Turkish Journal of Computer and Mathematics Education *Vol.12 No.6 (2021), 2581-2593*

Research Article

Efficient of Floading Approach for Resource Allocation and Optimization in Mobile Edge Computing Environment

C.Anuradha (Corresponding Author)

M.Ponnavaikko

Abstract: Cloud computing provides a platform for services and resources over the internet for users. The large pool of data resources and services has enabled the emergence of several novel applications such as smart grids, smart environments, and virtual reality. However, the state-of-the-art of cloud computing faces a delay constraint, which becomes a major barrier for reliable cloud services. This constraint is mostly highlighted in the case of smart cities (SC) and the Internet of Things (IoT). Therefore, the recent cloud computing paradigm has poor performance and cannot meet the low delay, navigation, and mobility support requirements.Machine-to-machine (M2M) connectivity has drawn considerable interest from both academia and industry with a growing number of machine-type communication devices (MTCDs). The data links with M2M communications are usually small but high bandwidth, unlike conventional networking networks, demanding performance management of both energy consumption and computing. The main challenges faced in mobile edge computing are task offloading, congestion control, Resource allocation, security and privacy issue, mobility and standardization .Our work mainly focus on offloading based resource allocation and security issues by analyzing the network parameters like reduction of latency and improvisation of bandwidth involved in cloud environment. The cloudsim simulation tool has been utilized to implement the offload balancing mechanism to decrease the energy consumption and optimize the computing resource allocation as well as improve computing capability.

Keywords: Offloading, Resource allocation, Cloudlets, Cloud computing, Mobile edge computing, Performance management

1. Introduction

The use of mobile devices such as smart phones, tablets and notebooks is growing day by day, and the per capita number of connected mobile devices is expected to exceed 1.5 by 2020. This motivates research into the importance of offloading mobile cloud.

We could see that desktop computers are gradually replaced and getting smaller, yet being powerful mobile computing devices. Laptop, Smart phone, Tablets, notebook and e-readers like Kindle are various forms of mobile devices each with different capabilities and characteristics. The development from desktop to mobile device paved way for the development of various third party mobile applications in order to satisfy the ever growing demand of the mobile application users. These growths lead to millions of mobile application with specific requirements that are hosted in different platforms such as App store (Apple 2016), Google's Play (Google 2016) and Microsoft's Windows Phone Store (Azure 2016).

The characteristics of mobile devices that attracted consumers are portability and fast connectivity; however, mobile data processing and storage come as limitations and restrictions. The smaller the unit, the less efficient it is in terms of space, memory and computing, let us take the example of device portability (which is indicated by physical form and size) and Power (which is indicated by resources available). Such trade-offs contribute to the development of numerous types of mobile devices and apps to meet the specifications of the end consumer. Therefore, considering the exponential growth of mobile device technologies, there is a persistent disparity between hardware, device classes, software designed, and implementations.

The smartphone device industry is flooding up with a multitude of apps that use elevated volumes of computation, memory and connectivity in order to close the distance. For starters, the game programme features complex AI features and augmented reality and applications such as the Google glass framework (Google 2016) process vast amounts of real-time sensor data in memory, which often uses complicated computations in natural language processing (NLP). Therefore, personal computers sit behind the equivalents of handheld devices.

In the recent times semiconductor technology is advancing and computation capabilities of hand held mobile devices have increased tremendously. Nowadays with the increase in power to hand held devices, we have built

sophisticated application in them using national language processing and augmented reality. Still memory and storage are limitation though. In smart phones battery life time continues to be another major limiting factor, especially when working on computationally intensive applications. Even when we have smart phones with so many pros, cons like latency sensitivity and compute intensive while using applications like Real time face recognition keeps them beyond capabilities.

Smart applications should be extended with extra tools to make them stronger in order to solve the problems. The latest catch word today is a cloud that is capable of generating unlimited volumes of energy. It is possible to improve the efficiency of computational and resource-intensive activities on smart phones with Cloud (Soyata et al. 2012). In the context of the machines, cloud servers act; consumers get the impression of interacting with the system itself. The platform is called Mobile Cloud Computing (MCC). The user of the application can offload the application to the cloud and return the result to the system itself. Considering the facial recognition case, the system needs to transfer the whole image to the server, which entails major obstacles such as high network latency and restricted Wide Area Network (WAN) bandwidth. There would be changes in QoE if we can decrease the size of the data to be transmitted.

1.1 Mobile Cloud Offloading

We can overcome the performance limitation of running applications in smart phones by using Cloud servers. This can be done by offloading parts that involve intensive computation to Cloud. Following figure 1 represents the benefits of offloading a Chess game running on smart phone, tablet or laptop using cloud. Such games have complex AI mechanisms to predict opponent's next possible move. AI consumes energy and time from mobile devices, extreme usage of resources from mobile makes mobile run out of battery very soon. With the use of mobile cloud offloading, the cloud servers will take the load of computing complex AI engine hence battery usage is less.

To enhance the capabilities of mobile devices by using elastic cloud resources is termed as Mobile Cloud Offloading or Cyber foraging, in which high intensity computation and execution are offloaded to the remote cloud. Once processed the results are synched with mobile device. Cloudlet was one of the earlier technology used in which mobile is augmented with resource for computation from hotspot nearby. Cloudlet is advantageous because of high speed connectivity and low connection latency to the device. Challenges associated with Cloudlet involves complexity in deployment and connectivity in-between Cloudlet and mobile. To overcome this challenge cloudlet has to be designed with lesser level of granularities. Hence we conclude that architecture of cloudlet is not flexible and scalable. Other possible way is to move the computation to remote cloud. When we compare Cloud and Cloudlet architecture, Cloud has latency but it is constantly connected to cellular network.

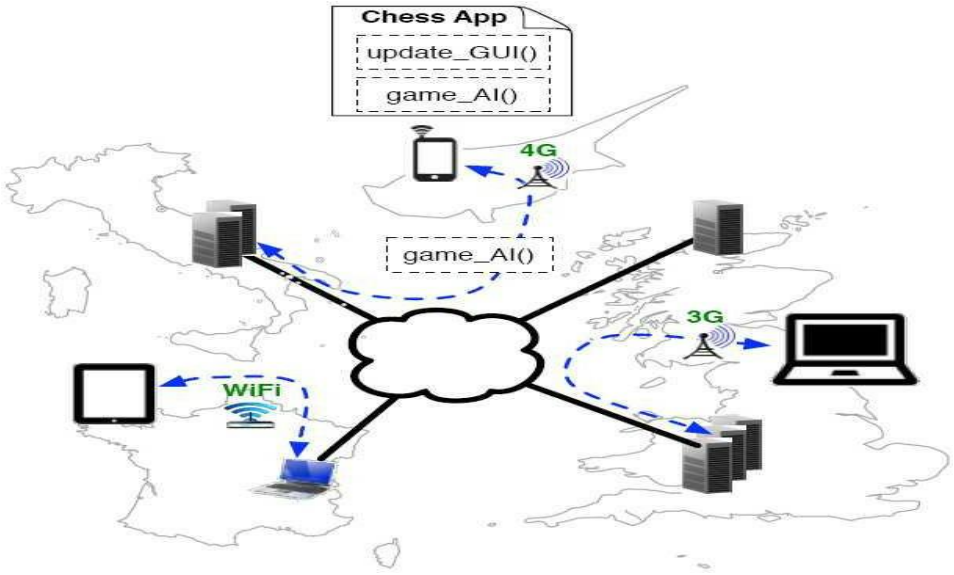

Figure 1. Chess Game offloading scenario

There are various issues encountered while offloading applications to cloud. One of the challenges that arise is if we can offload application directly to nearest cloudlets or to cloud servers. Here important concern is if we can trust the cloudlets.

Next challenge that arises is security and authenticity of the cloud which you wish to offload. Is there possibility to trust cloud severs blindly? What is the level of surety to offload at cloud servers? Is the cloud server trustable with the personal data that is being offloaded from mobile device? One cannot ignore these challenges involved even when cloud is claimed to be 100% secure. Efficient security mechanisms needs present if we offload to cloud servers.

Other challenges are restriction in size, heaviness, charge in battery and heat dissipated from the device (Gartner 2012). Even with day to day improvement in technology such limitations continue to persist in mobile devices when compared to superior counter parts like desktop, laptop, tablet and other severs.

Also we have challenge when we decide to offload, is there a real need to offload complete application or not. One efficient way is to just portioning and offloading the required parts alone, in this case how best this can be achieved? How to split the executable components of application and decide which will reside on cloud?

Last challenge involved depends on mobility of device. Mobile devices are in constant movement, also impacting factors like network bandwidth, charge in mobile battery and changing load, so how are these factors going to be managed with offloading? Hence decision making on cloud offloading needs to taken dynamically. When connected to internet constantly bandwidth and network latency due to traffic exist, thus impacting the application Quality of Service (QoS) and Quality of Experience (QoE).

2. RELATED WORKS

Jay Gandhi et al. (2020) indicated that Mobile Edge Computing (MEC) is an evolving standard where networks are equipped with cloud computing resources such as storage and computing. Together with NFV (Network Element Virtualization) and SDNN, MEC is listed as one of the main 5G technologies (Software Defined Network). For both service providers and consumers, MEC is useful. It is an integral component of the Internet of Things (IoT) and 5G architecture that promises a noticeable decrease in mobile device latency and energy usage. MEC's primary aim is to incorporate cloud facilities and IT helps at the edge of the network in the vicinity of RAN customers (Radio Access Network). MEC tends to decrease latency and increase bandwidth, which is helpful in terms of rapid response time. One of the vital elements of 5G technology is MEC. This thesis included a detailed study including MEC Analogue on the MEC definition. It introduced a distinct edge paradigm along with its architecture, such as fog computing, cloudlet, MEC. During the survey, the numerous issues with its associated work were covered. Real time deployment modelling methods and problems are explained. Finally, there was talk of various MEC scenarios and implementations.

Raditya Martha, et al. (2019) indicated that cloud computing enhancement was evolving very rapidly, and can be calculated from the elements that these technology trends can now accommodate. Cloud infrastructure presents profound changes to the creation, production, and launch and management of IT services. Cloud computing services are usually a virtual application that runs both hardware-based and software-based and SOA on cloud servers, which are sufficient and more efficiently and efficiently handle the computing, software and hardware needs of companies because resources can be more configurable to achieve maximum cost-benefit. Cloud expansion has improved the operating power of those that have used it since the cloud negates the criteria for devices where a person must accomplish a certain role. Infrastructure, repair, upgradeability, and energy costs will greatly reduce tasks that have high hardware specifications for high processing operations that consume significant investment costs.

In addition to cloud computing, eco-friendly/green computing climate-friendly applications that can bring benefit to users/companies and the environment from cloud infrastructure itself have proven that cloud computing has outstanding investment significance for enterprises and job elements that require high processing capabilities. This essay then analyses and synthesises current cloud computing research. The outcomes of this study is to provide an incentive for growth and further analysis in the area of cloud computing usage as an important and productive investment in technology in a business.

On the rapid development of 5G and Internet of Things (IoT) technologies, Xiong Li, et al. (2018) discussed more and more mobile devices with specific sensing capabilities and large amounts of data access to the network. The conventional cloud computing architecture does not fulfil the criteria for IoT implementations, such as low

latency and easy data access. Because of the characteristics of simplicity, simple to use, fast scalability, location isolation and efficiency, cloud computing is generally embraced. Meanwhile, the Internet of Things (IoT) enables any computer to reach the Internet through the growth of sensing technologies and microelectronics technology. The target can be detected by radio-frequency identification (RFID) method in the IoT vision, and wireless sensor networks (WSN) can detect the atmosphere parameters Mobile edge computing can solve these problems and increase the system's execution performance. The authors suggested a privacy protection data aggregation scheme for IoT-assisted mobile edge computing applications in this work. There were three respondents in the model, i.e. terminal computer, edge node, and public cloud hub.

Nasir Abbas, et al. (2017) demonstrated that Mobile Edge Computing (MEC) is an evolving framework that expands cloud computing services to the edge of mobile base station-leveraging networks. It can be extended to smartphone, broadband and wireline scenarios as a promising edge application, leveraging software and hardware platforms, situated at the network edge in the vicinity of end-users. As an integration of cloud computing and mobile computing, Mobile Cloud Computing (MCC) has provided mobile users with significant capacities and equipped them with the storage, computing and energy services offered by the unified cloud. Nevertheless, MCC faces visible problems, such as high latency, security risk, poor coverage and lagged data transfer, with a plethora of mobile devices popping up. MEC has merged numerous device service providers and suppliers efficiently into smartphone users, companies and other vertical markets. In the 5G architecture, it is an essential component that supports numerous creative applications and services where ultra-low latency is required. It presented the concept of MEC, its advantages, architectures, and areas of application; where we emphasise similar studies and future directions in particular.

The decision to offload is an important feature of cloud offloading for mobile devices. Wu et al. (2013) performed the research on the basis of decision making on network availability. As a green operation, time and resources are based on being saved. An domain partitioning and artificial bee colony optimization model is suggested on the basis of the method. Partitioning is concerned on the basis of Bayesian choice. For conversion to graph, the package level technique is used.

The android platform is the subject of the Cuckoo (Kemp et al. 2012) offloading scheme. This model supports both remote execution and local execution. This strategy is specifically updated by Android. The Cuckoo tests all model calls and then checks whether, based on calls and history, offloading is beneficial. This approach requires the use of a remote service driver and a service rewriter module while executing remote execution. This technique requires high speed of calculation and minimal use of resources. In this article, a few weaknesses are addressed here, such as minimal protections and call back service.

Wolski et al. (2008) suggested a solution that, whether offloading is appropriate or not, allows decisions dependent on bandwidth results. Offloading to the cloud would only be useful if the local execution time reaches the remote execution time. The solution to the decision to offload is offered by the Bayesian Decision Dilemma. To make correct choices, a model is used that forecasts bandwidth. To show the best outcomes, this model is compared with other traditional models of decision making.

The dynamic guided adaptation (Kwon & Tilevich 2014) tackles restrictions dependent on offloading. All the intense resources of functionality focuses on the offloading partitions. Based on energy, server capacity, time and protection, the user would have to set the benchmark. The offloading solution is based on an adaptive scheme, transforming the offloading application into a mathematical model of a constraint satisfaction problem (CSP). In order to perform local and remote execution, three servers are involved; time, energy and money are analyzed and compared.

Chathura et al. (2014), suggested an MCC (Mobile Cloud Computing) energy-saving and network-capable algorithm, based on the cloudlet layer of middleware. Mobile and cloud are connected with mobile networks; due to network unavailability, there is a risk of disconnection. There will be no other way to connect to the Cloudlet if the network is lost. In order to deal with data disconnection, several steps are carried out. The model suggested here by Lin et al. (2013) is based on time energy and measures the discharge into the cloud of mobile devices. When offloading, the work provides both time and energy, which lowers reaction time and energy. It requires ternary judgement to make the decision of offloading.

Barbera et al. (2013) focused with examples of real time and considers both connectivity and electricity costs as mobile computing is offloaded. For clones, there are two types of devices involved: off clone, which supports offloading processing, and back clone, which can be used to recover records.

The offloading architecture is new here, Amal et al. (2015), and this work considers actual CPU load and State of Charge (SoC) in cell phones when considering offloading parameters. A definitive judgement on the quality of the encounter is made. Critical delay, energy and equilibrium are other parameters.

The multi-parameter judgement algorithm for offloading mobile computing was suggested by Jessica et al. (2014). This takes several criteria into account, such as local mobile power, channel capacity and battery level. Based on requirements and urgency, the assignment is classified as off-loadable and non-off-loadable and a decision for each module is made.

CADA (Lin et al. 2013) works with context-sensitive algorithms for offloading, and mobile resource movement can influence offloading efficiency. This approach takes into account the user's daily habits and then makes the decision on offloading. If the consumer diverges from the standard actions, CADA would do a wrong prediction.

The following points are considered as existing problems that have been found form the literature survey.

- The existing cloud technology facing low data rates and speed.
- No Virtualization between clouds.
- Maintaining and providing, security and authentication of each clouds separately at third party end is difficult.
- The recent cloud computing paradigm has poor performance and cannot meet the low delay, navigation, and mobility support requirements.

3. CLOUDLETS IN OFFLOADING

A cloudlet is a built building feature that arises from the convergence of versatile processing and distributed computing. As suggested in the 3-layer command chain, it is a middle layer, which consists of a mobile phone, cloudlet and cloud as they are presumably aware of them. The aim of the cloudlet proposal is to "bring the cloud nearer" the mobile phone. Cloudlet illustrates four main characteristics

A cloudlet does not have any hard state, but it may have a cloud-stored state instead. The lack of a complicated state makes it possible to transmit and self-oversee cloudlets. Compared to mobile phones, a cloudlet should illustrate sufficient computational control. It should provide a boundless supply of power (associated with an electrical plug) and a large and reliable cloud network. Near nearby - In terms of dormancy, what's more, high transmitting capability should be near the mobile phone. This usually suggests that, for example, it can be connected with the same neighbourhood structure by means of Wi-Fi.

3.1 STAGES FOR OFFLOADING MECHANISM

Stage 1: Mobile device classification

Stage 2: Priority determination

Stage 3: resource allocation.

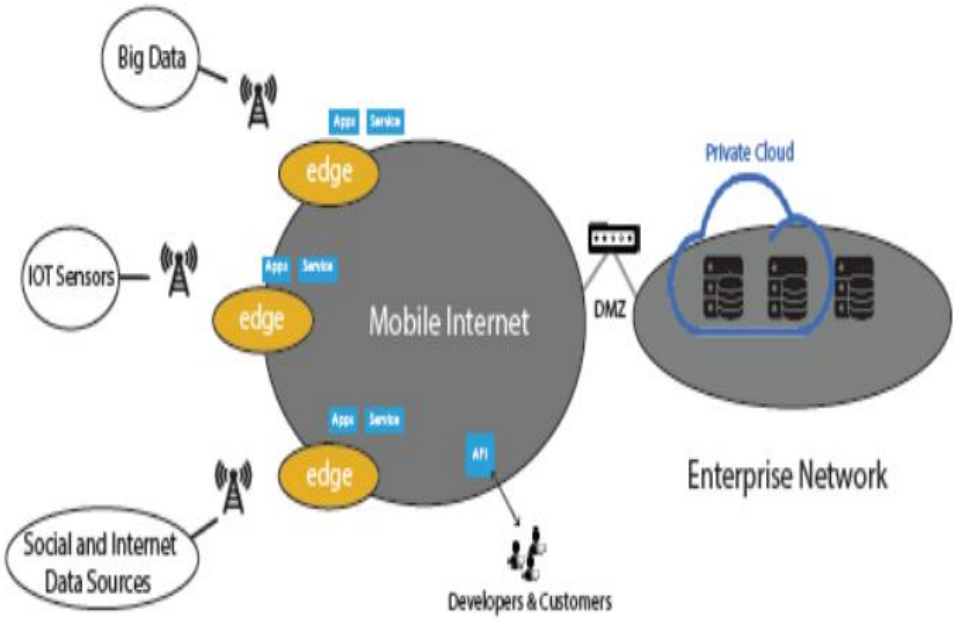

Figure 2. Mobile edge computing architecture

In our proposed system, we made several mobile devices as in the name of cloudlets. The Cloudlets are have data like audio, text, videos that are send to the Data centres. In Telecom network data centres are called as BSS. Then Virtual Machines are MEC cloud storage which are located in Data centre itself.

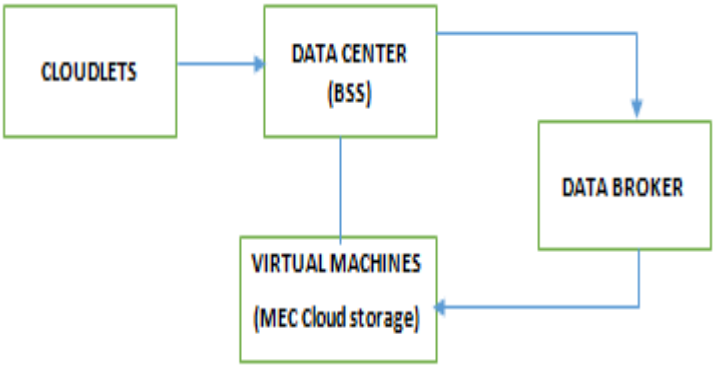

Figure 3. Block Diagram of Proposed System

The data flow from the client are Transfer to Data centre, Data centre create data broker's to find the optimised neighbour Data centres and Virtual machines. After finding Virtual machines and data stored in data centres are virtualize to all virtual machines (MEC Cloud Storages). Then data stored in all Virtual machines, Data broker are deactivated. Client mobile devices can access data from any of virtual machines of the data centres.

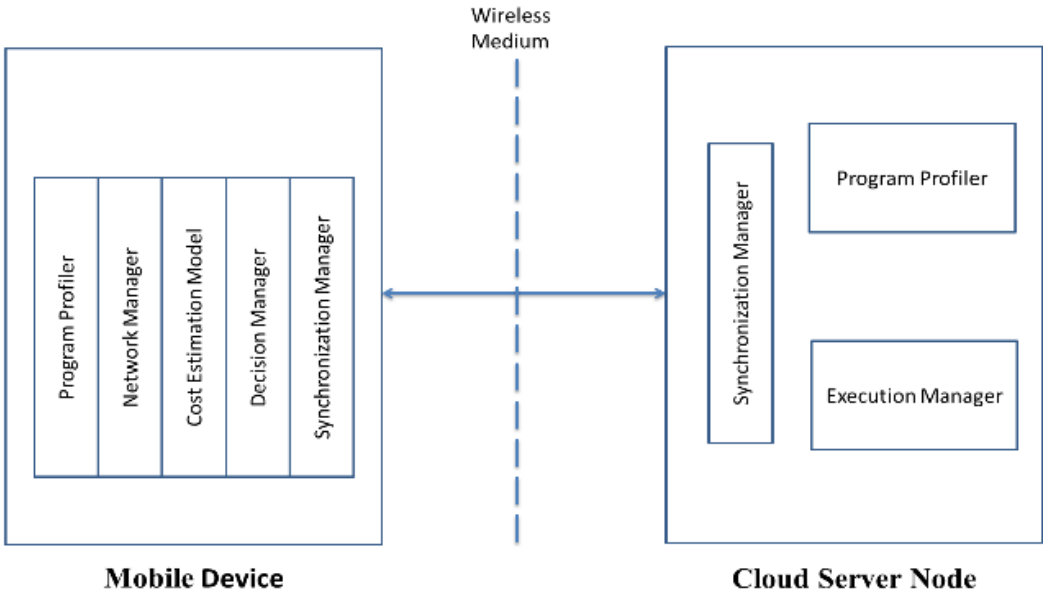

Figure 4 Architecture of proposed framework

The proposed framework embraces client-server communication system, in which the resources provided by the cloud are the servers and the mobile device is the client. On the client side, the structure is comprised of the accompanying segments, to be specific, a program profiler, a network manager, a cost estimation model, a decision manager and a synchronization manager. On the server side, it incorporates a server side synchronization manager, a program profiler and an execution manager. The framework architecture is depicted in Figure 4. The framework components, design and relations between every component are portrayed beneath.

The program profiler keeps track of the running program. The data incorporates 1) all the codes to be executed 2) the memory allotted 3) the execution time and 4) the system execution location (e.g. cloud, local mobile). The profile's information is updated at each execution and is stored in the mobile device database. It is utilized as a part of the cost estimation model to forecast.

The network manager gathers the setting data of the mobile asynchronously at execution time with the goal that it can stored any changes in the connection. The information is sent to the cost estimation models when required. The information being observed incorporates: 1) WiFi association state 2) battery level and its bandwidth.

This is the main part of the framework, comprising of an arrangement of cost estimation models that calculates the offloaded errands' execution cost and a choice making algorithm which is used to make the choice to offload the task. In detail, it takes into account all the information gathered from the program profiler, the execution cost estimation and network manager from the models and chooses at runtime.

The synchronization manager of the structure is in charge for the synchronization of exchange of data between the mobile device and the remote cloud. Synchronization manger keeps running on both the server side and the client side of the system. It gives elements of correspondence keeping up service between client and server, offloaded code execution and managing the server scaling demand. Especially, on the client side, the synchronization manager arranges the offloading instruction and the remote execution procedures, for example, what number of VMs are expected to execute the offloading instruction to the server for running. On the server side, the synchronization manager de-serializes the request from clients, synchronize the status between client OS and server OS and begins the running. The moment synchronization manager de-serializes the request, it verifies if the needed programs and files exist on the server side, provided that this is true, it then processes the client request and returns the results to the server, else, it calls the client to get the file and the corresponding libraries for the remote execution. On the off chance that the server does not execute the task, the handler spares the state and sends the task back to the clients for further running.

The program profiler on the server side keeps track of the time taken to execute the code and returns it to the client. The data is saved as a major aspect of the program profile on the client's gadget database. Once the client and VM is synced, the execution manager processes the request and runs the offloaded code.

The cost model consists of three parts, namely the task execution time denoted by D, wireless channel energy consumption denoted by E and monetary cost denoted by C when related. Then the total cost of executing task ti as shown in Equation 1.

$$C(ti) = \alpha 1 \times D(ti) + \alpha 2 \times pd \times E(ti) \quad (1)$$

Where α 1, and α 2 are weight factors that the system can adjust the portion of the cost to different scenarios. pd is a coefficient reflecting how serious the power consumption on battery life affects the device performance.

4. IMPLEMENTATION AND RESULTS

By benchmarking a sample framework for mobile devices on CloudSim, we test the proposed structure. The Architecture demonstrates the CloudSim programming system's multi-layered outline and its segments of design. CloudSim's early advents used SimJava as the separate occasion recreation engine (Howell and Mc-Nab, 1998) that supports such centre features such as event queuing and sorting, cloud device substances (host, broker, data centre, virtual machines (VM), services), corresponding in the middle of segments, and reproduction clock operation.

The reproduction layer of CloudSim supports the demonstration and recreation of virtualized data centre situations with committed memory, stockpiling, bandwidth and VM administrative interfaces. At this layer, a cloud provider who needs to focus on the competence of different approaches in assigning its hosts to VMs (VM provisioning) would need to execute its procedures. Such use should be possible by automatically extending the usefulness of central VM provisioning. At this level, there is an unmistakable qualification associated with the provisioning of VM hosts. In view of the characterised QoS levels of the SaaS provider, a cloud can be simultaneously assigned to an arrangement of virtual machines that run applications. Furthermore, this layer reveals the features that a cloud application designer can extend to perform complicated job information and application execution research. The User Code (UC) is the top layer in the CloudSim that uncovers essential substances for hosts (number of machines, their particular, et cetera), applications (number of tasks and their necessities), VMs, number of customers and their type of application, and arrangements for merchant bookings.

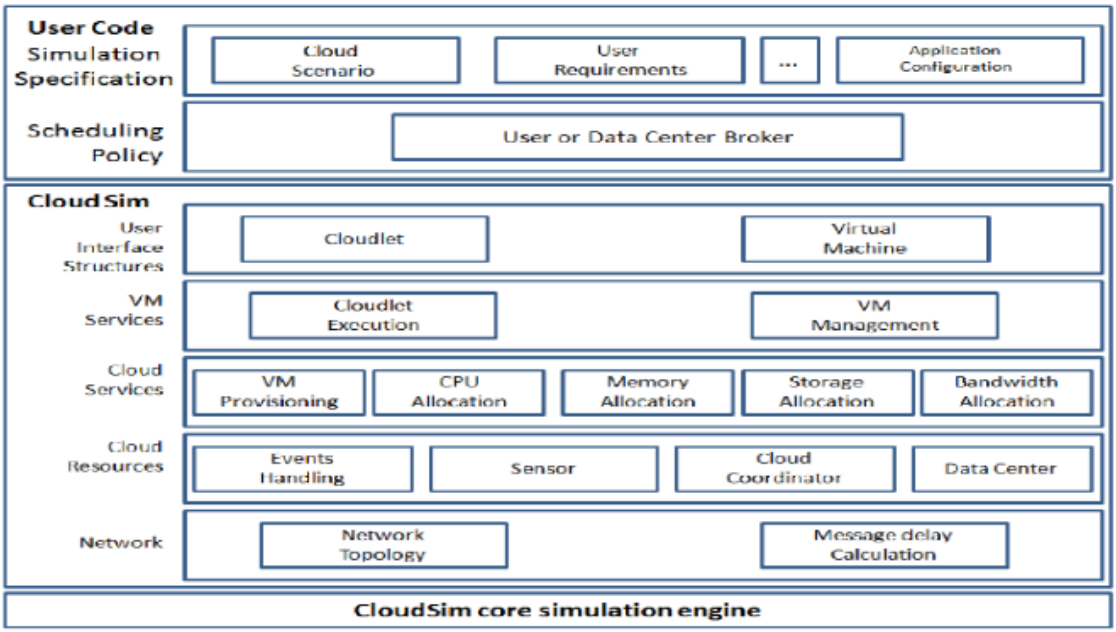

Figure 5. CloudSim Core Simulation Engine

The framework comprises of server nodes that run mobile virtual device instances and laptop mobile device instances. The mobile device is used on the server machine to run an application's offloaded service component at runtime. First, we initialise the CloudSim that generates data centres and cloudlets, then the broker receives the

offloaded application and runs this application on a VM. Any number of VMs can be created by a broker in data centres if an application needs to be executed.

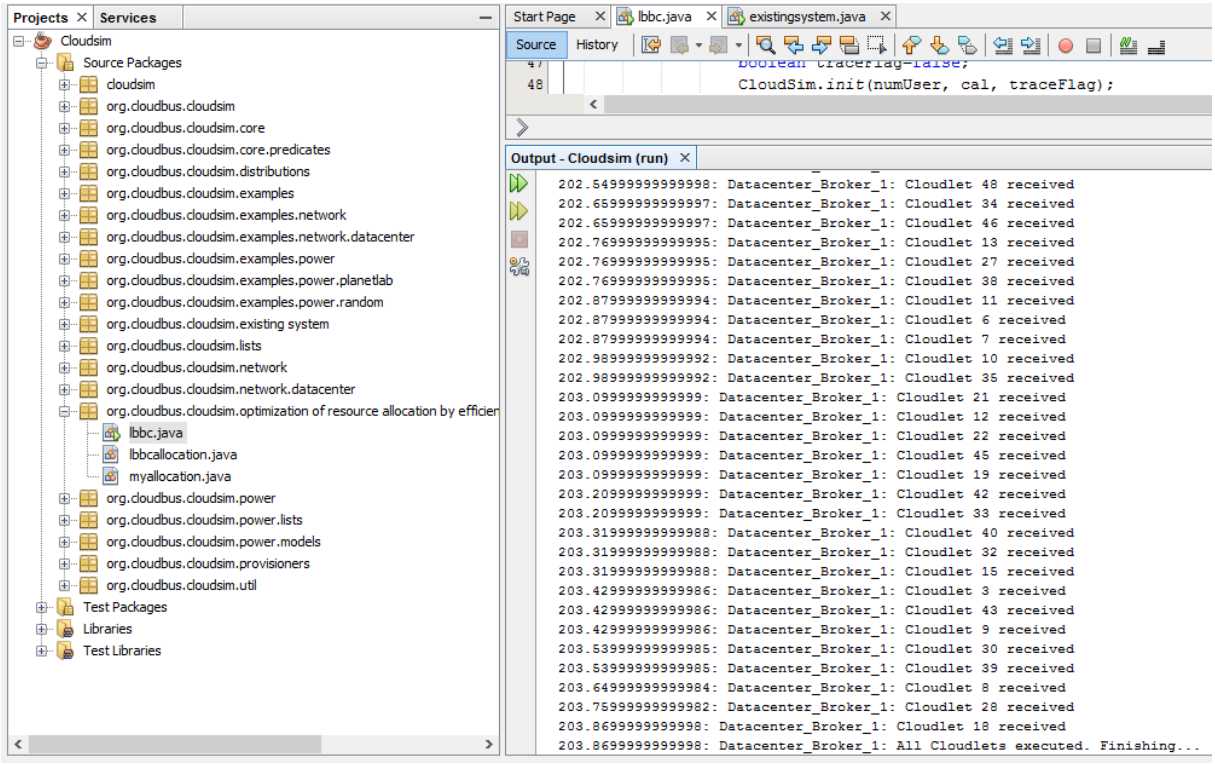

Figure 6. Data access of Cloudlet from VM.

Table 1. TABULATED FORMAT OF RESOURCE ALLOCATION

Cloudlet ID	ID	VM ID	STATUS	EXECUTION TIME	START TIME	FINISH TIME
0	1	1	SUCCESS	200.036	0.1	200.136
1	3	3	SUCCESS	200.146	0.1	200.246
2	14	4	SUCCESS	200.256	0.1	200.356
3	9	9	SUCCESS	200.256	0.1	200.356
4	49	9	SUCCESS	200.256	0.1	200.356
5	11	1	SUCCESS	200.366	0.1	200.466
6	36	6	SUCCESS	200.366	0.1	200.466
7	7	7	SUCCESS	200.476	0.1	200.576
8	13	3	SUCCESS	200.586	0.1	200.686
9	27	7	SUCCESS	200.586	0.1	200.686
10	39	9	SUCCESS	200.586	0.1	200.686
11	10	0	SUCCESS	200.696	0.1	200.796

12	33	3	SUCCESS	200.806	0.1	200.906
13	20	0	SUCCESS	200.916	0.1	201.016
14	15	5	SUCCESS	201.026	0.1	201.126
15	21	1	SUCCESS	201.136	0.1	201.236
16	24	4	SUCCESS	201.136	0.1	201.236
17	19	9	SUCCESS	201.136	0.1	201.236
18	29	9	SUCCESS	201.249	0.1	201.349
19	0	0	SUCCESS	201.359	0.1	201.459
20	41	1	SUCCESS	201.359	0.1	201.459
21	31	1	SUCCESS	201.469	0.1	201.569
22	22	2	SUCCESS	201.469	0.1	201.569
23	23	3	SUCCESS	201.469	0.1	201.569
24	46	6	SUCCESS	201.469	0.1	201.569
25	35	5	SUCCESS	201.579	0.1	201.679
26	18	8	SUCCESS	201.579	0.1	201.679
27	38	8	SUCCESS	201.689	0.1	201.789
28	40	0	SUCCESS	201.799	0.1	201.899
29	30	0	SUCCESS	201.909	0.1	202.009
30	43	3	SUCCESS	201.909	0.1	202.009
31	4	4	SUCCESS	202.019	0.1	202.119
32	34	4	SUCCESS	202.129	0.1	202.229
33	44	4	SUCCESS	202.129	0.1	202.229
34	26	6	SUCCESS	202.129	0.1	202.229
35	17	7	SUCCESS	202.129	0.1	202.229
36	12	2	SUCCESS	202.239	0.1	202.339
37	6	6	SUCCESS	202.239	0.1	202.339
38	16	6	SUCCESS	202.349	0.1	202.449
39	42	2	SUCCESS	202.517	0.1	202.617
40	2	2	SUCCESS	202.627	0.1	202.727
41	5	5	SUCCESS	202.627	0.1	202.727
42	28	8	SUCCESS	202.627	0.1	202.727
43	25	5	SUCCESS	202.73	0.1	202.83
44	32	2	SUCCESS	202.84	0.1	202.94
45	45	5	SUCCESS	202.84	0.1	202.94

46	37	7	SUCCESS	202.84	0.1	202.94
47	47	7	SUCCESS	202.84	0.1	202.94
48	48	8	SUCCESS	202.988	0.1	203.088
49	8	8	SUCCESS	203.104	0.1	203.204

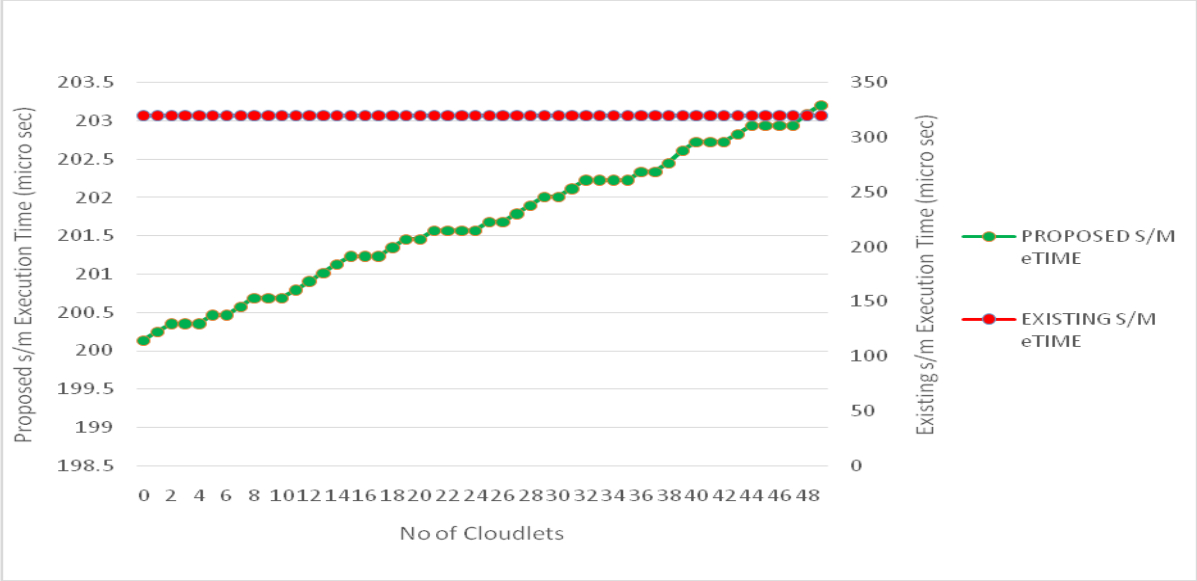

Figure 7. Execution time analysis for Resource Allocation

We can observe from the comparison graph that the execution time is reduced, which shows that our system is optimised and that data speed access to cloud storage is increased. It improves our data access speed due to the efficient allocation of resources and optimised algorithmic strategy in the network as shown in figure 7.

5. CONCLUSION

The use of flexible as a single material to do numerous registration initiatives and the increase in dependence on it made it a highly powerful discovery field. In the processing area for improving portable use and improving its service, a few examination results were collected. In the writing on energy-productive processing and improving the energy use of the versatile through distributed computing use, not many oddities are found. With the assorted existence of flexible figuring energy demands, this postulation will fill the void with clever ideas, estimation and machinery development scheduleThrough an audit of the current calculations, the broad writing study gave outstanding material to new analysts to have bits of knowledge in versatile energy-productive design. The usage of ongoing clouds, such as Amazon Web Services (AWS), would include a fair report of the planned new procedures. An engineering-based approach is intended to expand cell phone power, Virtual Network Computing (VNC) to offload cell phone-to-cloud calculation, use of Amazon Web Services (AWS).

We also explored the MEC offloading processes in heterogeneous 5G networks in this work. In order to increase the energy efficiency of the discharge device, a dilemma has been devised to minimize the energy usage of the execution of the computing task along with that of the contact process. Simulation is often conducted with different inputs and conditions to show the authenticity of the proposed algorithms and the algorithm's actions. The CloudSim simulator is used to test offloading results. Overall, the task would be a major bridge with the use of cloud technology to provide the energy requirements and energy utilisation values. More efficiently solving the problem, which jointly optimises the decisions on computing offloading and the strategies for the allocation of radio resources

to minimize the energy cost of the system under delay constraints. In addition, we carried out a simulation study that clearly demonstrates the improvement in energy efficiency.

References

1. Amal, Ellouze, Maurice, Gagnaire, Ahmed& Haddad 2015, 'A Mobile Application Offloading Algorithm for Mobile Cloud Computing', *3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, pp. 34-40.
2. Archana Srivastava (2014). A Detailed Literature Review on Cloud Computing,*Asian Journal of Technology & Management Research Vol. 04*
3. Barbera, M V, Kosta, S, Mei, A & Stefa, J (2013). To Offload or Not to Offload? The Bandwidth and Energy Costs of Mobile Cloud Computing,*IEEE INFOCOM*, pp. 1285 - 1293
4. Chathura MSM, Kun, Yanga, Liang, Hub & Jianming, Zhangc (2014). Energy Efficient and network aware offloading algorithm for Mobile Cloud Computing, *Special Issue on Mobile Computing for Content/Service-Oriented Networking Architecture*, *vol. 74-9*, pp. 22– 33
5. Dario Sabella, Alessandro Vaillant, Fabio Giust (2016). Mobile Edge Computing Architecture The role of MEC in the Internet of Things,*IEEE Consumer Electronics Magazine*.
6. Fatema Vhora, Jay GandhiA (2020). Comprehensive Survey on Mobile Edge Computing: Challenges, Tools, Applications, *International Conference on Computing Methodologies and Communication*
7. Flores, H & Srirama, S (2013). Adaptive code offloading for mobile cloud applications: Exploiting fuzzy sets and evidence-based learning, *Proceeding of ACM Workshop MCS-Mobisys*, pp. 9-16
8. Gartner Group. Press Release: Gartner Says Worldwide Sales of Mobile Phones Declined 2 Percent in First Quarter of 2012 (2012).http://www.gartner.com/it/page.jsp?id=2017015
9. Gede Indra Raditya Martha and Apol Pribadi (2019). A Literature Review – Firm Investment on Cloud as Efficient & Effective Technology, *International Conference on Electronics Representation and Algorithm (ICERA)*
10. Gember, Aaron, Chris, Dragga & Aditya, Akella (2012). ECOS: Practical Mobile Application Offloading for Enterprises, *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*, pp. 199-210
11. Haibo Yang, Mary Tate, A Descriptive Literature Review and Classification of Cloud Computing Research, *Communications of the Association for Information Systems: Vol. 31*, Article 2, 2
12. Wu, H, Wolter, K & Grazioli, A (2013). Cloudlet-based Mobile Offloading Systems: A Performance Analysis, *31st International Symposium on Computer Performance Modeling*, pp. 24-26.
13. Haijian Sun, Fuhui Zhou, Rose Qingyang Hu (2019). Joint Offloading and Computation Energy Efficiency Maximization in a Mobile Edge Computing System,*IEEE Transactions on Vehicular Technology*, *Vol. 68*, Issue: 3
14. Jessica, O, Emilio, CS & Sergio, Barbarossa (2014). Multi-parameter Decision Algorithm for Mobile Computation Offloading, *IEEE Wireless Communications and Networking Conference*, pp. 3005-3010
15. Jinke Ren, Guanding Yu, Yinghui He (2019). Collaborative Cloud and Edge Computing for Latency Minimization, *IEEE Transactions on Vehicular Technology*
16. Junaid Qadir, Anwar Khan, et al. (2016). Towards Mobile Edge Computing: Taxonomy, Challenges, Applications and Future Realms, *Preparation of Papers for IEEE Transactions and Journals*
17. Ke Zhang, Yuming Mao, Supeng Leng, Quanxin Zhao (2016). Energy-efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks, *IEEE Publications, Vol. 1*
18. Kemp, R, Palmer, N, Kielmann, T, Bal, H (2012). Cuckoo: a computation offloading framework for smartphones, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 76*, pp. 59-79
19. Kwon, YW & Tilevich, E (2014). Constraint-driven dynamic adaptation of mobile applications for quality of service, *6th International Conference on Mobile Computing, Applications and Services*, pp. 143- 152

20. Lin, TY, Lin, TA, Hsu, CH, King, CT (2013). Context-aware decision engine for mobile cloud offloading, *IEEE conference on wireless communications and networking*, pp. 111–116
21. Lixing Chen, Sheng Zhou, and Jie Xu (2018). Computation Peer Offloading for Energy-Constrained Mobile Edge Computing in Small-Cell Networks,*IEEE/ACM Transactions on Networking*
22. Lola Yorita AstriA Study (2015). Literature of Critical Success Factors of Cloud Computing in Organizations, *International Conference on Computer Science and Computational Intelligence (ICCSCI)*
23. Luyue Ji and Songtao Guo (2018). Energy-Efficient Cooperative Resource Allocation in Wireless Powered Mobile Edge Computing,*IEEE Internet of Things Journal*
24. Meng Li, F. Richard Yu, Pengbo Si, and Yanhua Zhang (2018). Energy-efficient Machine-to-Machine (M2M) Communications in Virtualized Cellular Networks with Mobile Edge Computing (MEC), *IEEE Transactions on Mobile Computing Conference*
25. Nasir Abbas, Yan Zhang, Amir (2017). Mobile Edge Computing: A Survey,*IEEE Internet of Things Journal*
26. Rania Fahim El-Gazzar (2014). A Literature Review on Cloud Computing Adoption Issues in Enterprises,*IFIP Advances in Information and Communication Technology*
27. Sabyasachi Gupta, Jacob (2020). Lifetime Maximization in Mobile Edge Computing Networks,*IEEE Transactions on Vehicular Technology*
28. Selim, Rich Wolski, Chandra Krintz, Dan (2008). On the Efficacy of Computation Offloading Decision-Making Strategies, *International Journal of High Performance Computing Applications, vol. 22*, Issue 4, pp. 460-479
29. Selim, Rich Wolski, Chandra Krintz, Dan (2008). On the Efficacy of Computation Offloading Decision-Making Strategies, *International Journal of High Performance Computing Applications, vol. 22*, Issue 4, pp. 460-479
30. Shengali Pan, Zhiyong Zhang, et al (2019). Dependency-Aware Computation Offloading in Mobile Edge Computing: A Reinforcement Learning Approach,*Special Section on Intelligent Data Sensing, Collection and Dissemination in Mobile Computing*
31. Shivarudrappa, Deepak, Shashank, Bharadwaj & MingLung, Chen (2011). COFA: Automatic and Dynamic Code Offload for Android, *Boulder, CO, USA*
32. Soyata, T, Muraleedharan, R, Funai, C, Kwon, M & Heinzelman, W (2012). Cloud-Vision: Real-Time face recognition using a Mobile- Cloudlet-Cloud acceleration architecture,*Proceedings of the 17th IEEE Symposium on Computers and Communications*, pp. 59-66
33. Xiaoyi Tao, Kaoru Ota, Mianxiong Dong, Heng Qi (2017). Performance Guaranteed Computation Offloading for Mobile-Edge Cloud Computing, *IEEE Wireless Communications Journals*
34. Xiong Li, Shanpeng Liu, Fan Wu, Saru Kumari (2018). Privacy Preserving Data Aggregation Scheme for Mobile Edge Computing Assisted IoT Applications,*IEEE Internet of Things Journal*