performance and resources of mobile devices by offloading intensive tasks to clouds

Dynamic Provisioning

Dynamic provisioning of the system allows mobile users to create their respective VMs with their required configurations that are dynamically spawned without reserving of resources in advance.

Enhancing Quality of Service

The QoS of a mobile application can be increased with improved QoS factors such as average response time, signal strength, bandwidth, available battery level and ram size. The execution time of an application could be minimized by offloading to clouds.

II. EXISTING FRAMEWORKS

Chan et al [1] proposed Clone Cloud framework to minimize battery consumption on the mobiles by moving computation intensive application to clouds. Partitioning the application can be done by adding program partitioning with profile program. Decision making module decides the execution environment during runtime either at mobile or cloud. If the decision at cloud means, all the current executing threads on mobile device stopped and transferred to clouds and result back to mobile devices. Cuervo et al., [2] proposed Mobile Assistance using Infrastructure [MAUI] is a programming model that enables offload an application code to a nearby cloud infrastructure. MAUI basically meant to target applications developed in .net framework. MAUI first methodology to count on the programmes to postulate the portioning program. And then process the program by migrating the whole VM to clouds. It partitions the methods that can be offloaded to cloud during runtime and sends those tasks to remote locations for processing and results sent back to mobile devices.

S. Kosta et.al[3] tries to address Think Air offloading framework that makes it easy for app developers to offload their application to clouds. This framework is based on smartphone virtualization aims at providing method level offloading. ThinkAir supports real time resource allocation for mobile users and also facilitates the provision for parallel processing of applications helps to reduce execution time and to increase processing power for an application. Kemp et al., [9] presented CUCKOO framework is the first practical MCO framework which targets android platform. It mainly focused on those mobile environments where there is a problem of frequent loss of network connections with remote machines. It also provides the concept of both early and late binding of the processed code. Cuckoo targets only Android platform which includes runtime system, resource manager, application positioned in mobile devices to know the registered available resources to accomplish development process.

III. SDCOF ARCHITECTURE

In SDCOF Architecture, the compute intensive tasks of the mobile applications are migrated to cloud server with the help of internet connections via 4G/5G or high speed WIFI connections. Cloud server executes the intensive parts and the results are back to mobile device and resume the process.

A. Architecture of the System

The framework of SDCOF computation offloading system is depicted below. Compute intensive methods of the mobile application should be denoted as @ Remote during development process of mobile application. Mobile devices hold application data and code along with different modules of offloading decision-making process. The computation intensive task is executed on the cloud server on the request of mobile users, along with the clone of mobile devices. Client handler in cloud handle and execute a request from mobile device and sends the result back to mobile.

The major components of the framework are described below:

A. Profiler

Smonitor collects the profiler information that helps the system to make accurate decision regarding offloading by collecting context information about device, application and network. Device profiler states the status of mobile like CPU, battery status and screen brightness. Network profiler monitors the information about network condition such as type, signal strength, latency and bandwidth. Application profiler collects the features about input size, no of instructions and expected execution time. Average response time, signal strength, bandwidth, available battery level and ram size data are used for decision making process collected by smonitor. Average response time is calculated by taking average of how much time taken to execute compute intensive task at mobile device. By sending dummy packets from mobile to server, current bandwidth information is collected by network profiler. Mobile device sends ICMP packet to the server to calculate latency. By examining proc/meminfo files of android, RAM availability is determined.

B. Clustering Devices

In order to minimize the execution time of execution handler choice or execution time, clustering process is proposed to collect and cluster all the mobile device requests. The parameters involved in creating clusters are mobility, distance, signal strength and battery power. Usually, the fuzzy c means clustering calculation involves the membership of all mobile device, denotes the level of the request holding a place with the cluster. When two of these request characteristics are similar, then the distance between the devices is lesser, FCM created clusters., otherwise it is considered to be non-identical devices. On FCM technique, distance is calculated using Euclidean distance creates supposition for the period of clustering process, as an outcome it may decrease cluster results. To reduce the above issue, Mahalanobis distance is used to calculate the distance between mobile device request for altering the FCM algorithm called Modified Fuzzy C means algorithm.

Mahalanobis distance consider covariance which forms an elliptical decision boundary that improves the clustering result. It also follows statistical approach which performs better on correlated data. MFCMC iteratively moves the cluster centers to the exact location inside mobile devices. No preset cluster is obtained for the membership grade of an exception are equal.

D. Lagrange Solver

Lagrange solver procure the basics for preparing the offloading decision module based on the predicted execution time of the task offloading in remote by deriving minima and maxima of the time and energy cost for remote execution. The parameters for offloading decision are not related linear. Time and cost function is determined by the network performance and not entirely depends on increment or reduction with server speed.

E. DCNN Model

To make decision for offloading, it is very necessary to find the estimated execution time of the task at mobile device. In the above approach, Deep Convolutional Neural Network model is utilized to estimate mobile execution time. It calculates the estimated time by adapting the dynamic behaviour involved in mobile devices.

F. SENN and Decision

This classifier module provides support to decide upon the location of execution either at local or remote. Depends upon the estimated execution time at remote and

local execution along with the gathered data from other modules, it predicts the execution environment.

G. Decentralized Mechanism

ABAT procedure with tabu search is utilized in the above system to offer decentralized mechanism. After the request from the mobile users for mobile cloud execution, the allocation of resources is performed. Using this approach, the decentralized mechanism is organized for the mobile resources which improves energy saving.

H. Security

Transmission of data or storage of data needs more secured mechanism to avoid misinterpretation of data from unauthorized access. Therefore, protection of data images is necessary when transmitting to clouds. Encrypting and decrypting technique plays a vital role in protecting image data. To propose more secured framework, Blowfish technique is included to prone image encryption and decryption process for image security while transferred to clouds. The original image is encrypted into non readable format called cipher text by generating keys. 64-bit block size and 448 bits and a key length of 32 bits to is used. Each s boxes generate 32-bit output by accepting 8-bit input.

I. Client Handler

Client handler at cloud receives and handle the offloading request from mobile devices. It processes the request and the results are sent back to device.

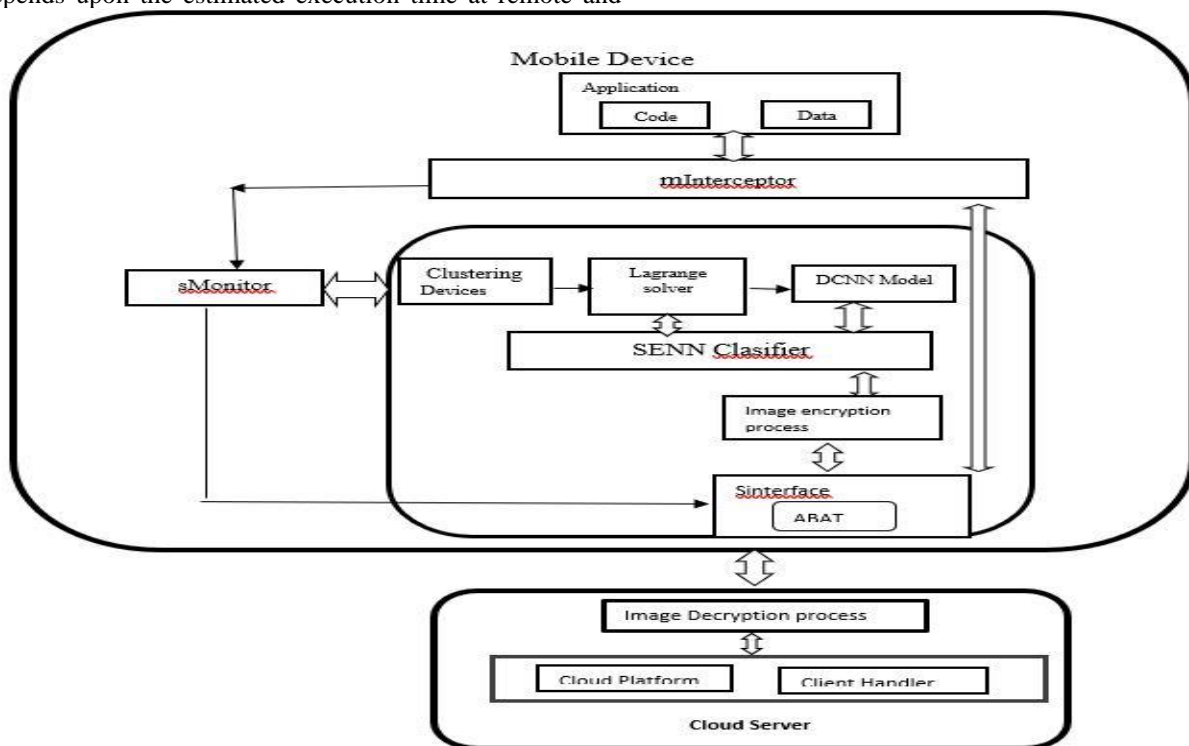


Figure 1: SDCOF Architecture

Algorithm 1: SDCOF framework Execution Flow

Input: input profiler parameters for each annotated method.

Output: results for determining the execution environment.

1. Read the method name which is denoted as @remote.
2. Check the status of the network like bandwidth of the network, signal strength and latency using network profiler.

3. If there is no connection or not enough bandwidth
4. Execute the task locally Else
5. For each method I do

6. Read the dynamic parameters through profiler module and input to SENN classifier to predict offloading decision for execution as an optimized model
7. If the output of the decision process module is yes, then encrypt the image using blowfish algorithm.
8. Send the methods to server for remote execution and send results back to mobile device.
9. Else execute the method locally.
10. Endif
11. End process.

IV. PERFORMANCE EVALUATION

The performance evaluation of SDCOF framework are executed in cloudsim environment. Picaso and Mathdroid applications are taken for consideration. The candidate process is project method of ReadMatLab class and for Mathdroid is ComputeAnswer of MathDroid class. Factors are remained stable at LAN setting. Server CPU availability is at 2GHZ and RAM at 1GB. Latency is fixed at 20 ms and the bandwidth is 100 mpbs. The trials are evaluated for 50 instances of the application in terms of energy consumption and response time.

A. Response Time

Table 1: Energy Consumption (j)

Application	Energy Consumption versus Methods		
	Energy Consumption(j)		
	MPOMAC	EXTRADE with MFCMC	SDCOF
Picaso	17	14	13
MathDroid	1.7	1.5	1.2

Table 1 displays the evaluation of MPOMAC, EXTRADE with MFCMC and SDCOF with respect to Energy Consumption.

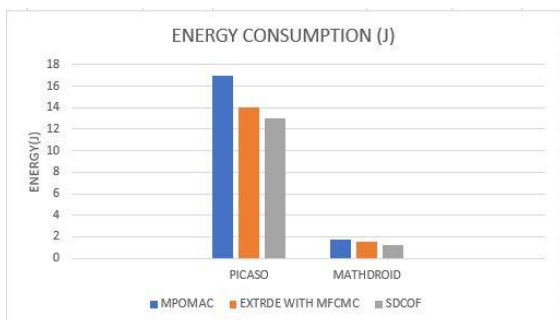


Figure 2: Energy Consumption (j)

Figure 2 shown that for Picaso and MathDroid application, the energy consumption for offloading by means of SDCOF is minimized than that of previous works.

B. Response Time

Table 2: Response Time (ms)

Application	RESPONSE TIME versus Methods		
	Response Time(ms)		
	MPOMAC	EXTRADE with MFCMC	SDCOF
Picaso	490	400	360
MathDroid	90	70	50

Table 2 displays the evaluation of MPOMAC, EXTRADE with MFCMC and SDCOF with respect to Response Time.

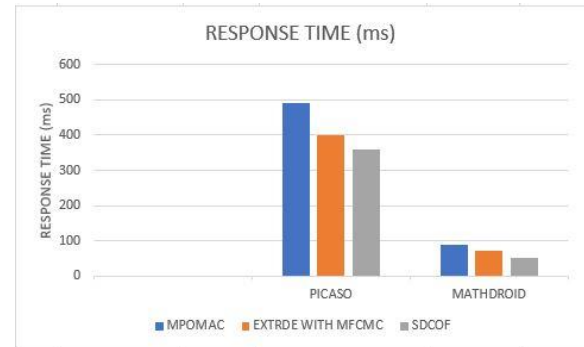


Figure 3: Response Time (ms)

Figure 3 shown that for Picaso and MathDroid application, the response time for offloading by means of SDCOF is minimized than that of previous works.

V. CONCLUSION

Offloading mobile applications to cloud is not practically acceptable if response time and battery consumption is too high when mobile devices execute compute or resource intensive applications. SDCOF Model is proposed in this research which provides low response time and minimize battery consumption by including accurate decision-making process for application execution between cloud and mobile devices. This model predicts more accurate estimated task processing time by analysing the dynamic behaviour of the mobile application. The proposed model is predicted to achieve more accurate decision-making process, time and energy saving can be obtained. In future, real time applications which involve VM placement in clouds will be used to evaluate the results.

REFERENCES

1. G. Chun, Sunghwan Ihm, Petros Maniatis, Mayar Nasik and, Ashwin Patti, "Clonecloud: elastic execution between mobile device and cloud", Proc. Sixth Conference on Computer Systems, pp. 301-314, 2011
2. Cuervo Aruna Balasubramanian, Dae-ki cho, Alec Wolman, Stefan Saroiu, Ranveer chandra, and Paramvir Bahl, "Maui: Making Smartphones Last Longer with Code Offload", Proc. ACM MobiSys 2010, San Francisco, CA, June 15–18, 2010.
3. S. Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and, Xinwen Zhang, "Thinkair: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading," Proc. IEEE INFOCOM, Orlando, FL, pp. 25– 30, 2012
4. R. Kemp, N. Palmer, T. Kielmann, H. Bal, "Cuckoo: a computation offloading framework for smartphones", Mobile Computing, Applications, and Services, Springer, pp. 59–79, 2010.
5. Deepsubhra Guha Roy, Debashis De, Anwesha Mukherjee, Rajkumar Buyya 2016. Application aware cloudlet selection for computation offloading in multi cloudlet environment. Springer 2016 . pp-1672-1690
6. Satyanarayanan, M., Bahl, P., Caceres, R. and Davies, N.,2009. The case for VM-based cloudlets in mobile computing. IEEE pervasive Computing, 8(4), pp. 14–23.

7. Khoda, M.E., Razzaque, M.A., Almogren, A., Hassan, M.M., Alamri, A. and Alelaiwi, A., 2016. Efficient computation offloading decision in mobile cloud computing over 5G network. *Mobile Networks and Applications*, 21(5), pp.777-792.
8. Anwesha Mukherjee, Debashis De, Deepsubhra Guha Roy., 2016. A Power and Latency Aware Latency Aware Cloudlet Selection Strategy for Multi Cloudlet Environment. *IEEE Transactions on Cloud Computing* 2016, pp.1-14
9. Kemp, R., Palmer, N., Kielmann, T. and Bal, H., 2010. Cuckoo: a computation offloading framework for smartphones.,” *Proceedings of the 2nd International ICST Conference on Mobile Computing, Applications, and Services (MobiCASE 2010)*, pp. 59-79.
10. Zhang, X., Kunjithapatham, A., Jeong, S. and Gibbs, S., 2011. Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing. *Mobile Networks and Applications*, 16(3), pp.270-284.
11. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M., 2010. A view of cloud computing. *Communications of the ACM*, 53(4), pp.50-58.
12. Ma, X., Zhao, Y., Zhang, L., Wang, H. and Peng, L., 2013. When mobile terminals meet the cloud: computation offloading as the bridge. *IEEE Network*, 27(5), pp.28-33.
13. Yan, S.H.L., Shanzhi, C.H.E.N. and Xiang, X.U., 2017. MAGA: A Mobility-Aware Computation Offloading Decision for Distributed Mobile Cloud Computing. *IEEE Internet of Things Journal*, 5(1), pp.164 - 174.