Journal of Contemporary Issues in Business and Government Vol. 27, No. 3,2021
https://cibg.org.au/

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

SURVEY: ENHANCING ENERGY PROFICIENCY IN SMART MOBILE DEVICES USING COMPOSITE OFFLOAD DECISION ALGORITHMS

***Mr. Sridhar S K[1] Dr. AmutharajJ[2]**
Dr. S Vijayanand[3]
[1]Research scholar, Dept. Of CSE, BITM,
Ballari.
[2]Professor & Head, Dept. Of ISE, RRCE,
Bengaluru.
[3]Professor, Dept. Of CSE, RRCE,
Bengaluru.
sridharsk.bitm@gmail.com,amutharajj@yahoo.com,
kgsvanand@gmail.com

ABSTRACT: The resource limited mobile devices are the foremost confronter for the growth of Mobile Computing. Although the advance development of mobile applications are able to provide substantial benefits to end users as and when required through continuous online services and accessibility, it has become quite formidable to relish the mobile computing services to its maximum capability due to energy paucity and non availability of effective and collaborative decision making component. So the combination of local, remote mobile cloud computing and a quick collective offload decision that makes a flawless task offloading can intensify to novel computing as an added feature to mobile cloud computing (MCC) to enhance mobile device performance and utilize the available energy proficiently. The proposed framework consists of three major components namely, mobile client, local mobile device cloud (LMC) and remote cloud. These components communicate with each other to generate a composite offload decision based on several system parameters that accounts to enhanced energy proficiency and performance.

Keywords: MCC, LMC, VM, REQ, RES

1. Introduction

The MCC is a standard model for mobile applications where the selective data to be processed and storage can be taken away from the mobile device to potential and centralized computing digital platforms managed in large data centers.

Figure 1. Basic MCC View

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

Once the offloaded data is processed to desired information it can be accessed over the wireless network connection based on a thin native client or web browser on the mobile devices by reducing the burden on local device processor component. The MCC extends the effective utilization of limited resources available on mobile devices to execute complex and rich mobile applications using appropriate composite offload algorithms. The mobile computing is applicable to much wider range of mobile subscribers as most of the businesses use software as a service (SaaS) applications for their business via mobile. So MCC will be an integrated part of business economy in near future. The gradation of wireless network bandwidth used by most mobile applications is lesser than wired networks. Interim, the execution of computationally complicated programs increases on the mobil e devices-for example, image processing, gaming, social networking, educational, augmented reality and other multimedia applications run on mobile phones. So, there is a larger gap between the significant demand for rich program execution and the availability of limited resources on mobile devices. Therefore, the concept of computation offloading technique is necessary with the prime goal of transferring the large computations and heavy CPU bound tasks from resource constrained mobile devices to resourceful server systems situated in large data centers. This reduces the burden on execution component of mobile device by avoiding longer application run time which leads to large amount of energy consumption thereby ensuring system utilization to be more energy proficient with optimal performance.

.

2. Related Work

The static and dynamic offloading approaches are effective to upgrade the potential of smart mobile device by conserving energy, minimizing the time of response, or reducing the overall execution cost. There is a need to deal with an issue of non availabil ity of quality architectures and efficient offloading algorithms [39]. A most energy proficient data offloading option can be built based on determination of the devices that are most probable to respond in quick time. This information can be extracted from social context and contact history of the device owner and the device itself [38]. The prime goal of min-cost offloading partitioning (MCOP) algorithm is to develop the optimal partitioning plan for different cost models with environment adaptability and low time complexity features. This optimal task distribution method between mobile devices and remote servers minimizes process time and saves energy [37]. The factors affecting MCC such as system computation power, network bandwidth, device energy and data security need advance research to make effective offloading decisions which are more realizable in mobile cloud with no performance deterioration [28]. Load balancing technique can focus on migrating the currently running tasks from low load system to the medium load systems and can push low load systems to standby mode saving considerable device energy [27]. A dynamic offload decision is taken to either migrate or not based on the result of comparison of task deadline time with the CPU

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

time of task on mobile device and on the cloud. Every task is labeled with its task migration cost computed on offload completion [26]. The success of offloading scheme depends on its capability to decide on energy savings and provide stable services. This requires the detailed study of issues in system interoperability, fault tolerant, context awareness and user mobility [25]. To avoid unnecessary energy consumption and data transmission delay, the smart mobile devices can make use of cloudlet or fog kind of architecture situated closer to them to offload its tasks when no real cloud connection is possible. A cloudlet is a collection of several multi core systems which act as a small scale data center placed on nominated regions and is connected to a larger data cloud serve r via the Internet [24]. Offloading high complexity tasks from mobile devices to the remote cloud is worthwhile. Several experiments are conducted using different network interfaces say, 3G, 4G, and Wi-Fi and the outcome unveiled the cloud capability to minimize the energy consumption by 60 to 90% for 4G and Wi-Fi networks respectively. The Input file size is the major criteria to make a data offload decision that directs toprocess it either locally on the node, or migrate the file to the remote cloud server [4]. The offloading services presented in ad hoc mobile cloud and the mobile opens up the beneficial space for end user in MCC in terms of infrastructure and its maintenance. The data can move from one region to another making the computation environment more complicated and less predictable [18]. The proposed MEC network is capable of managing different time constrained computation tasks at various wireless devices. Every task execution area can be either local wireless device or edge servers or real cloud server based on low-complexity computation offloading policies investigation which in turn guarantees quality of service by reducing wireless device power consumption. The Linear programming relaxation-based (LR-based) algorithm and a distributed deep learning-based offloading (DDLO) algorithm are applied for MEC networks separately for detailed studies. After repeated experiments, the numerical statistics reveal that DDLO algorithms provide efficient performance and are able to make an offloading decision within one millisecond [2]. The importance of planning and analysis of the point when the task offloading decision has to be made on several different environments is presented. A problem statement is formulated by taking both device energy consumption and task run time delay. Then, it employs enumerating and Branch-and-Bound algorithms to generate the optimal or near-optimal decision for reducing the overall system cost. This result in high accuracy output generation in quick time. Thus it is more prefe rred for real world applications. Artificial intelligence can be incorporated alongside to improvise execution time at faster rate. [1].

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

2.1. Major Catalyst Factors for Cloud Offload decision Computing are identified as below [39]:

- **User Preferences:** Based on user data sensitivity & confidentiality.
- **Server Specifications:** Based on CPU speed, processor load and Memory & storage availability on server side.
- **Application Execution Time:** Based on the CPU burst time requirement of an application to be executed.
- **Mobile Client Specifications:**Based on Battery level, CPU speed, processor load and Memory & storage availability on client side.
- **Network Specifications:**Based on Cellular or Wi-Fi & available bandwidth.

2.2. Traditional cloud offloading decision frameworks:

2.2.1. Static Offloading Framework:

The process of partitioning the application is carried out at the time of design. Therefore low precision because they do not take into account the actual execution context [29].

2.2.2. Dynamic Offloading Framework:

A system focused on user annotations of off-loadable components necessarily indicated by application developers and pre-installed components on a remote cloud server to decide whether to offload modules or not at runtime. Since decisions are made at runti me, there is more comprehensive information available [29].

2.2.3 Virtual Machine Cloning:

This is a platform where the full image of the mobile application is recorded on the remote cloud server and stored there. The execution of the mobile device application is deferred during offloading and migrated to the VM clone in the cloud [30].

- **Program profiling:** It is the method of extracting program metadata factors say power requirement, actual data size, run time, and memory use that helps in code offload decision.
- **Device profiling:** It is a method of gathering the status information of mobile device in terms of energy availability, processor use, current workload and wireless connection for communication.

2.2.4 Offload VM Load Framework:

It redistributes the application workload partitions between different Virtual Machines (VM's) based on their distance to the current VM [32]. This reduces the workload burden on the current VM. There by increasing the application execution performance and speed. The Buffer Allocation Method (BAM) is proposed to remove the data redundancy and no duplicate data is transferred during offloading computation. This implies faster rate execution, lesser energy consumption and the workload get offloaded in a sensible manner [31].

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

3. Recent advances in cloud offload computing schemes

3.1. Based on Machine learning approach

Ref. No.	OFFLOAD DECISION CRITERIA	ALGORITHM/APPROACH
[3]	Based on Task Size Prediction. Required CPU cycles, Max Uplink, Total Transmission delay are evaluated. Divides ongoing task into subtasks and each sub task are wholly executed on a node.	Deep Learning based on LSTM. Sub Task Migration Algorithm.
[4]	Adaptable Task Allocation. Energy consumption estimator.	Decision Tree, K-NN. Power Profiler API.
[7]	Based on the history of node Request & Response time in network. It is able to predict future request response times.	Deep Belief Network using Regression Layer
[9]	Predicts Real Time SBS Traffic. Operates smoothly on single-SBS as well as	Deep learning method with M-LSTM
	on multi-SBS circumstances.The normalized mean-squared error (NMSE) is applied to characterize the performance of traffic prediction model.	Cross Entropy Method
[14]	Based on Past Experiences stored through rewards. Follow Me Cloud (FMC) Controller makes precise decision on past experiences stored through rewards. Transition Probability is not required.	Task Migration with Deep learning Q- network.
[17]	Selection of task components for offload is based on residual energy of the mobile device, energy usage, network constraints, computational workload, and data	Deep Learning Offload Scheme

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

	transmission &communication delays.	
[40]	Based on estimation of offload system ben efits and costs at a certain point in an application.	A runtime performance prediction generator. Global optimization for code partitioning. Mobile data offload solver.

3.2. Based on Task Scheduling approach

Ref. No.	OFFLOAD DECISION CRITERIA	ALGORITHM/APPROACH
[5]	Based on Mobile device battery level AJAS assigns task to master slave nodes with dynamic computer resources allocation.	AJA Scheduler Framework
[11]	Considers Task Position, Execution Sequence,Operating Voltage and Frequency of Mobiles	Task Workflow Scheduling Whale Optimization
[15]	Based on Task division and Sub Task Cluster creation	Integer particle swarm optimization (IPSO)-based algorithm. Cluster Scheduling
[16]	Based on Task Pattern, Server Selection through Auction andPaymentdetermination	Adaptive offloading Auction and Task Scheduling. High Computational Efficiency and Individually rational.
[33]	Distributes Tasks to Aura cloud with payment Initiation.	Task Offload & distribution model using Map-Reduce. Decision Algorithm at both Mobile agent and IOT device end is installed

3.3. Based on Congestion awareness approach

Ref. No.	OFFLOAD DECISION CRITERIA	ALGORITHM/APPROACH

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

[8]	Based on Congestion Degree & Server Core occupied info.Allocates the jobs of a newly arrived program for execution to coresunused server to prevent the disruption of programs under execution.	Congestion Awareness Framework. Multi Objective Uniform diversity Genetic Algorithm.
[14]	Considers Task arrival probability, Average energy and deadline for execution.	Dynamic Parallel Compute Offload & Energy Management(DPCOEM).

3.4. Based on Server specification and capability

Ref. No.	OFFLOAD DECISION CRITERIA	ALGORITHM/APPROACH
	Server Idle Time Slots adjustment &	Load Aware resource allocation & task
[21]	Migration by task rescheduling. The newly arrived applications gets their desired resource allocation for execution using MRATS TGTB assists applications by providing cloudlet facility, when no direct cloud communication not possible. TCMO smoothen the cloudlet overload problem through rescheduling of tasks of application.	scheduling has 3 sub approach: Multi-objective resource allocation and task scheduling scheme(MRATS). Tree generation based idle slots of time (TGTB). On cloudlet overload, only delay-tolerant application activities can be offloaded (TCMO).

3.5. Based on User Mobility awareness approach

Ref. No.	OFFLOAD DECISION CRITERIA	ALGORITHM/APPROACH
[12]	Based on User Mobility and Servers use in same direction.	Task Deadline Aware. Mobile edge server(s) termed as fog can be generated for every valid candidate task using fog generation algorithm. Efficient mobile edge server is chosen and corresponding offload

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

		ratio is computed.
[13]	Considers Task arrival probability and Execution deadline	Dynamic Parallel Compute Offload & Energy Management(DPCOEM)
[23]	Based on Task size, SBS anduser mobility information	User Mobility Aware. The heuristic task Assignment algorithm. Accurate delay estimation scheme.

3.6. Based on Mobile Client device Energy Estimation

Ref. No.	OFFLOAD DECISION CRITERIA	ALGORITHM/APPROACH
[6]	Local Mobile Cloud Energy Sharing Effect. A local device with limited battery status can use reasonably small cloud with few remote servers to manage its energy proficiency.	ColloboRoid Architecture Device Management support Remote Resource Accessibility policy.
[14]	Based on average energy for mobiles.	Dynamic Parallel Compute Offload & Energy Management(DPCOEM)
[36]	Low residual Energy mobile clients are preferred first. Performance Comparison changed by cloudlet environment and number of clients.	Energy driven job scheduling and weight allocation. Local computation power driven priority allocation. Iterative method to figure out optimized policy.

3.7. Based on communication path

Ref. No.	OFFLOAD DECISION CRITERIA	ALGORITHM/APPROACH
[9]	Analyze all AP route and Predict the Shortest Path to reduce the offloading time.	Origin-destination AP routing Non-dominated sorting genetic algorithm

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

	Conducts multi objective optimization to minimize the power consumption of edge computing nodes.	scheme. A Multiple criteria decision marking scheme (MCDM). A Simple additive weighting (SAW) method.
[20]	By going through all possible routes, a specific sized task is assigned to the path with improved network conditions in the future while meeting the delay constraint.	Dynamic Mobile Aware Partial Offloading Algorithm (DMPO). Compares DMPO with partial and dynamic partial offloading with mobility management schemes.
[35]	Based on Different File Size input to 3G, 4G & Wi-Fi cloud.	MECCA Rule based approach Power consumption and Processing Time are evaluated for different Task Size and different network interfaces

3.8. Based on Cache Management

Ref. No.	**OFFLOAD DECISION CRITERIA**	**ALGORITHM/APPROACH**
[19]	More Common results of computation are stored on server cache. Beneficial to offload the components to the server since it minimizes the overall execution time delay.	Cache Enhancement for optimum data offloading. Multi-user approach is beneficial compared to single users through the collaboration between users.

3.9. Based on Application factors

Ref. No.	**OFFLOAD DECISION CRITERIA**	**ALGORITHM/APPROACH**
[22]	Based on "@offload" annotation in code by developers with expertise knowledge. Simulation Resulted in less execution time and energy consumption.	User Level Online Offloading Framework. Automated offload able method selection algorithm Annotated Method Profiler
[34]	Based on Application size and complexity.	ARM Emulation Framework over cloud server.

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

[41]	Based on program and device profiling information.	Program and device profiler Jade Optimizer approach.

4. Research Findings and their Issues

After extracting the working procedure of different cloud offload computing schemes, their offload decision criteria to minimize data transfer time, execution time and energy consumption and the algorithms used to evaluate the significant aspects in prioritizing the features for next level implementation, now we have several issues presented in line with the schemes discussed as follows:

4.1. Issues in Machine learning approach

- Content caching problem not considered.
- An Algorithm that predicts User Mobility based on Artificial Intelligence is required.
- If Data size increases then Local processing time increases.
- If computing load increases then Edge node processing time increases.
- Subtask Migration reduces a portion of processing delay by compromising delay in data transmission.
- Experiments for different type of tasks to be conducted.
- Experiment on other network interfaces like 4G, 5G need to be tested.
- Need to experiment in Real World Scenario to check its suitability.
- Data Encryption on Migration needs to be considered.
- Multi user scenario not considered.
- Need to experiment with several different applications.

4.2. Issues in Task scheduling approach

- The Adaptive Job allocation Scheduler (AJAS) essentially requires IOT device batterystatus information, list of application, performance capability for computational offload, and job allocation table data.
- Focuses more on Job Reallocation.
- More Critical Factors such as Network Bandwidth, resourceful remote server, user mobility, server error and environmental adaptability has to be taken into account.
- Efficient allocation of Communication and computation resources must be considered.
- Need to check the suitability to Real World scenario.
- Not applied to multiple applications with multiple deadlines for execution.
- Live Migration Not Implemented
- Optimal Scheduling Algorithms Required.

4.3. Issues in Congestion awareness approach

- Simulation not carried with cooperation among multiple edge servers
- Need to consider other smart mobile devices.

4.4. Issues in Server specification and capability

- VM Management is not considered
- Unreliable network connections are not considered

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

- User Mobility Problem is not considered.

4.5. Issues in User Mobility awareness approach

- Estimation of Task Execution Time need to be has accuracy.
- Need to consider other smart mobile devices.
- Need to work for multiple tasks.

4.6. Issues in Mobile Client device Energy Estimation

- Experimented with Audio, Video and GPS info files.
- Local device saves its energy on small cloud set with minimum remote devices.
- Pareto-Optimal.

4.7. Issues in communication path

- Need to use minimal number of Edge Computing Nodes (ECNs) between which the task migrates.
- Simulation not carried for base stations that are unevenly deployed
- Not optimal when there is a change of network condition.
- Allows the decision making considering only single position network condition.
- From several tests using different network interfaces such as 3G, 4G, and Wi-Fi, the findings have shown the cloud's ability to minimize the power consumption for Wi-Fi and 4G networks.

4.8. Issues in Cache Management

- As the application run time increases, the most common computation results are cached in the server.
- Thus, the end user prone to migrate the components to the server to minimize the overall run time delay.
- Simulated in MATLAB and not in real life scenario

4.9. Issues in Application factors

- User Mobility needs to be considered to have more prediction accuracy.
- Multiple User and server operations needed to be considered.
- ARM Emulation Optimization is needed.
- Proper analysis of communication and computational overhead can automate the offloading decision.
- Diversity of operating systems in mobile devices.
- Access a distributed platform transparently
- Cloud platforms are not considered

5. Proposed Research

5.1. Action Plan for mobile client device:

5.1.1. Classification of Application/task:

There are 3 different classes namely low, medium and high level application based on complexity. As and when the task execution request arrives, it is made sure that to which class it actually belongs to based on content profiling.

5.1.2. Local In device application execution for low complexity and sensitive applications:

There is no data offloading at this low class level. The In device application execution is preferred for all low complexity level applications and sensitive,

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

confidential data program execution.

5.1.3. Establish LMC communication for medium complexity application:

The LMC is established for applications classified as medium complexity for execution. It is a wireless connectivity between the nearby mobile devices registered for mutual cooperation and coordination in program execution.

5.1.4. Use real cloud offloading for high complexity application:

The **real** cloud offloading from local client device for applications classified as high complexity for execution. This real cloud offloading can also be initiated from LMC on behalf of local client device whose battery power is not sufficient to perform the offload mechanism.

5.2. Action Plan for Local Mobile Cloud established:

The LMC gets activated on receiving application program execution request from local client device registered and connected to it.

- One of the members of LMC can accept the request and respond to the original client.
- It can make an offload decision either to execute the application request on local client device or on remote cloud server on behalf of local client device whose battery power is not sufficient to perform the offload mechanism.
- It can also set the destination path of original client device to receive the real cloud response on completion of program execution to avoid unnecessary network congestion.

5.3. Action Plan forReal Remote Cloud Server:

The real cloud server must always prefer the mobile client request with low energy status tag for execution. Only high complexity applications should be offloaded to real cloud server.

- The cloud server must adopt content caching mechanism.
- The cloud server should process the preferred mobile client request first.
- The cloud server should send response to original client path.

5.4. Action plan for application with or without annotations:

- The mobile client can offload the data request to the server based on annotations build with high quality software or applications.
- The mobile client can offload the data request without annotations by profiling it to classify into different classes and then choose an appropriate server for execution.
- It’s preferred and is an added advantage if developers can provide annotations with quality software to ease the offload decision making mechanism.

6. Proposed Architecture

The proposed system architecture consists of three major components namely, mobile client device, LMC and remote cloud server:

6.1. Client Side Algorithm (Smart mobile):

This algorithm uses 3 system parameters such as content, user, & Battery

Journal of Contemporary Issues in Business and Government Vol. 27, No. 3,2021
https://cibg.org.au/

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

profiles to make appropriate offload decision.

6.2. Server Side Algorithm (Remote cloud):

Once the client server connection is established and Offload decision is made from client end. The Server side algorithm gives its acceptance based on User, Battery, cloud profile and Quality of Service that it can provide to enhance energy proficiency in smart phones. Therefore, the mutual agreement between client and server plays a crucial role in final offload decision.

6.3. Local mobile device cloud:

It is a network of connected mobile devices, where several medium level tasks can be offloaded between them and sometimes to remote cloud. An offload request to remote cloud on behalf of mobile client can also be generated from LMC and set original client communication path to directly send the final result response redirection to save transmission energy.

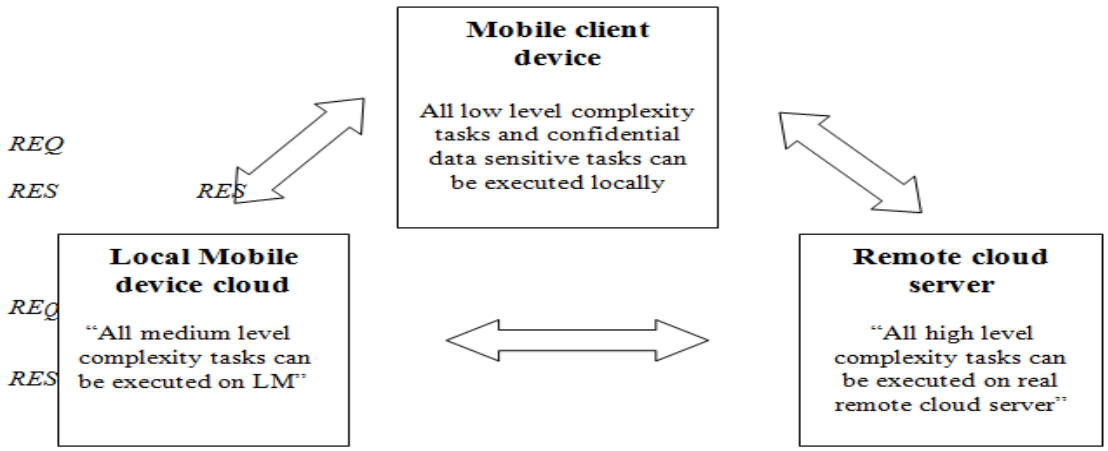

Figure 2. Proposed System Architecture

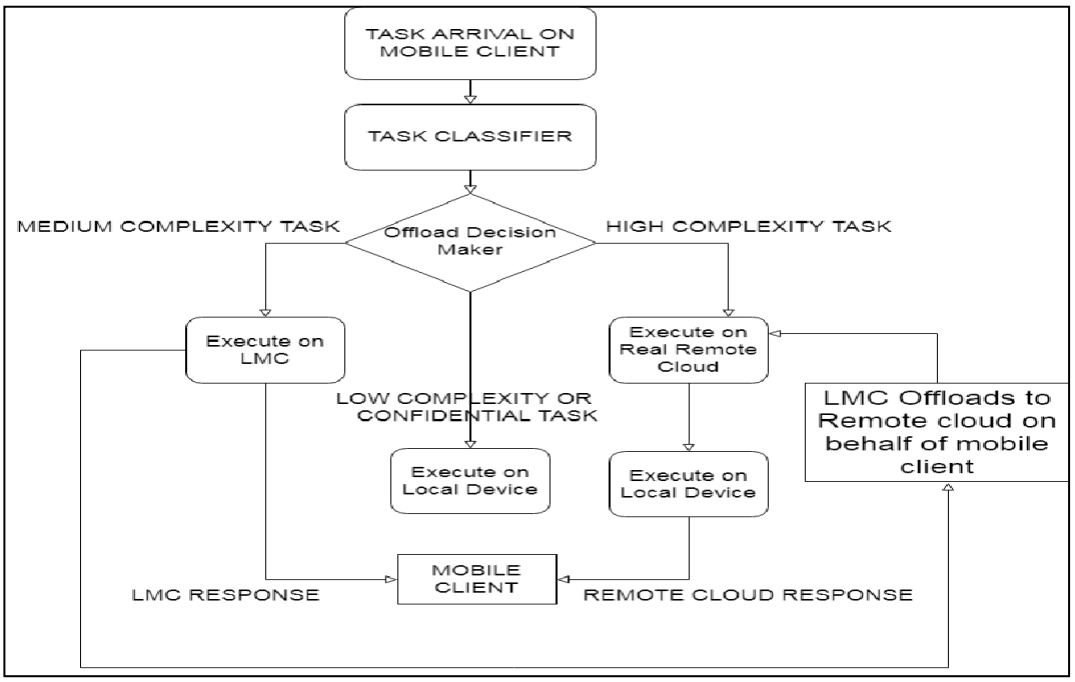

Figure 3. Workflow of Proposed Framework

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

Figure 3. Shows the classification of given user request into different levels and a appropriate execution approach is chosen to each request type using composite offload decision for execution.

6.4. Preferred System parameters are:

- **Content Profile:** This profile contains the parameters that describe the task content affects the energy consumption of the task.
- **User Profile:** It includes the parameters that are user dependent. It may include networking status (enabled/disabled) according to the needs & their location and the maximum time that the user allows for the offloading.
- **Battery Profile:** It determines the remaining energy on the device battery at any instant and ensures the completion of offload process followed by request execution.
- **Cloud Profile:** It consists of the cloud server status information and its ability to provide the desired Quality of Service (QoS) at current time event.
- **Local execution:** User specified request is executed with local processor of Mobile device.

- **Remote cloud execution:** Refers to a data centre with many resourceful remote servers able to execute client side heavy computational requests.

As the entire Offload logic is to be implemented on both Client side and Server side, an effective support of local database and communication manager is necessary.

A major requirement that arises from this inference is that the data or code that will be transferred must be structured in such a manner that identification and categorization of data can be done effectively at both the ends over network.This helps majority of apps which still perform most of the complex computation and stores related data on the mobile devices themselves and not in the cloud there by reducing significant amount energy consumption with improving in performance.

Acknowledgment

My sincere gratitude to Dr. Amutharaj J for being my research supervisor, well wisher and one who always motivates me to take a deeper look at the research findings and take it to next level by setting new goals of professional research and progress in career.

Conclusion

The proposed work aims to design and adopt a combined offload decision framework. It is expected to have reduction in the overall time taken to execute applications in remote cloud or local mobile device cloud and send back results to mobile device client. An effective classification and optimal task distribution based on user offload annotations in high quality software applications by expert

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

developer leads to mobile power & energy optimization. This eases the burden on client side to make quick offload decision and reduces the time taken to partition code for computational offloading. It is extremely useful when local mobile client consumes more time and energy in program execution. It is ideal to the mobile devices where energy proficiency is more important than offload time consumption. An easy and effective idea to make the offloading decision periodically to identify which portions of the application are appropriate for running on mobile devices and on cloud servers.

References

[1] xu, Jiuyun & Hao, Zhuangyuan & Sun, Xiaoting, "Optimal Offloading Decision Strategies and Their Influence Analysis of Mobile Edge Computing", July 2019, Sensors, pp. 1-19, 3231, DOI: 10.3390/s19143231.

[2] Huang, Liang & feng, Xu & Zhang, Luxin & Qian, Li Ping & Wu, Yuan. ,"Multi-Server Multi-User Multi-Task Computation Offloading for Mobile Edge Computing Networks",Mar 2019, Sensors, pp.1-19, 1446. DOI: 10.3390/s19061446.

[3] Miao, Yiming & Wu, Gaoxiang & Li, Miao & Ghoneim, Ahmed & Alrakhami, Mabrook & Hossain, M. Shamim. (2019). Intelligent task prediction and computation offloading based on mobile-edge cloud computing. Future Generation Computer Systems. 102. 10.1016/j.future.2019.09.035.

[4] Nawrocki, Piotr & Śnieżyński, Bartłomiej & Słojewski, Hubert. (2019). Adaptable mobile cloud computing environment with code transfer based on machine learning. Pervasive and Mobile Computing. 57. 49-63. 10.1016/j.pmcj.2019.05.001.

[5] Kim, Hyun-Woo & Park, Jong & Jeong, Young-Sik. (2019). Adaptive job allocation scheduler based on usage pattern for computing offloading of IoT. Future Generation Computer Systems. 98. 10.1016/j.future.2019.02.071.

[6] Lee, Hochul & Lee, Jaehun & Lee, Young & Kang, Sooyong. (2019). CollaboRoid: Mobile platform support for collaborative applications. Pervasive and Mobile Computing. 55. 10.1016/j.pmcj.2019.02.006.

[7] Alelaiwi, Abdulhameed. (2019). An efficient method of computation offloading in an edge cloud platform. Journal of Parallel and Distributed Computing. 127. 10.1016/j.jpdc.2019.01.003.

[8] Guo, Kai & Yang, Mingcong & Zhang, Yongbing & Jia, Xiaohua. (2019). Efficient resource assignment in mobile edge computing: A dynamic congestion-aware offloading approach. Journal of Network and Computer Applications. 134. 10.1016/j.jnca.2019.02.017.

[9] Zhao, Xianlong & Yang, Kexin & Chen, Qimei & Peng, Duo & Jiang, Hao & Xu, Xianze & Shuang, Xinzhuo. (2019). Deep learning based mobile data offloading in mobile edge computing systems. Future Generation Computer Systems. 99. 10.1016/j.future.2019.04.039.

[10] Xu, Xiaolong & Li, Yuancheng & Huang, Tao & Xue, Yuan & Peng, Kai & Qi, Lianyong & Dou, Wanchun. (2019). An energy-aware computation

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

offloading method for smart edge computing in wireless metropolitan area networks. Journal of Network and Computer Applications. 10.1016/j.jnca.2019.02.008.

[11] Peng, Hua & Wen, Wu-Shao & Tseng, Ming-Lang & Li, Ling-Ling. (2019). Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment. Applied Soft Computing. 80. 10.1016/j.asoc.2019.04.027.

[12] Tang, Wenda & Zhao, Xuan & Rafiq, Wajid & Qi, Lianyong & Dou, Wanchun & Ni, Qiang. (2019). An Offloading Method Using Decentralized P2P-Enabled Mobile Edge Servers in Edge Computing. Journal of Systems Architecture. 94. 10.1016/j.sysarc.2019.02.001.

[13] Zhang, Cheng & Zheng, Zixuan. (2019). Task migration for mobile edge computing using deep reinforcement learning. Future Generation Computer Systems. 96. 10.1016/j.future.2019.01.059.

[14] Deng, Yiqin & Chen, Zhigang & Yao, Xin & Hassan, Shahzad & Ibrahim, Ali.M.A.. (2019). Parallel Offloading in Green and Sustainable Mobile Edge Computing for Delay-constrained IoT System. IEEE Transactions on Vehicular Technology. PP. 1-1. 10.1109/TVT.2019.2944926.

[15] Liu, Jianhui & Zhang, Qi. (2019). Code-Partitioning Offloading Schemes in Mobile Edge Computing for Augmented Reality. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2891113.

[16] Luo, Shuyun & wen, yuzhou & xu, weiqiang & puthal, deepak. (2019). Adaptive Task Offloading Auction for Industrial CPS in Mobile Edge Computing. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2954898.

[17] Ali, Zaiwar & Jiao, Lei & Baker, Thar & Abbas, Ghulam & Abbas, Ziaul & Khaf, Sadia. (2019). A Deep Learning Approach for Energy Efficient Computational Offloading in Mobile Edge Computing. IEEE Access. 7. 1-1. 10.1109/ACCESS.2019.2947053.

[18] Mr. C.arun, Dr. K.prabu, "Survey on three components of mobile cloud computing: mobile cloudlets, context aware service and privacy", IJCEA, UGC approved, Volume XII, issue 1, Jan-2018, ISSN 2321-3469.

[19] Yu, Shuai & Langar, Rami & Fu, Xiaoming & Wang, Li & Han, Zhu. (2018). Computation Offloading With Data Caching Enhancement for Mobile Edge Computing. IEEE Transactions on Vehicular Technology. PP. 10.1109/TVT.2018.2869144.

[20] Yu, Fangxiaoqi & Chen, Haopeng & Xu, Jinqing. (2018). DMPO: Dynamic mobility-aware partial offloading in mobile edge computing. Future Generation Computer Systems. 89. 10.1016/j.future.2018.07.032.

[21] Zhang, Feifei & Ge, Jidong & Li, Zhongjin & Li, Chuanyi & Wong, Chifong & Kong, Li & Luo, Bin & Chang, Victor. (2018). A load-aware resource allocation and task scheduling for the emerging cloudlet system. Future Generation Computer Systems. 87. 10.1016/j.future.2018.01.053.

[22] Neto, Jose & Yu, Se-Young & Macedo, Daniel & Nogueira, José Marcos & Langar, Rami & Secci, S.. (2018). ULOOF: a User Level Online Offloading

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

Framework for Mobile Edge Computing. IEEE Transactions on Mobile Computing. PP. 1-1. 10.1109/TMC.2018.2815015.

[23] Wang, Zi & Zhao, Zhiwei & Min, Geyong & Huang, Xinyuan & Ni, Qiang & Wang, Rong. (2018). User mobility aware task assignment for Mobile Edge Computing. Future Generation Computer Systems. 85. 10.1016/j.future.2018.02.014.

[24] Srilatha, s. Rajeshwari and Kanu rani. "A survey on applications of cloudlet infrastructure in mobile cloud computing." (2017),International journal of latest trends in engineering and technologyvol.(8)issue(1), pp.309-318,doi:http://dx.doi.org/10.21172/1.81.040, e-issn:2278- 621x.

[25] Xiumin Wang; Xiaoming Chen; Weiwei Wu, "Towards truthful auction mechanisms for task assignment in mobile device clouds", on Computer Communications, 2017, DOI: 10.1109/INFOCOM.2017.805, 7198, pp. 1-9.

[26] Gaurav Setia, Raj Kumari, Veenu Mangat,"User preference driven offloading scheme for mobile cloud computing", 4th International Conference on Signal Processing and Integrated Networks (SPIN), 2017, pp. 524 – 529.

[27] Pavel Mach, Zdenek Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading", IEEE Communications Surveys & Tutorials, Volume: 19, Issue: 3, third quarter 2017, pp. 1628-1656, DOI: 10.1109/COMST.2017.2682318.

[28] Paranjothi, Anirudh & Khan, Mohammad Shoeb & Nijim, Mais. (2017),"Survey on Three Components of Mobile Cloud Computing: Offloading, Distribution and Privacy", Journal of Computer and Communications. 5.,pp. 1-31, DOI: 10.4236/jcc.2017.56001.

[29] AraniBhattacharya, Pradipta De, "A survey of adaptation techniques in computation offloading", Journal of Network and Computer Applications, Volume 78, Elsevier, 15 January 2017, pp. 97-115, 2016, DOI: 10.1016/j.jnca.2016.10.023.

[30] Jiaying Meng, Wenbin Shi, HaishengTan, Xiangyang Li, "Cloudlet Placement and Minimum- Delay Routing in Cloudlet Computing", 3rd International Conference on Big Data Computing and Communications (BIGCOM), IEEE, 2017, pp. 297-304, DOI: 10.1109/BIGCOM.2017.58.

[31] Mohammad Goudarzi, Mehran Zamani, Abolfazl Toroghi Haghighat, "A fast hybrid multi-site computation offloading for mobile cloud computing", Journal of Network and Computer Applications, Volume 80, Elsevier, 2017, Pages 219-231, DOI: 10.1016/j.jnca.2016.12.031.

[32] Liqing Liu, Zheng Chang, Xijuan Guo, Shiwen Mao, Tapani Ristaniemi, "Multi-objective Optimization for Computation Offloading in Fog Computing", IEEE Internet of Things Journal, Volume: PP, Issue: 99, pp. 1-12, 2017, DOI: 10.1109/JIOT.2017.2780236.

[33] Hasan, Ragib & Hossain, Mahmud & Khan, Rasib. (2017). Aura: An incentive-driven ad-hoc IoT cloud framework for proximal mobile computation offloading. Future Generation Computer Systems. 86.

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.234

10.1016/j.future.2017.11.024.

[34] Shuja, Junaid & Gani, Abdullah & Naveed, Anjum & Ahmed, Ejaz & Hsu, Robert. (2017). Case of ARM Emulation Optimization forOffloading Mechanisms in Mobile Cloud Computing. Future Generation Computer Systems. 76. 407-417. 10.1016/j.future.2016.05.037.

[35] Aldmour, Rakan & Yousef, Sufian & Yaghi, Mohammad & Kapogiannis, Georgios. (2017). MECCA offloading cloud model over wireless interfaces for optimal power reduction and processing time. pp. 1-8. 10.1109/UIC-ATC.2017.8397639.

[36] Ahn, Sanghong & Lee, Join & Park, Sangdon & Newaz, S.H. & Choi, Jun. (2017). Competitive Partial Computation Offloading for Maximizing Energy Efficiency in Mobile Cloud Computing. IEEE Access. PP. 1-1. 10.1109/ACCESS.2017.2776323.

[37] Wu H., Knottenbelt W., Wolter K., Sun Y. (2016) "An Optimal Offloading Partitioning Algorithm in Mobile Cloud Computing" In: Agha G., Van Houdt B. (eds) Quantitative Evaluation of Systems. QEST-2016, Volume 9826. Springer, Cham, DOI: 10.1007/978-3-319.

[38] Hamid Jadad, Abderezak Touzene, Nasser Alzeidi, Khaled Day, Bassel Arafeh "Realistic Offloading Scheme for Mobile Cloud Computing", International Conference on Mobile Web and Information Systems (MobiWIS), 2016, Volume 9847. Springer, Cham, pp 81-92, DOI: 10.1007/978-3-319-44215-0_7.

[39] Khadija Akherfi, Micheal Gerndt, Hamid Harroud, "Mobile cloud computing for computation offloading: Issues and challenges", Open access, Applied Computing and Informatics (Dec- 2016),

Autho

Mr. Sridhar S K, Research scholar, working as an assistant professor in Ballari Institute of Technology & Management, Ballari. My research work is under progress on Mobile cloud computing. I have undergone expert level workshop on embedded systems by Wipro in 2016.I have received IBM certification on Mobile application development in 2017 and attended an FDP on Research Methodology & LaTex by VTU in 2019. I have received Inspire Faculty Excellence award and participated in Content Guru Contest on Python Programming conducted by Infosys. I desire to contribute society with my research knowledge in mobile cloud interaction enhancements