PERFORMANCE ENHANCEMENT THROUGH COMMUNICATION OFFLOADING FOR ENERGY EFFICIENCY ON MOBILE CLOUD COMPUTATION

NOSHARWHAN ADIL, PRINCE WAQAS KHAN, YUNG-CHEOL BYUN

ABSTRACT: RECENTLY, THERE HAS BEEN AN ENORMOUS INCREASE IN MOBILE DATA USAGE WITH WIDESPREAD SMARTPHONE PROLIFERATION SIMILAR DEVICES AND THE GROWING POPULARITY OF VIDEO STREAMING SERVICES. CLOUD COMPUTING IS THE STRUCTURAL DESIGN IN WHICH VIRTUAL MACHINES, CLOUD SERVERS, HOSTS, AND TRADERS PARTICIPATE TO EXECUTE ANY JOB ON THE CLOUD. THE MIGRATION OF THE VIRTUAL MACHINE IS THE MAJOR PROBLEM THAT HAS BEEN EMPHASIZED DURING THIS SECTION. BECAUSE OF THE OVERHEAD OF VIRTUAL MACHINES, THE TASK'S EXECUTION TIME IS INCREASED. INFORMATION AND COMMUNICATION TECHNOLOGY HAVE EMERGED TREMENDOUSLY IN THE PAST FEW YEARS, MAINLY DUE TO THE INTRODUCTION OF SMARTPHONES. HOWEVER, LIKE ITS PREDECESSORS, THE NEW TECHNOLOGY CAME WITH ITS LIMITATIONS AS WELL. THE HANDHELD GADGETS WE CALL SMARTPHONES FACE SOME SEVERE CHALLENGES IN PERFORMANCE (COMPUTATION), STORAGE, AND ENERGY. FIRST, TWO CHALLENGES ARE SOMEHOW ELIMINATED BY THE INCREASE IN PROCESSING POWER AND IMPROVEMENT IN OPERATING SYSTEMS. ENERGY MANAGEMENT IS ONE OF THE MOST DEMANDING PROBLEMS IN SMARTPHONE. THIS RESEARCH AIMS TO TACKLE THE ISSUE BY USING THE CLOUD COMPUTING CONCEPT. THE PRIMARY FEATURE OF SMARTPHONE IS TO COMMUNICATE. THE LARGER THE COMMUNICATION IS, THE HIGHER WOULD BE THE ENERGY CONSUMPTION. IN THIS RESEARCH, WE PROPOSE A NOVEL APPROACH FOR OFFLOADING, AND THIS METHOD IS THE RIGHT SOLUTION TO RESOLVE THE ENERGY CONSUMPTION ISSUE FOR COMMUNICATION-INTENSIVE APPLICATIONS. TO DEMONSTRATE OUR PROPOSED METHOD'S EFFECTIVENESS, WE PERFORMED DIFFERENT ANALYSIS TESTS.

INDEX TERMS: COMMUNICATION OFFLOADING, CLOUD COMPUTATION, ENERGY EFFICIENCY, ENERGY MANAGEMENT, MOBILE DATA, SMARTPHONES, VIRTUAL MACHINES

---

1. INTRODUCTION

SMARTPHONES have grown to be a necessity for everyday life as they encourage people with luxurious applications and utility. Smartphone applications provide user to perform communication more effectively and efficiently as compared with other related devices. People use smartphones for daily activities ranging from entertainment to solving professional tasks. From their earlier predecessors, Feature Phones, the hardware of smartphone devices is incredibly advanced. This advancement in hardware has allowed the programmers to write complex applications. Another factor that has raised the usage of Smartphone devices is the progress of telecom industry. Fast internet speed with cost-effective packages is among the main reasons for the increased use of smartphone devices. With Smartphone devices, mobility is a significant utility when it can indeed be used anywhere. Due to small energy units, the idea of mobility is fascinating, but it is limited. The battery life of existing is not enough in many situations. The user either alter their behavior or limit their usage to preserve the battery. Different solutions are offered on different abstraction levels, from the hardware component low energy efficiency to the user level [1]. Nowadays, the demand from large retailers is increasing and providing a wide range of QoS to customers is a challenging task for network operators [2].

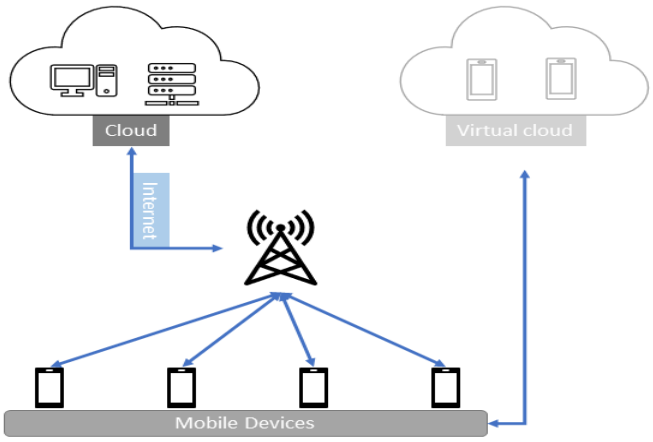

***Fig. 1**. Mobile Cloud Architecture.*

Cloud computing is a phrase that is mostly used to distribute facilities introduced in the cyberspace. Cloud computing is a kind of Internet-based technology that aims to deliver standard internet of things (IoT) assets and deliver customer information on necessity [3]. It is the only prototype that is global, and offers access to request that decreases cost, time, and needs with a smaller amount of work. IoT is gaining attention now a days because of its scalability and various applications [4]. In cloud computing, nearby exist information hubs where the material is kept for impending usage. Mobile Cloud computing is a skill that combines cloud computing, mobile computing, and wireless networks; therefore, cloud sources, network operators, and mobile users can use computing sources. Cloud environment is now applicable in many areas for better traffic management [5]. Mobile cloud computing is increasingly common on a day-to-day basis, and many companies accept these skills. These companies offer prospects to cloud sources as well as network operators [6].

- *Nosharwhan Adil is with the Department of Computer Engineering, Jeju National University, Jeju City 63243, Korea (e-mail: nosharwhan@ gmail.com).*
- *Prince Waqas Khan was with University of Agriculture Faisalabad, Pakistan. He is now with the Machine Learning Lab, Department of Computer Engineering, Jeju National University, Jeju City 63243, Korea (e-mail: princewaqas12@hotmail.com).*
- **Corresponding Author: Yung-Cheol Byun is working as a full professor at the Department of Computer Engineering, Jeju National University, and also serving as the director of IIST, Jeju City 63243, Korea (e-mail: ycb@jejunu.ac.kr).*

2. Literature Review

Cloud Computing remains a different method of giving computing properties and services. Cell phones are related to portable systems through resources to build stations that manage and create up the organizations and useful edges between systems and cell phone the promise of low latency connectivity and efficient bandwidth usage has led to the recent shift from cloud computing to edge computing [7]. Mobile consumer data and solicitations are transferred to the focal processor connected with servers giving multipurpose system administrations. Supporters ask for are transported to a cloud through the web. In the cloud, cloud controllers manage the solicitations toward furnish portable clients with the concerning cloud benefit [8]. Figure 2 shows the different layers of cloud computing platform.

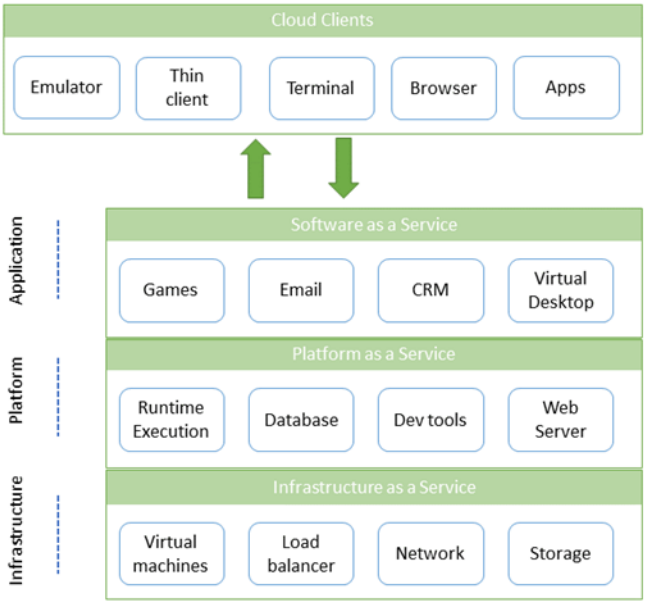

***Fig. 2**. Cloud Computing Layers.*

Mobile Cloud Computing insinuates a structure where data are accumulating and data taking care of occurs outside the mobile phone. Versatile Cloud applications Transfer figuring power and data gathering from phones to extreme, united handling stages arranged in a virtual space, such as the cloud, open by methods for a distant relationship from the perspective of a lightweight neighborhood client. Cell phones confront challenges these days, such as battery life and data transfer capacity, and distributed computing wipe out the above issues. Distributed computing enables the client to utilize foundation, stages, and programming through ease, on-request cloud suppliers. All the escalated asset computation can be kept running in the cloud rather than a versatile outline. MCC is an exceptionally encouraging pattern for the eventual fate of versatile registering [9]. The communication offloading of cell phones to the computational cloud framework depends on the assurance of strategies to be run remotely, with the goal that the advantages can be accomplished regarding time and asset utilization, which spares vitality. In writing, we have discovered a wide variety of propositions with administrators who choose powerfully, whether part of the application will be run locally or remotely. These propositions likewise utilize close-by PCs, additionally called surrogates [10]. These surrogates are asset-rich PCs associated with the Internet and are accessible for use with cell phones in the region without deferrals and WAN unsteadiness. Their objective is like that of intermediary servers. It was found in a money-saving advantage investigation report, which concentrated on the vitality sparing method of emptying the estimation. Figure 3 shows the overview of offloading process.

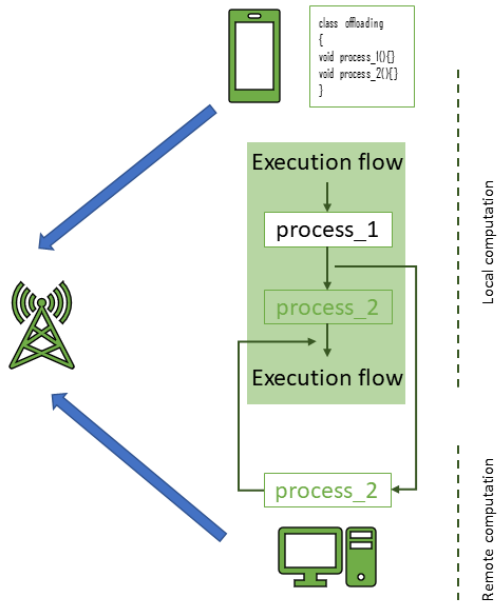

***Fig. 3**. Offloading Process Overview.*

3. Frameworks of Offloading In Mobile Cloud Computing

Although there are a few offloading instruments accessible for offloading serious calculation parts of portable applications to the cloud, we can characterize these components into two general classes: First is systems considering virtual machine cloning. The other is the systems considering code offloading. This area presents distinctive current offloading structures. For every one of the designs, we distinguish the methodologies utilized as a part of the last segment's three stages. Toward the area's finish, the diverse systems will be contrasted with deference with their most essential properties.

3.1. Clone Cloud

Clone Cloud structure enhances cordless life and execution on the cell phone by divesting concentrated segments to mist servers. The parceling advance in this system joins static program investigation with program profiling to create an arrangement of off loadable parts while meeting a few requirements, such as techniques that versatile utilization sensors ought to be performed locally. The structure utilizes string point granularity for the dividing of utilizations. The static examination is to find imperatives on conceivable movement indicates while profiling points fabricate a fee demonstrate used for offloading and implementation. Parceling and joining of the requests are done at the application level [11]. The impartial of the circled implementation framework in Clone Cloud was to realize a section of an agreed claim development performing confidential an application-layer computer-generated mechanism. In the Clone Cloud structure presented in Figure 4, the scattered execution encounters a couple of stages as takes after. Exactly when the customer tries to run an allocated, the structure appearances in a file of pre-figured sections for existing implementation conditions (for instance, open framework exchange speed and cloud

resources). The eventual outcome of the affirmation is a section setup report [12].

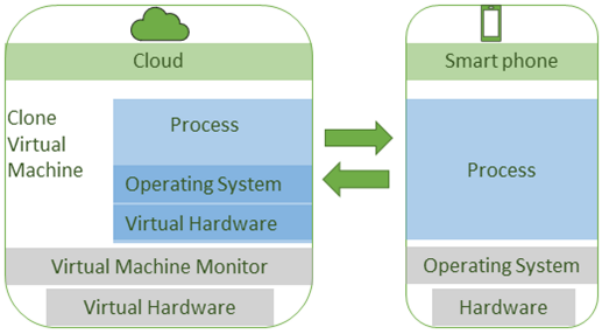

Fig. *4. Clone Cloud Execution Model.*

The fragment is stacked through the claim twofold, which gadgets the picked techniques with development. Happening the moveable telephone, when the framework's execution achieves an improvement point, the continuous string is deferred. Its public is secured and passed on towards a relating clone. In this clone, the string country stands launched fixated by another series with a relative stack and reachable burden things, and sometime later continued. On the cloud worker, when the moved string achieves a re-joining truth, it is delayed, bundled, and after that, dispatched back to the cell phone. Decisively, the got bundled series is joined into the status of the essential framework in the remote. To assess the model, the producers executed three applications sickness scanners, picture pursue, and security protecting focused on publicizing. Internet security plays an important role in protecting people who use the Internet through a variety of electronic devices in everyday life [13]. All assessments are regular of five executions. Telephone, Clone Cloud with Wi-Fi, and Clone Cloud with 3G are the pre-owned conditions. A conspicuous propensity is that more major remaining tasks at hand advantage from offloading, considering the improvement cost's compensation over a more noteworthy computation.

3.2. MAUI

MAUI, separating is done in the context of organizer elucidations to show which sections can be performed at all and which can't throughout the activity step; two supplies must be met. The first is to ask for sets essential to remain in both adaptable and server borders. The second is to go-betweens, profilers then solvers obligation be exhibited on together the cell phone and server sides. Toward the starting, MAUI profiler procedures the gadget features. By that point, it continues watching the program and structure qualities amidst the entire execution time in light of how these attributes would much be able to of the time modification and some old or off Centre estimation may lead MAUI to settle up on the wrong choice. The offloading option is taken on at runtime. The structure picks which pieces ought to be remotely accomplished by the selection of the MAUI solver. The choice depends upon the responsibility of the MAUI profiler. Figure 5 demonstrates MAUI design. On the PDA, the structure contains the running with areas: a delegate, a profiler, and a solver. Each time a technique is described, the High-level view of MAUI's architecture [14]. MAUI profiler assesses him intended used for its imperativeness saving possible and shapes the mechanism and the framework to get the rank info. By then, the MAUI solver wears down the fallouts gave with the profiler and chooses the objective where the procedure determination be the least bit implemented; the mediator is accountable for regulator and statistics trade between the server and the mobile phone. Happening the waiter side, the profiler and the server delegate perform equivalent parts as their customer side accomplices.

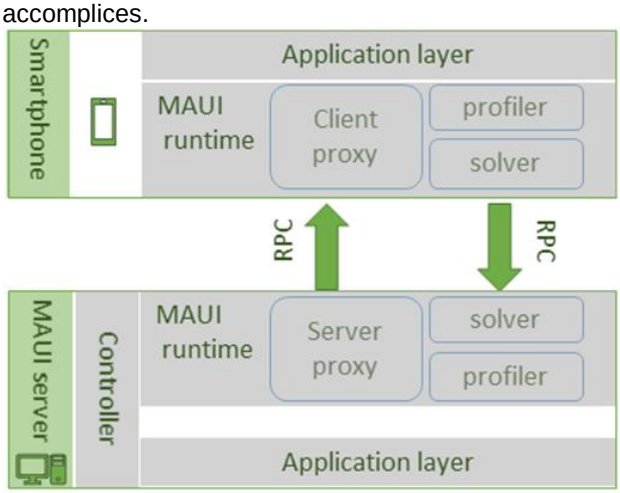

Fig. 5*. MUAI architecture.*

3.3. CLOUDLET

Offloading to the cloud isn't, for the most part, an answer, since of the Tall WAN dormancy, basically for claims with progressing confinements. In this way, the cloud must be moved closer to the flexible customer as cloudlets. Propose in a Virtual Machine founded cloudlet Agenda. A cloudlet canister is portrayed as an encouraging space for offloaded errands passed on to distant capitals, as dissimilar as discrete servers or parallel structures. Cloudlets are virtual machines (VM) from the perspective of assistance versatility, adaptability, and adaptability [15]. Cloudlets are broadly disseminated Internet foundation segments whose capacity assets and registering cycles can be misused by close-by cell phones while maintaining a strategic distance from the long dormancy, accessible for getting too far off cloud assets. These cloudlets would be arranged in like manner territories, for example, coffeehouses, with the goal that cell phones can interface and work as a thin customer. We can take a circumstance where the customer must implement an estimation raised application. By runtime, the request realizes a neighboring cloudlet along with divests the assessment heightened compact application. In any case, the convenient application can find a substitute cloudlet and track the request in a rapid because of the deficiency of framework arrange a time [16].

3.4. JADE

Having a similar concern yet from a substitute perspective is shown in a structure that monitors usage and contraption significance, and that typically picks where the code would be implemented. The objective of Jade remained to help the upsides of essentialness careful count offloading for flexible applications while restricting the weight on specialists to gather such an application [17]. The mobile phone that offloads count is known as the client. The gadget that implements the offloaded code is known as the server. Versatile applications hold small errands that can be offloaded to the server. The Jade runtime train like this pick was to perform small endeavors and begins passed on execution.

- Profiling: The structure ought to have refreshed data about the request and the gadget's rank. Application profiling is the way to gather data about projects, such as vitality utilization, information estimate, execution time, and memory use. Likewise, gadget profiling collects data about gadgets status, such as battery level, CPU use, and remote association.
- Message: The framework ought to have the capacity to associate with the other servers, arrange by the remote server designed for rid segments, and also the direct information among the cell phone and the remote server. It also takes after ranking of distant performance; Spare data identified with every single associated gadget (e.g., association speed, equipment setup).
- Optimization: The structure decides an upgraded offloading way to decrease vitality utilization and improve its execution.
- Consequently, the Jade structure changes request implementation, scheduled one cell phone into an isolated accomplishment advanced for the remote association, control utilization, and server abilities. With a specific end goal to check the vitality measure that Jade can put something aside for cell phones, creators confront location applications on cell phones. The application performs encounter identification on 50 cinemas through the scope of a piece under 200 KB. Outcomes appeared that Jade lessens the energy utilization for confront identification applications.

3.5. MIRROR SERVER

The mirror server structure that practices Telecommunication Service Provider (TSP) founded isolated organizations. A TSP is a correspondence pro association that gives voice correspondence organizations, for instance, landline phone utilities. Mirror server grows mobile phones' limits by giving three one of a kind sorts of organizations: count offloading, security, and limit. Mirror server is a historic waiter which holds VM formats for various Mobile contraption stages. This structure does not need a dividing the whole application is offloaded. In the course of the action step, another VM event is made. This VM is a christened adjustable mirror then the mirror server manages to direct and pass on the compact mirrors on an enrolling system in the telecom arrange. Applications are executed in the mirror VM events, and consequences stand refunded to the SMD. The schedule uses a streamlined part to offload [17].In the Mirror Server planning, on the SMD sidelong, a client association part (Sync-Client) is sent inside the SMD working structure (OS) to amasses fuse insights other than imparting to the glassworker for the association. On the worker side, with a specific real goal to hold mirrors and PDAs synchronized, the synchronization module (Sync - Server) animates reflects as exhibited by the information gave by Sync-Client and the Traffic Monitor module, which screens organize advancement between the cell phone and the IP orchestrate. The rule major issue is that reflect workers are not proposed for information preparing and by ideals of that single constrained associations (for example, record taking care of, report filtering) can be given stood apart from the gathering of associations in cloud worker farms. In the proposed structure, antivirus scanner application can be sent as an association on the mirror worker, and the application can get to the record framework on mirrors. The benefits of move the antivirus scanner to the glasses are gigantic. It ensures battery control on telephones. Filtering won't sway PDA designs' everyday livelihoods in the interim. The microchip and I/O certified tasks are enthused to the mirror worker. Running the mirror's expansiveness will be through and through speedier than that on the versatile because of its constrained assets [19].

3.6. CUCKOO

Cuckoo misuses the present activity show in Android, which kinds the division among the heightened and non-concentrated fragments. This development offers a graphical UI [20]. Offloading choice at run time and offloads the straightforward a lot of depicted sections of the application. On the off chance that the far-off assets are not available (for example, create association isn't open) by then, the application can be executed on contiguous assets (the telephone).

3.7. PHONE2CLOUD

Calculation offloading system called Phone2Cloud. The goal was to enhance the vitality productivity of cell phones and enhance the application's execution. Not at all like the past structures, creators Centre around directing an entirely quantitative investigation on vitality sparing of the framework by leading application examinations and situation tests. The cut off is portrayed in the light of estimates to concede moves with a real objective to offload more data on Wi-Fi while regarding the application's obstruction edge. The framework will sit tight for Wi-Fi (just if 4G stores are typical inside application's delay opposition) to wind up openhand an offloading middle person that associations the offloading decision engine to the remote execution director. The decision engine is worked remembering the real objective to dismember the power use because of offloading. Previously execution, two sorts of connections are made. First is the typical implementation time of the application running on the SMD is differentiated. The customer's deferral Tolerance edge and second is the power used to run the application on the SMD discovered and appeared differently concerning control usage required to run a similar application on the cloud [21]. In the first place, ordinary execution time and client's deferral quality limit are considered on the off chance that the client's deferral restriction limit is humbler than regular execution time before the application is off-stacked the cloud. Other than what's expected, the choice of motor models to run the application on the cloud is more critical than control use to run the application on the SMD. If it is the condition, the application is executed locally. Something interesting, the application is offloaded to the cloud. A format of the arrangement of Phone2Cloud is given in Figure 6. Phone2Cloud is made from seven fundamental Mechanisms.

- Implementation time pointer: It is extraordinary of the significant areas in Phone2Cloud. It is a scrounge deal to imagine the regular execution season of a whole application on a cell phone.
- Bandwidth screen and asset screen: Bandwidth screen is utilized to screen current transmission limit usage of the system while the asset screen oversees checking CPU outstanding burden and assorted assets. The two screens serve the offloading choice motor and the execution time Analyst self-governing.
- Offloading choice motor: It is within a bit of Phone2Cloud. Offloading choice motor picks whether to offload the application's parts from the PDA to the cloud worker. While it runs the application locally, it gathers neighborhood execution manager to actualize the application. Else, it

brings the offloading center individually to offload assessment to the cloud.

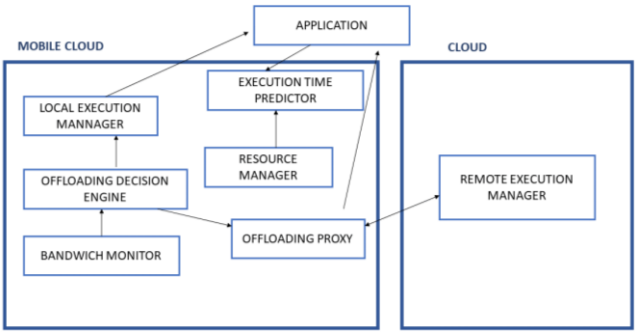

Fig. 6*. Architecture of Phone2Cloud.*

The creators look at how the vitality utilization and execution time of uses will be influenced. The assessed applications are word tally, way discoverer, and sort application. The structure spares vitality and enhances applications' execution and clients' understanding of cell phones. For example, confront discoverer applications costs more energy on cell phones than on a cloud server, and the contrast between the two expenses gets greater as info develops. The purpose of this is because information transmission expenses less vitality than running the application locally. Additionally, the vitality utilization in cell phones becomes quicker than that on the cloud server. In this way, confront discoverer applications ought to be offloaded to the cloud

.

4. Proposed System Architecture

Smartphone applications provide user to perform communication more effectively and efficiently as compared with other related devices. People use smartphones for daily activities ranging from entertainment to solving professional tasks. From their earlier predecessors, Feature Phones, the hardware of smartphone devices is incredibly advanced. This advancement in hardware has allowed the programmers to write complex applications. Another factor that has raised the usage of Smartphone devices is the progress is the Telecom industry [22]. Fast internet speed with cost-effective packages is among the main reasons for the increased usage of Smartphone devices. With Smartphone devices, mobility is a significant utility when it can indeed be used anywhere. Due to small energy units, the idea of mobility is fascinating, but it is limited. The battery life of existing is not enough in many situations. In the proposed research, the framework has been presented to offload communication from smartphones to the cloud. Offloading gives ideas to the developer for writing applications in such a way that they can utilize resources. The proposed framework will let the developers create functionality in the cloud for communication offloading. Once a mobile application is developed for any Smartphone operating system, the application will be uploaded to their corresponding Application Stores. For all further communication, the respective Application Store is responsible for all the resources needs to perform communication. Applications will send the communication to one component in Clone Cloud, which will send it to the relevant smartphone. Clone Cloud component will send communication after verification through Push Notification Technology for optimal Bandwidth utilization. Before proposing the solution, we have examined the mobile applications in detail and surveyed them at a University to find the applications mostly used by students. We have selected students for analysis because they use mobile applications ranging from Games, Communication-related apps, books, and other educational apps. Moreover, they spend fair enough time and have enough technical shrewdness to draw some useful analysis from the survey. The study's purpose was to mainly ask from the students about the applications they are widely using, time spent on each application, mobile set information, and how often they charge their mobiles. From the survey results, we concluded that regardless of gender, either the most used applications were related to social media or communication-related applications such as WhatsApp, Viber, etc. A fair percentage of these students more than 3-4 hours a day using these applications. Gaming apps were also used by most of the students. Also, on average, a university student needs to charge their mobile more than once throughout a single day. The survey results show the most used applications by the students of a selected University. The top three applications are somehow, directly or indirectly, connected to communication. Keeping in mind that the default Messaging application is not included, as it is not dependent on the internet and solely depends on Mobile Operators. We have also analyzed the number of hours an average student spends on the above mobile applications from the survey. It is noticed that a fair bit of time is again used by applications designed for communication purposes. Table 1 shows the number of hours an average student spends on the above-mentioned Mobile Applications.

TABLE I

Average daily usage on each App.

Sr #	Application Name	Average daily usage (Hours)
1	Facebook	4.16
2	WhatsApp	3.88
3	Skype	2.57
4	candy crush	1.35
5	Instagram	0.62

Also, from the survey, it was noticed that an average student recharge their mobile devices more than twice a day. Moreover, the percentage of different smartphone devices are shown in Table 2.

TABLE 2

Smartphone's Operating System.

Sr #	Operating System	%age
1	Android	83%
2	iOS	11%
3	Windows	6%

Considering the above results, we noticed that most users for Smartphone devices are Android users. We chose the Android platform to build our sample application, but the solution we will discuss in this document is not limited to any Operating System or any programming language. The answer is generic and can

be implemented on any other platform. The survey also proved that the most used applications are somehow related to communication. Unlike other solutions where offloading was done on computation, we propose a solution to offload transmission over to the cloud. The reason is quite evident from the survey that most applications in use are related to communication, and this is an area that needs offloading to save the device resources.

5. Communication Offloading

The proposed model consists of a cloud-based architecture to offload communication. There will be a method for all the applications, a clone of correspondence that will be determined to cloud for each application freely. Computation offloading proved to be helpful in improving energy efficiency [23]. At the point when mail is sent from application to the cloning procedure on the cloud, the clone technique will propel that correspondence to the organization on cloud condition. There are two completions of that organization. One lives on the cloud, and others will be running on a mobile phone. The cloud organization will welcome the data from clone procedures on the cloud and subsequently push the data to the crowd organization to needed wireless. The organization on the cloud is similarly careful to recognize the correct correspondence to address mobile phone. The crowd organization on the phone will perceive the proper application and forward the data to the application. There will be help running in the compact as opposed to each application running on it's on the organization. Figure 7 shows the overview of application architecture. Algorithm 1 explain the Pseudo-code statements for offloading.

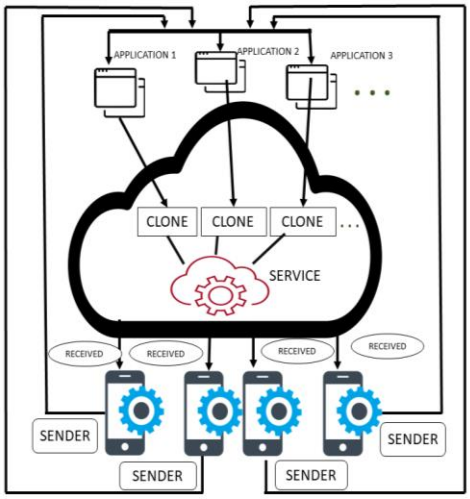

***Fig 7**. Application Architecture.*

To send correspondence from adaptable using any application, there will be no convincing motivation to incorporate cloud as this will cause an overhead. The applications inconvenient can talk about directly with the live applications concerning sending data. The live application will make the data object ought to have been sent and forward it to its clone on the cloud by encoding it with the Mobile Device ID that is affirmed to get the data. The clone will incorporate the application ID and send it to the organization over the cloud that will push the thing to the ideal device reliant on the Device ID. At the point when the item is jumped on the wireless, the organization will make the data to the right application using the Application ID; ultimately, the data will be unscrambled using the current phone ID. In this circumstance, every compact has its Unique ID that will be used to scramble and unscramble data for security. A unique application ID will be given to each application to ensure the data is sent to the correct application. Figure 8 shows the workflow diagrams to represent the sending and receiving of communication.

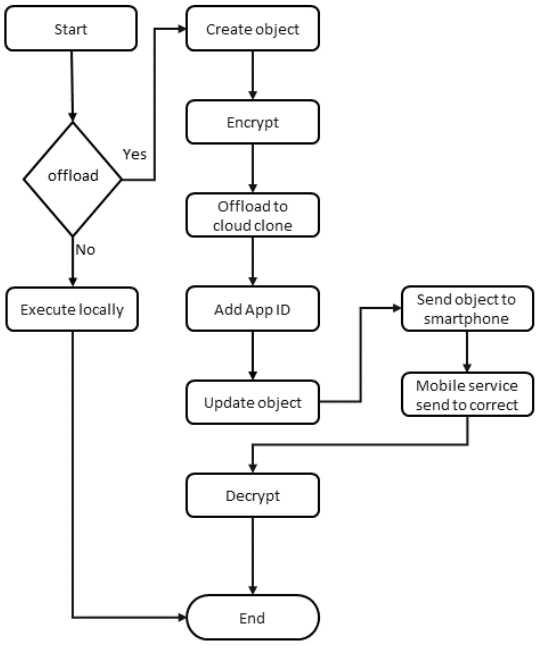

***Fig 8.** Workflow Diagrams.*

5.1. Algorithm 1 Pseudo-code statements.

```
Step 1: Application provider
      initialize (qo)
      encrypt ()
      transfer to cloud ()
Step 2: Cloud
      start services (qo)
        receive (qo)
      app ID with q_o
       transfer to app
   Step 3: initialize services
      IF {cloud is valid}
         update ()
      ELSE
         STATE return error message ()
      ENDIF
```

6. Test Results

For evaluation purposes, we have created four similar Applications in the Android Platform. The primary purpose of these applications is just communication, so, functionality-wise, one application has four instances. The reason for selecting an Android platform is that it is an Open source, and most smartphones used today are on Android OS. Simultaneously, there is no limitation in this proposal that only an Android device/OS is compatible with this solution. For the evaluation purpose, we have used the device described in Table 3:

TABLE 3

Devices used in evaluation.

Sr #	OS	Battery
Google Pixel	7.1	2770 MAH
Huawei mate 9	7.0	4000 MAH

Samsung galaxy S9	8.0	3000 MAH
Sony Xperia Z2	4.4.2	3200 MAH
Galaxy S7	6.0	3000 MAH

These Smartphone devices are chosen because of availability. These devices are tested using Wi-Fi and 3G (same operator). To evaluate, we have also created four Mobile Applications with the same code base, but the communication is done conventionally i.e., without using Cloud Model. The reason to do this is to identify the usage of resources of these four applications against the four applications that use the Cloud Model.

6.1. Battery Test

Battery Used in Apps without Cloud Implementation: The Battery was Charged 100% at the begging of the Trial. We used only these four with regular processes running in the background. The findings of utilizing these applications continuously for 2 hours are listed in Table 4. Table 5 shows the battery used in apps with cloud implementation.

TABLE 4

BATTERY USED IN APPS BEFORE CLOUD IMPLEMENTATION.

Device Name	OS	Battery	Test Start %age	Test End %age
Google Pixel	7.1	2770 mAh	100	68
Huawei mate 9	7.0	4000 mAh	100	86
Samsung galaxy S9	8.0	3000 mAh	100	71
Sony Xperia Z2	4.4.2	3200 mAh	100	83
Galaxy S7	6.0	3000 mAh	100	75

TABLE 5

BATTERY USED IN APPS WITH CLOUD IMPLEMENTATION.

Device Name	OS	Battery	Test Start %age	Test End %age
Google Pixel	7.1	2770 mAh	100	71
Huawei mate 9	7.0	4000 mAh	100	89
Samsung galaxy S9	8.0	3000 mAh	100	73
Sony Xperia Z2	4.4.2	3200 mAh	100	88
Galaxy S7	6.0	3000 mAh	100	81

6.2. Network Usage Test

Network Usage in Apps without Implementation: For this test, we used the Apps on Wi-Fi and 3G. WiFi is one of the best options used for overcoming the traffic demand during offloading [24]. We sent the same number data on both Wi-Fi/3G and Apps with and without cloud implementation. Table 6 shows the network usage in apps without cloud implementation and Table 7 shows the network usage in apps with cloud implementation.

TABLE 6

NETWORK USAGE IN APPS WITHOUT CLOUD IMPLEMENTATION.

Device Name	OS	Battery	WIFI	3G
Google Pixel	7.1	2770 mAh	1000 KB	998 KB
Huawei mate 9	7.0	4000 mAh	990 KB	1008 KB
Samsung galaxy S9	8.0	3000 mAh	989 KB	992 KB
Sony Xperia Z2	4.4.2	3200 mAh	999 KB	999 KB

TABLE 7

NETWORK USAGE IN APPS WITH CLOUD IMPLEMENTATION

Device Name	OS	Battery	WIFI	3G
Google Pixel	7.1	2770 mAh	860KB	890 KB
Huawei mate 9	7.0	4000 mAh	858 KB	901 KB
Samsung galaxy S9	8.0	3000 mAh	862 KB	888 KB
Sony Xperia Z2	4.4.2	3200 mAh	866 KB	892 KB

6.3. Performance Test

Performance Usage in Apps without Implementation: For Performance, we analyzed both with cloud and without cloud using Wi-Fi and 3G. For performance monitoring, we have used a free 3rd part tool from the Android marketplace i.e., OS Monitor by EOLWRAL. Moreover, we have listed the highest value of Performance monitor during the test. Table 8 shows the performance monitoring in apps before cloud implementation and Table 9 shows the performance monitoring in apps after cloud implementation.

TABLE 8

PERFORMANCE MONITORING IN APPS BEFORE CLOUD IMPLEMENTATION.

Device Name	OS	Battery	WIFI
Google Pixel	7.1	App1: CPU Usage11.6 App2: CPU Usage11.6 App3: CPU Usage11.6 App4: CPU Usage11.6	App1: CPU Usage12.1 App2: CPU Usage12.1 App3: CPU Usage12.1 App4: CPU Usage12.1
Huawei mate 9	7.0	App1: CPU Usage9.7 App2: CPU Usage9.7 App3: CPU Usage9.7 App4: CPU Usage9.7	App1: CPU Usage10.0 App2: CPU Usage10.0 App3: CPU Usage10.0 App4: CPU Usage10.0
Samsung galaxy S9	8.0	App1: CPU Usage13.3 App2: CPU Usage13.3 App3: CPU Usage13.3 App4: CPU Usage13.3	App1: CPU Usage13.9 App2: CPU Usage13.9 App3: CPU Usage13.9 App4: CPU Usage13.9
Sony Xperia Z2	4.4.2	App1: CPU Usage8.6 App2: CPU Usage8.6 App3: CPU Usage8.6 App4: CPU Usage8.6	App1: CPU Usage9.4 App2: CPU Usage9.4 App3: CPU Usage9.4 App4: CPU Usage9.4
Galaxy S7	6.0	App1: CPU Usage10.3 App2: CPU Usage10.3 App3: CPU Usage10.3 App4: CPU Usage10.3	App1: CPU Usage10.7 App2: CPU Usage10.7 App3: CPU Usage10.7 App4: CPU Usage10.7

TABLE 9
PERFORMANCE MONITORING IN APPS AFTER CLOUD IMPLEMENTATION

Device Name	OS	Battery	WIFI
Google Pixel	7.1	Cloud Service: CPU Usage 6.9 App1: CPU Usage 5.3 App2: CPU Usage 5.3 App3: CPU Usage 5.3 App4: CPU Usage 5.3	]Cloud Service: CPU Usage 7.5 App1: CPU Usage 5.5 App2: CPU Usage 5.5 App3: CPU Usage 5.5 App4: CPU Usage 5.5
Huawei mate 9	7.0	Cloud Service: CPU Usage 5.2 App1: CPU Usage 4.4 App2: CPU Usage 4.4 App3: CPU Usage 4.4 App4: CPU Usage 4.4	Cloud Service: CPU Usage 5.7 App1: CPU Usage 4.6 App2: CPU Usage 4.6 App3: CPU Usage 4.6 App4: CPU Usage 4.6
Samsung galaxy S9	8.0	Cloud Service: CPU Usage 7.3 App1: CPU Usage 6.9 App2: CPU Usage 6.9 App3: CPU Usage 6.9 App4: CPU Usage 6.9	Cloud Service: CPU Usage 7.7 App1: CPU Usage 7.1 App2: CPU Usage 7.1 App3: CPU Usage 7.1 App4: CPU Usage 7.1
Sony Xperia Z2	4.4.2	Cloud Service: CPU Usage 4.7 App1: CPU Usage 3.9 App2: CPU Usage 3.9 App3: CPU Usage 3.9 App4: CPU Usage 3.9	Cloud Service: CPU Usage 4.9 App1: CPU Usage 4.3 App2: CPU Usage 4.3 App3: CPU Usage 4.3 App4: CPU Usage 4.3
Galaxy S7	6.0	Cloud Service: CPU Usage 6.0 App1: CPU Usage 5.8 App2: CPU Usage 5.8 App3: CPU Usage 5.8 App4: CPU Usage 5.8	Cloud Service: CPU Usage 6.5 App1: CPU Usage 6.2 App2: CPU Usage 6.2 App3: CPU Usage 6.2 App4: CPU Usage 6.2

6.4. COMPARISON

As shown in Figure 9, battery usage drops when we move to the cloud model, as described in the thesis. On Average, with the entire device that we have tested, we found that about 19% of the battery is saved within two hours of testing. Figure 10 represents the distribution of CPU usage with cloud environment.

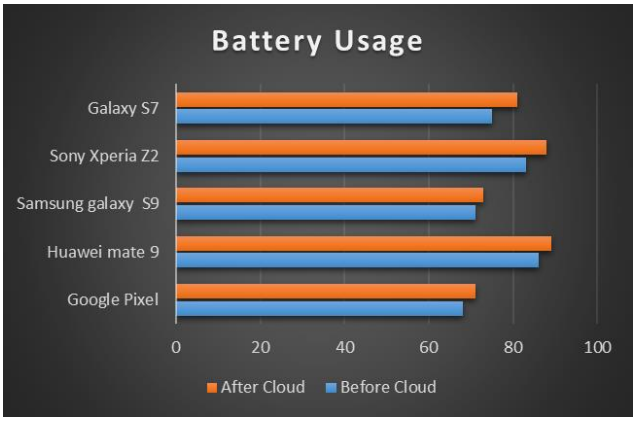

Fig 9. *Battery usage.*

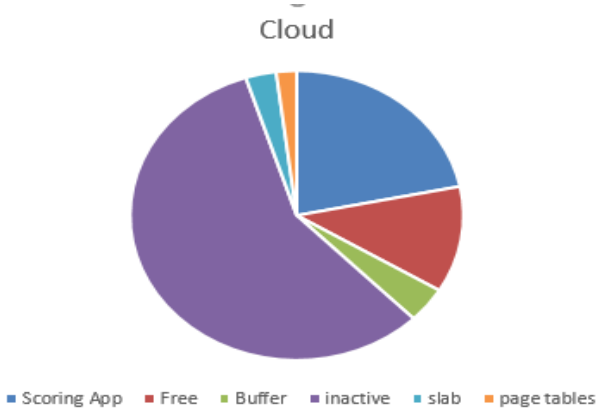

Fig 10*. CPU Usage – With Cloud.*

7. CONCLUSION

Communication offloading can help save smartphone resources. Generally, an average mobile user uses five to six mobile applications that can take benefit from this model. The more applications using our proposed model will devote more resources to a smartphone device. There is a slight high usage of the processor using this model, but we used all applications simultaneously during testing, which is an infrequent scenario in the real world. Existing offloading choice calculations would now be able to profit by our outcomes and utilize this examining approach as a supplement to think about obscure administrations in their choices. We additionally plan to understand a nonexclusive open-source structure that can offload discretionary asset serious assignments in a brilliant and utterly programmed path by considering our testing approach for obscure figuring administrations. We examine the calculation offloading emotional issues in a multi-cell versatile edge registering situation. We propose a universal consecutive offloading game methodology and plan a multi-client calculation offloading calculation where the portable clients successively settle on offloading choices dependent on the present obstruction condition and accessible calculation assets. Numerical outcomes show that our proposed analysis can accomplish proficient execution and scale well as the system framework size increments. We will examine the joint transmission power control and computational assets portion to calculate offloading dynamics in future work issues.

ACKNOWLEDGMENT

This work was supported by the 2020 education, research and student guidance grant funded by Jeju National University.

REFERENCES

[1] Wu, H., Sun, Y., & Wolter, K. (2018). Energy-Efficient Decision Making for Mobile Cloud Offloading. IEEE Transactions on Cloud Computing.

[2] Abbas, K., Afaq, M., Ahmed Khan, T., Rafiq, A., Iqbal, J., Ul Islam, I., & Song, W. C. (2020). An efficient SDN-based LTE-WiFi spectrum aggregation system for heterogeneous 5G networks. Transactions on Emerging Telecommunications Technologies, e3943.

[3] Almohanna, A. A., Win, K. T., & Meedya, S. (2020). Effectiveness of Internet-Based Electronic Technology Interventions on Breastfeeding Outcomes: Systematic Review. Journal of Medical Internet Research, 22(5), e17361.

[4] Khayyat, M., Elgendy, I. A., Muthanna, A., Alshahrani, A., Alharbi, S., & Koucheryavy, A. (2020). Advanced Deep Learning-based Computational Offloading for Multilevel Vehicular Edge-Cloud Computing Networks. IEEE Access.
[5] Khan, P. W., Byun, Y. C., & Park, N. (2020). IoT-Blockchain Enabled Optimized Provenance System for Food Industry 4.0 Using Advanced Deep Learning. Sensors, 20(10), 2990.
[6] Dillon, T., Wu, C., & Chang, E. (2010, April). Cloud computing: issues and challenges. In 2010 24th IEEE international conference on advanced information networking and applications (pp. 27-33). Ieee.
[7] Zakharov, M., Kirichek, R., Khan, P. W., Muthanna, A., & Koucheryavy, A. Analysis of Traffic Generated During Molecular Analysis Based on The Internet of Things. International Journal of Advanced Science and Technology.
[8] Noor, T. H., Zeadally, S., Alfazi, A., & Sheng, Q. Z. (2018). Mobile cloud computing: Challenges and future research directions. Journal of Network and Computer Applications, 115, 70-85.
[9] Kemp, R., Palmer, N., Kielmann, T., & Bal, H. (2011, October). Energy efficient information monitoring applications on smartphones through communication offloading. In International Conference on Mobile Computing, Applications, and Services (pp. 60-79). Springer, Berlin, Heidelberg.
[10] Vallina-Rodriguez, N., & Crowcroft, J. (2012). Energy management techniques in modern mobile handsets. IEEE Communications Surveys & Tutorials, 15(1), 179-198.
[11] Francis, T., & Madhiajagan, M. (2017). A Comparison of Cloud Execution Mechanisms: Fog, Edge and Clone Cloud Computing. Proceeding of the Electrical Engineering Computer Science and Informatics, 4(1), 446-450.
[12] Chun, B. G., Ihm, S., Maniatis, P., Naik, M., & Patti, A. (2011, April). Clonecloud: elastic execution between mobile device and cloud. In Proceedings of the sixth conference on Computer systems (pp. 301-314).
[13] Kashif, M., Malik, S. A., Abdullah, M. T., Umair, M., & Khan, P. W. (2018). A Systematic Review of Cyber Security and Classification of Attacks in Networks. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 9(6), 201-207.
[14] Cuervo, E., Balasubramanian, A., Cho, D. K., Wolman, A., Saroiu, S., Chandra, R., & Bahl, P. (2010, June). MAUI: making smartphones last longer with code offload. In Proceedings of the 8th international conference on Mobile systems, applications, and services (pp. 49-62). ACM.
[15] Rimal, B. P., Van, D. P., & Maier, M. (2017). Cloudlet enhanced fiber-wireless access networks for mobile-edge computing. IEEE Transactions on Wireless Communications, 16(6), 3601-3618.
[16] Shaukat, U., Ahmed, E., Anwar, Z., & Xia, F. (2016). Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges. Journal of Network and Computer Applications, 62, 18-40.
[17] Zhang, J., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. IEEE Transactions on evolutionary computation, 13(5), 945-958.
[18] Jiang, M. H., Visser, O. W., Prasetya, I. S. W. B., & Iosup, A. (2017, June). Mirror: A computation-offloading framework for sophisticated mobile games. In 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM) (pp. 1-3). IEEE.
[19] Yokota, H., Kimura, S., & Ebihara, Y. (2004, March). A proposal of DNS-based adaptive load balancing method for mirror server systems and its implementation. In 18th International Conference on Advanced Information Networking and Applications, 2004. AINA 2004. (Vol. 2, pp. 208-213). IEEE.
[20] Mareli, M., & Twala, B. (2018). An adaptive Cuckoo search algorithm for optimisation. Applied computing and informatics, 14(2), 107-115.
[21] Xia, F., Ding, F., Li, J., Kong, X., Yang, L. T., & Ma, J. (2014). Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing. Information Systems Frontiers, 16(1), 95-111.
[22] Malik, S. U., Akram, H., Gill, S. S., Pervaiz, H., & Malik, H. (2020). EFFORT: Energy efficient framework for offload communication in mobile cloud computing. Software: Practice and Experience.
[23] Khan, P. W., Abbas, K., Shaiba, H., Muthanna, A., Abuarqoub, A., & Khayyat, M. (2020). Energy Efficient Computation Offloading Mechanism in Multi-Server Mobile Edge Computing—An Integer Linear Optimization Approach. Electronics, 9(6), 1010.
[24] Abbas, K., Ahmed, K. T., Rafiq, A., Song, W. C., & Seok, S. J. (2020, January). An LTE-WiFi Spectrum Aggregation System for 5G Network: A Testbed. In 2020 International Conference on Information Networking (ICOIN) (pp. 753-755). IEEE.