International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
***Article Received:** 25 March 2023 **Revised:** 12 April 2023 **Accepted:** 30 May 2023*

Enhancing Mobile Efficiency with Cloud-Based Computation Offloading: Critical Issues and Challenges

[1]Imran Khan, [2]Dr. Sumit Bhattachar
[1]Research Scholar, Department of Computer Science, Singhania University, Pacheri Bari, Jhunjhunu, Rajasthan.
[2]Associate Professor, Department of Computer Science, Singhania University, Pacheri Bari, Jhunjhunu, Rajasthan

Abstracts : One efficient method that is built using MATLAB is HEACO, which stands for Heterogeneous Energy Efficient method for Computation offloading. Factors including RAM use, number of tolerable tasks, CPU utilisation, and battery consumption are considered. One way this method measures performance is by looking at the standard deviation. Three viewpoints are used to present the chosen issues: the security viewpoint, the resource-intensive architecture of the current frameworks, and the ideal offloading platform. This method efficiently moves programmes with intensive computations that mobile devices cannot handle to the cloud, all while keeping these criteria in mind.

Keywords: *Mobile, Cloud, Computation, Offloading Critical Issues and Challenges*

INTRODUCTION

You may be wondering how your phone manages to save so much information given its limited storage capacity. Apps like Spotify and Wnyk Music use user-generated content to construct personalized playlists based on song likes. They must be using their lightning-fast processing power on their cloud servers to filter the music and create a playlist that suits your interests. Your smart mobile devices just do not have the processing capability to handle the massive amounts of data that you send them (K. Kumar, 2010). Unbeknownst to the user, cloud computing, intelligent car monitoring access, and global positioning systems are all at work whenever Google Maps is used. It is possible to get real-time information about traffic conditions and the quickest route to a user's specified destination by collecting data about their vehicle's position and speed using GPS and processing it at Google's cloud servers.

By the way, are you OK with sharing media files, whether they're photos or movies, on the cloud? Regarding the security and privacy of your sensitive information, shouldn't you take it seriously? With the use of cloud computing and offloading services, you may access a more powerful and quicker computing resource. This service is pay-as-you-go and available on demand. These three key elements make up cloud computing: platform as a service, software as a service, and infrastructure as a service. The worldwide compounded annual growth rate (CAGR) data demonstrate the dominance of cloud computing. As of 2016, the growth rates of IaaS, PaaS, and SaaS were 41%, 26.6%, and 17.4%, respectively, according to CAGR. As previously said, there has been a significant increase in mobile users. However, this growth is still insufficient to handle or assist with computationally heavy activities like image recognition, deep learning-based decision-making, and natural language processing. Keep in mind that mobile devices can't match to desktops or server PCs when it comes to processing power. Still, portability is the key to mobile cloud computing's success and the amount of effort invested into it.

Various studies are being conducted to improve the computational capacities of SMDs, which are currently facing a problem with limited resources. One example is how the compute offloading framework demonstrated how this practice helped down mobile devices' energy consumption costs. This research primarily covers two points: first, how to increase the computational power and storage capacity of SMDs; and second, such computational offloading frameworks are now available and described. Also, problems with computational offloading and mobile cloud computing. Problems that need exploring further.

LITERATURE REVIEW

Abusaimeh, Hesham. (2022). The growing volume of data places a burden on mobile devices' limited resources, including batteries, processors, storage, and bandwidth. As a result, it becomes important to offload compute-intensive operations to a nearby edge cloud server. A new area of study called Mobile Cloud Computing (MCC) has evolved to solve the problems with mobile devices by moving them to the cloud. Some of the biggest obstacles to effective offloading are the high expense of running and maintaining cloud servers

and the fact that devices are mobile, which might lead to connection problems.

Myagila, Kasian. (2020). There has been a severe lack of memory, computing power, storage space, and battery life in mobile devices. Complex applications have been devised and deployed as a result of technological advancements, and these applications need computer devices with enormous capacities. The concept of cloud computing arose as a remedy for computers with limited storage space. The ever-changing nature of mobile devices and their environments, together with the inherent unpredictability of wireless transmission, has made it difficult to integrate mobile operating systems with cloud computing. In order to evaluate mobile cloud computing as a solution for work offloading, this study surveys current research in the field. Finally, the research brought attention to unresolved concerns that must be resolved in order to execute optimum Mobile Cloud Computing, after reviewing typical frameworks that are utilised to implement Mobile Cloud Computing.

Biswas, Milon & Whaiduzzaman, Md. (2019). Given the meteoric rise of mobile apps and the ever-changing nature of cloud computing, Mobile Cloud Computing (MCC) has emerged as a promising new technology. Mobile cloud computing (MCC) refers to the incorporation of cloud computing into a mobile technology setting. Cloud computing is an architecture that moves data processing and storage from mobile devices to a central location in the cloud. Finding solutions to problems with storage, battery life, bandwidth, heterogeneity, and security are the main goals of mobile computing. The execution offloading method, which transfers processes between computers, is proposed as a means to relieve mobile devices of their computing load. Execution offloading involves statically or dynamically deciding, depending on execution time and process state, which portions of code should be performed. If the offloading mechanism is efficient and fits the energy consumption criteria, then MCC will be a success. To begin, we provide a working definition of the MCC and explain how offloading might reduce runtime and energy consumption. Then, in order to make the most of the cloud, we demonstrated several offloading mechanisms in a wireless heterogeneous network. When the models are to be employed and how they will strengthen and enable the applications were displayed. We concluded by proposing robust MCC application models that would improve the mobile industry of the future.

Akherfi, Khadija & Gerndt, Micheal & Harroud, Hamid. (2016). Even while mobile devices have come a long way, they are still seen as having limited processing capabilities. Users nowadays are pickier and expect to run computationally heavy apps on their mobile phones. So, to increase the functionality of mobile devices via offloading, Mobile Cloud Computing (MCC) combines mobile computing with Cloud Computing (CC). Computation offloading addresses the shortcomings of Smart Mobile Devices (SMDs) such short battery life, low processing power, and inadequate storage capacity by shifting the processing and burden to more powerful and resource-rich platforms. The contemporary offloading frameworks and compute offloading strategies are detailed and evaluated in this study, along with an examination of their key shortcomings. Also covered are certain critical implementation aspects, such as offloading mechanism and amount of partitioning, that determine the frameworks. At last, it provides a synopsis of the problems with MCC offloading schemes that need more study.

RESEARCH METHODOLOGY

We have developed a simulation for this approach in MATLAB. A random quantity of jobs, users, and cloudlets are generated by the method. To efficiently and effectively mimic offloading while minimizing energy consumption, parameters such as standard deviation, execution time, and number of tasks that may be tolerated have been considered. Cloudlets are used in this context. Cloudlets are little virtual computers that enable you move around and scale up or down as needed in the cloud. Cloudlets, which are small-scale cloud datacenters with mobility enhancements, are located, in theory, in the cloud's periphery. Cloudlets were first proposed by Ramon Caceres, Mahadev Satyanarayana, Nigel Davies, and Victor Bahl. The two concepts, clouds and cloudlets, are distinct in important ways. Because it provides the aforementioned services to users over the network, a cloudlet might be considered a kind of cloud computing.

In contrast to clouds, which are overseen by a third party, cloudlets are self-managed. In contrast to clouds, which rely on the public internet for connection building, cloudlets employ local area networks. Applications like as deep learning, virtual reality, and natural language processing—which are resource-intensive and interactive—were the inspiration for the cloudlet. A cloudlet and a cloud are just slightly different. The ownership of the Cloud is centralized, but that of the Cloudlet is decentralized. Cloudlets are considered instead of clouds for the sake of brevity and clarity. Transferring resource-intensive apps from the mobile device to the cloudlet is shown by the offloading procedure. In the MATLAB model, the offloading procedure is shown.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
Article Received: *25 March 2023* ***Revised:*** *12 April 2023* ***Accepted:*** *30 May 2023*

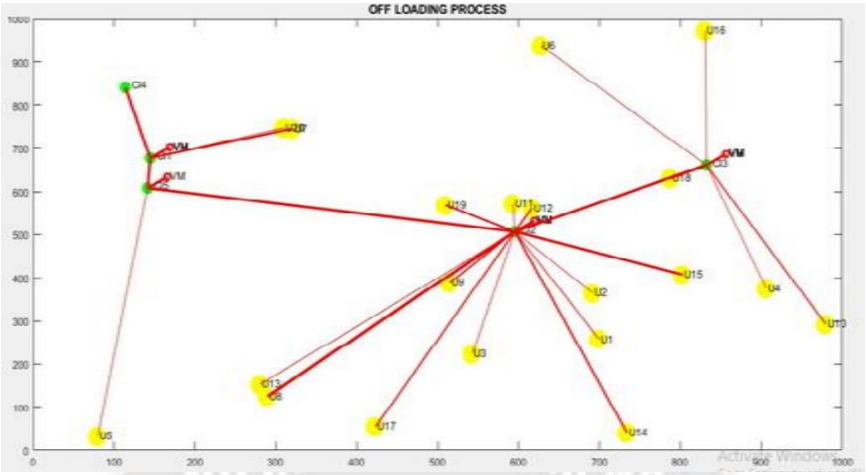

Figure 1: Offloading Process

Figure 1 shows the result of a basic distance calculation used to determine the distance between the user and the cloudlet. With these lines, we can see that users are moving their apps to cloudlets. The addition of the VM (Virtual Machines) is another component.If the current virtual machine is full or unavailable, the data is sent to other virtual machines. There is also the unloading between several cloudlets. Consequently, this method avoids deadlocks.

What follows is the HEACO algorithm.

Input: Process with m tasks $Task_i$
Output: Allocate all m number of tasks with balanced load

i. For (i=1, 2, 3, 4……..m) do
ii. Utilize BW_{rl} along with the time slot $slot_{rl}$ for every Execution_Unit;
iii. Determine $Node_{loc}$ with available idle time $T_{idealloc}$ for $Task_i$;
iv. Determine $Node_{min}$ with available idle time $T_{idealmin}$ for $Task_i$;
v. If ($Node_{min} == Node_{loc}$ $\| T_{idealloc} \leq T_{idealmin}$ Then
vi. Utilize $TC_{i.min} < TC_{i.loc}$,
vii. If ($BW_{i,min} \leq BW_{rl}$) then
viii. Assign $Task_i$ to the remote node.
ix. Assign $slot_{rl}$ of links on the paths from source node to the $Node_{min}$
x. Else
xi. Offload
xii. End if
xiii. End For

It is one of the most basic algorithms that verifies the process's execution unit's location and bandwidth. The execution unit receives it if both criteria are fulfilled. The job is still offloaded even after allocation if the execution unit is overcrowded or lacks the necessary bandwidth.

A suggested approach for compute offloading that is heterogeneous and energy efficient finds the shortest route from the mobile device to the cloudlet and then offloads the data. This algorithm is composed of two parts:

i. we use a basic distance calculation to determine how far away the cloudlet must be from the mobile device.
ii. Then, the offloading process begins, taking into consideration factors such as RAM used, number of manageable tasks, CPU utilisation, battery spent, and time left.

This HEACO Algorithm makes use of a number of factors, including the various nodes, tasks, and deployed bandwidth. The transfer procedure begins if the provided bandwidth falls short of the necessary bandwidth. To determine the shortest path between the cloudlets and the mobile users, the distance formula is considered. The HEACO Algorithm was built using this fundamental method. To offload resource-intensive mobile apps, this is the most basic energy-efficient strategy that makes use of a variety of characteristics.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
Article Received: *25 March 2023* ***Revised:*** *12 April 2023* ***Accepted:*** *30 May 2023*

RESULTS

Here we can see the different results that came out of using the HEACO.

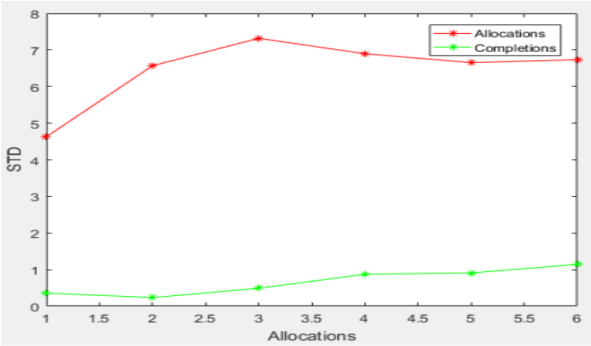

Figure 2: Standard deviation along with Allocations

As a measure of performance, standard deviation is used. Figure 2 shows the distribution vs. the standard deviation during allocations and completion. The red line represents a normal distribution of allocations; after a high allocation, there is a sharp spike. The route is almost straight (when completed), thus there isn't much of a variation. As a result, there will be very few exceptions. The total number of allocations in relation to the quantity of battery used and left may be shown in Figure 3. An application's total number of allocations is equal to the sum of all of its allocations.

There is a noticeable drain on battery life while the app is run. There is also an indication for the amount of battery life remaining. It is possible to see the curve below during unloading.

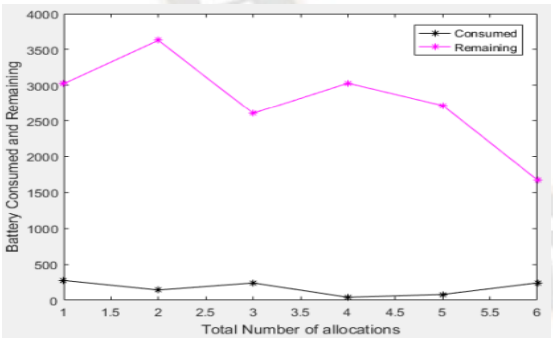

Figure 3: Battery consumed and remaining and Cloudlets

The graphic provides a visual representation of the battery usage and remaining power during unloading. During unloading, the battery once again displays a typical gradient. The shown graph indicates the presence of six cloudlets. Battery life is at its lowest at cloudlet 4, and it's almost dead at cloudlet 6, when the curve drops sharply. Evidently, there is a significant quantity of battery left compared to what has been spent. Therefore, the remaining battery was effectively used for offloading. When the HEACO algorithm is used, there is also no issue with battery scarcity. Figure 4 shows the relationship between the number of executions and their duration. In this case, four cloudlets are considered. The cloudlets are being shown the execution time. After a typical rise, the curve flattens out, and the execution time gradually increases once again. The execution count grows in direct proportion to the execution time. The figure depicts four cloudlets. The application is occupying cloudlet 1 at about 119 ms.

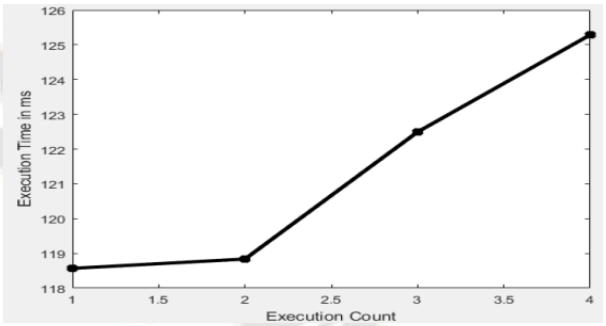

Figure 4: Execution Count along with the Execution Time

Figure 5 displays the results of six simulations that were run in the network. Due to the heterogeneous nature of the network and the fact that the processing units and virtual machines are initialised with different sets of attribute values each time, the simulation time is not consistent. The execution time ranges from 18 ms to 99 ms.

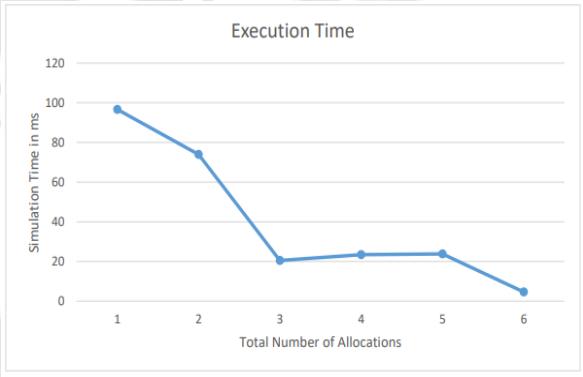

Figure 5: Slope of Total Number of Allocations and Simulation Time

Number of allocations as a function of standard deviation is shown in the graph below. The standard deviation is the variation in expectations that occurs when a processing unit is assigned a job or procedure. It is the value at which a certain percentage of a set's members deviate from the set's mean. If the programme is complete but the execution unit is overloaded, then the Standard Deviation measures the number of erroneous allocations. If the application is complete, then the Standard Deviation measures the number of completed tasks.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
***Article Received:** 25 March 2023 **Revised:** 12 April 2023 **Accepted:** 30 May 2023*

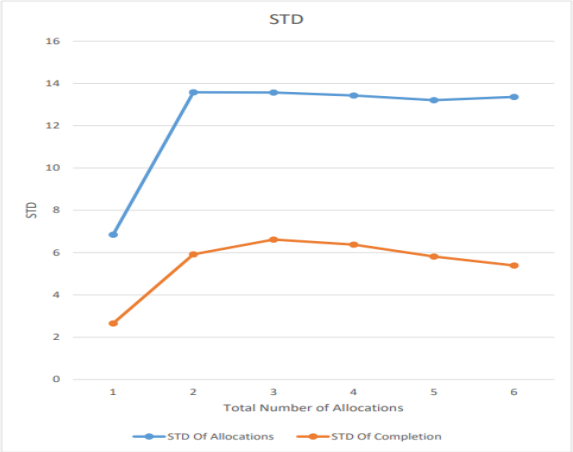

Figure 6: Total Number of Allocations along with Standard Deviation

With respect to the standard deviation, Figure 6 displays the overall number of allocations. The standard deviation parameter shows the extent to which the results differ from the initial values. Essentially, the HEACO Algorithm boils down to offloading the request if the cloudlet can't finish it. The search for the next closest cloudlet is initiated in such a circumstance. Users or mobile phones are represented by U in the picture below, while the available cloudlets are represented by C.

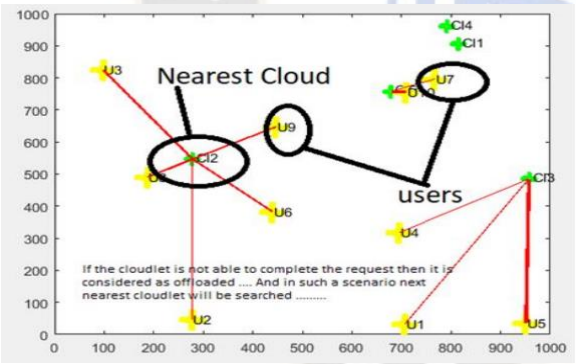

Figure 7: Offloading Process

Users, cloudlets, and the HEACO Algorithm's offloading process are shown in Figure 7.

GENERAL ISSUES AND CHALLENGES IN COMPUTATION OFFLOADING FOR MCC

All three viewpoints—the security viewpoint, the resource-intensive architecture of the current frameworks, and the ideal offloading platform—are used to highlight the chosen challenges.

Platform diversity

The wide variety of smartphone designs and operating systems is a problem for existing compute offloading frameworks. Here is an example that demonstrates this variety: You may use MAUI, an offloading framework, for the. As opposed to the.net framework, Mirror Server is an Android-compatible framework. Cloud computing services should be accessible to SMDs consistently, irrespective of the operating system or hardware in use. A difficult problem in the MCC area is the lack of a standardized offloading infrastructure for various smartphone platforms.

Security and privacy in mobile cloud applications

When processing applications in the cloud, data transfer security is of the utmost importance. Keeping sensitive information secure and private is of the utmost importance while unloading. Various perspectives may be used to approach these ideas: three places:(1) the mobile device itself, (2) the cloud, and(3) the network itself. Along with all the new technology, the number of complex assaults on mobile phones—the primary targets of cybercriminals—has been steadily rising. When it comes to the safety of cloud data centres, the main concerns revolve on the data transfer between the various network nodes. Cloud providers and mobile customers alike anticipate very high levels of security. There are persistent security risks in the existing frameworks that affect the binary transmission of application code during runtime. Strong safeguards and a safe environment are necessary for the three components of the MCC model, regardless of the solutions that are available.

The authors investigate safe offloading of relevant linear programming (LP) calculations with an eye towards optimising tasks and computations. Public LP solvers on the cloud and private LP parameters held by the user form the basis of this paper's work on offloading LP computing. In order to verify and produce efficient results, the authors zero in on the basic duality theorem of LP computing and formulate the necessary criteria for accurate results to satisfy. Secure data and compute outsourcing to an untrusted commodity cloud is presented by Bugiel et al. in their design. They suggest a design that uses a trusted cloud and a commodity cloud, which are really twins.

By dividing the calculations in this manner, important operations are primarily handled by the trusted cloud, while requests for offloaded data are done in parallel on encrypted data by the quick commodity cloud. Nevertheless, new challenges arose from the concept of segmenting processes and entrusting them to separate clouds. For example, it will be necessary to make obvious changes to the core infrastructure in order to install and maintain this cloud architecture. There is a growing security risk that is outpacing our ability to respond. To adapt to new developments and make the most of new services, security procedures must

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
Article Received: *25 March 2023* ***Revised:*** *12 April 2023* ***Accepted:*** *30 May 2023*

continuously improve and advance. Therefore, a security system that addresses every security risk simultaneously cannot be defined.

Fault-tolerance and continuous connectivity

The mobility of SMDs is a crucial feature in MCC. Reason being, two of the most important factors in determining whether consumers will be satisfied with mobile cloud services are the ability to move about freely and communicate independently while using them. On the other hand, getting constant connection and access to cloud services when travelling isn't possible due to a few limitations. Data transfer rates and available network capacity could fluctuate as mobile users move around. In addition, users could experience disconnections when transmitting or receiving data; hence, offloading methods should include appropriate fault-tolerant algorithms to resend the missing components, minimize reaction time, and lower mobile device energy usage. The assurance that offloaded programmes will run properly is of the utmost importance to mobile users.

Automatic mechanism

It is still necessary to automate the current computation offloading frameworks. By doing so, the unloading process may be carried out more smoothly as the surrounding area is explored. Automating this process is no simple feat; it requires the establishment of a protocol that is specifically designed to locate and identify services in light of the present environment and its limitations.

Offloading economy/cost

Users are responsible for paying the fees associated with using the resources of the cloud infrastructure in accordance with the Service Level Agreement (SLA) they have signed with the cloud provider. Users often wind up paying more for processes like data transmission between cloud providers and content dumping. Consequently, offloading choices should be based on economic considerations.

Partition offloading and external data input

Determining which application components should be offloaded and locating the appropriate server may be somewhat tricky during runtime. The time it takes for the offloaded parts of the programme to run could be impacted by the resource-intensive algorithms that attempt to solve this issue. Current application partitioning techniques do enable adaptive programme execution on both mobile devices and cloud servers, but they do not address the question of how to make the most of the elastic resources available in the cloud. This is really essential for creating the

Table 1 Some challenges and open issues in offloading frameworks for MCC.

	Open Issues in offloading frameworks	Challenges to available offloading frameworks
Access a distributed platform transparently	√	√
Continuous connectivity to cloud servers	–	√
Diversity of operating systems in mobile devices along with the variety of their architectures	√	√
Provide an effective execution of a process remotely and returns result to mobile device	√	√

Applications become scalable when they need to handle a high volume of mobile users and when they include data stored on other distant servers. Table 1 provides a concise summary of the key problems with existing offloading systems and unanswered research questions in MCC. The open issues describe difficulties that have not been handled in present offloading frameworks, while the challenges suggest problems that still need additional research and development in MCC's compute offloading frameworks.

CONCLUSIONS

The HEACO method takes into account heterogeneous characteristics such as execution time, battery usage, and standard deviation. This algorithm offers a practical way for mobile devices to save energy. Mobile devices are capable of doing basic arithmetic operations, but more intensive apps are moved to the cloud. The HEACO algorithm is a great way to transfer large, resource-intensive jobs from mobile devices to cloudlets, or even to other cloudlets if one is already full. To ensure user happiness, it offers application processing that is both efficient and light on energy consumption. Three viewpoints are used to present the chosen issues: the security viewpoint, the resource-intensive architecture of the current frameworks, and the ideal offloading platform.

REFERENCES

[1] A.Mtibaa, A. Fahim, K.A. Harras, M.H. Ammar, Towards resource sharing in mobile device clouds: power balancing across mobile devices, ACM SIGCOMM Comput. Commun. Rev. 43 (4) (2013) 51–56.

[2] Abusaimeh, Hesham. (2022). Computation Offloading for Mobile Cloud Computing Frameworks and Techniques. TEM Journal. 11. 1042-1046. 10.18421/TEM113-08.

[3] Akherfi, Khadija & Gerndt, Micheal & Harroud, Hamid. (2016). Mobile cloud computing for computation offloading: Issues and challenges. Applied Computing and Informatics. 14. 10.1016/j.aci.2016.11.002.

[4] Biswas, Milon & Whaiduzzaman, Md. (2019). Efficient Mobile Cloud Computing through Computation Offloading. 10.4172/0976-4860.1000225.

[5] D. Bernstein, Containers and cloud: from LXC to Docker to Kubernetes, in: IEEE Cloud Computing 1.3, IEEE, 2014, pp. 81–84.

[6] H. Qian, D. Andresen, Jade: reducing energy consumption of android app, Int. J. Network. Distrib. Comput (IJNDC) 3 (3) (2015) 150–158 (Atlantis Press).

[7] J. Hauswald, T. Manville, Q. Zheng, R. Dreslinski, C. Chakrabarti, T. Mudge, A hybrid approach to offloading mobile image classification, in: International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE, IEEE, 2014, pp. 8375–8379.

[8] M. Shiraz, A. Gani, R.H. Khokhar, R. Buyya, A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing, IEEE Commun. Surv. Tutorials 15 (3) (2013) 1294–1313.

[9] M. Shiraz, E. Ahmed, A. Gani, Q. Han, Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing, J. Supercomput. 67 (1) (2014) 84–103.

[10] M. Smit, M. Shtern, B. Simmons, M. Litoiu, Partitioning applications for hybrid and federated clouds, in: Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research, IBM Corp., 2012, pp. 27–41.

[11] M. Tulloch, Introducing Windows Azure for IT Professionals, Microsoft Press, 2013.

[12] Mehta, Shikha & Kaur, Parmeet. (2019). Efficient Computation Offloading in Mobile Cloud Computing with Nature-Inspired Algorithms. International Journal of Computational Intelligence and Applications. 18. 1950023. 10.1142/S1469026819500238.

[13] Myagila, Kasian. (2020). Mobile Cloud Computing as Mobile offloading Solution: Frameworks, Focus and Implementation Challenges. Computer Engineering and Intelligent Systems. 10.7176/CEIS/11-5-04.

[14] S. Mathew, Overview of Amazon Web Services, Amazon Whitepapers, 2014