JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 8, ISSUE 04, 2021

CLOUDLET BASED PRAPOSED NOVEL FRAMEWORK FOR IMPROVEMENT OF MCC (MOBILE CLOUD COMPUTING PERFORMANCE)

[1]Dr. Satendra Kurariya

[1]Research Guide, Dept. of Computer Science & Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal Indore Road, Madhya Pradesh, India.

[2]Pradeep Kumar Shriwas

[2]Research Scholar, Dept. of Computer Science & Engineering, Sri Satya Sai University of Technology & Medical Sciences, Sehore, Bhopal-Indore Road, Madhya Pradesh, India

Abstract

Even though smart phones are used by a lot of people, they don't have a lot of features. For example, they have weak processors, not much memory, and short battery lives. Mobile Cloud Computing (MCC) was created to make up for what smartphones couldn't do on their own. MCC uses Cloud Computing (CC) for its processing power and storage space, but the smartphone itself doesn't use much power. But sending jobs from smartphones to the cloud is still a major area of research, with the goals of maximising performance and resource use and improving service quality as a whole (QoS). With the help of a decision engine, we describe a method in which two servers are used in turn: one for First Upload Round (FUR) offloading and another for Second Upload Round (SUR) offloading. This is what we mean when we say "offloading." Compared to other cutting-edge technologies, our approach not only uses less energy, but it also works better. It has been shown that the proposed method cuts down on power use, with 4G showing the most benefit. Power use was cut by 93% when a 10-megabyte file was sent over 4G and by 85% when it was sent over Wi-Fi. To cloudlet based praposed novel framework for improvement of mcc (mobile cloud computing performance).

Keywords : Cloudlet , Mobile Cloud Computing Performance ,

ISSN- 2394-5125 VOL 8, ISSUE 04, 2021

1. Introduction

Researchers at the CCS Insight Institute think that by 2021, there will be 2.35 billion mobile phones in use around the world. This huge market share drives research and development in mobile cloud computing that aims to meet the needs of users who are becoming more picky. Cloud services can be used to meet the strict real-time needs of many modern applications. Given these limits, it is hard to meet expectations, especially when it comes to cloud infrastructures that are complicated.

Manufacturers keep adding more computing cores, GPUs, cameras, and other types of sensors[1] to these devices to make them more powerful. But this kind of progress leads to a rise in the complexity of applications that grows at a very fast rate[21,22]. The goal of Mobile Cloud Computing (MCC) is to move tasks that use a lot of resources from devices with limited resources to the cloud, which has more resources. By taking this approach, it may be possible to get more resources to help carry out the initiatives.

Offloading computation is a good way to meet response time requirements on mobile devices or other systems with limited resources. There's a chance that cloud storage could help a lot of apps on mobile devices that need to store a lot of data. [2] offer a navigational robot app that helps robots find things around them so they don't run into each other or get into other trouble. The robot may need to send its calculations to faster servers[23]. Context-aware computing is another type of application that can benefit from offloading computing tasks. This way of using computers involves analysing many streams of data from different sources, such as global positioning system (GPS) devices, maps, temperature sensors, and so on, to learn about a user's immediate surroundings.

MCC service providers offer "offloading," which is the process of taking some of a client's work off their hands[24], to those who need it. This provider owns both the server hardware and the offloadingframework that the apps run on. Offloading frameworks all have the same goal: to move processing from a client device to a server or other remote location, where it can be done by something more capable. These frameworks use different architectural styles and technical approaches, but they all try to do the same thing. Researchers have made a number of offloading frameworks in order to make the MCC paradigm easier for programmers to use.

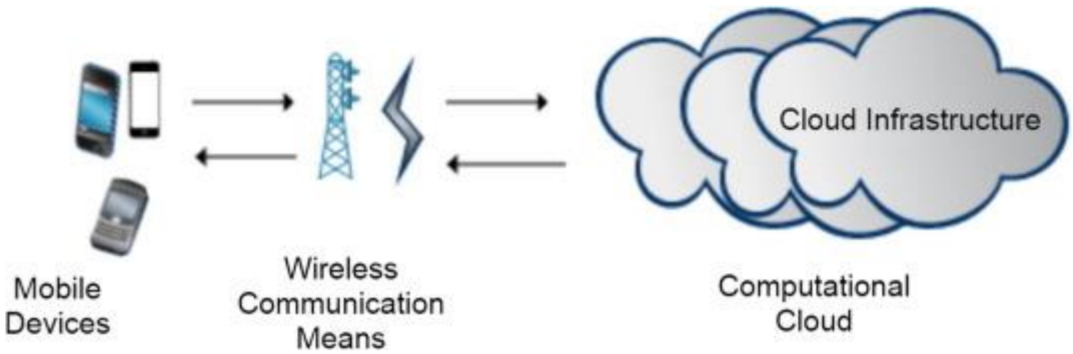

Figure 1: General structure MCC[11]

You can't just drop tasks without paying for it in some way. Depending on how the service provider makes money, the end user might pay for the cloud resources or the MCC provider might pay for them. Most of an MCC [25] provider's costs come from buying the necessary equipment at the beginning. On the other hand, the costs for the end user usually depend on how much the offloading framework is used and, in particular, how many surrogates[3] are put on the cloud. Planning an MCC infrastructure is hard because it has to work for a wide range of use cases. During the design phase, it is obviously impossible[26] to measure a system in the real world. The need to specify a lot of things that haven't been decided yet makes prototype implementations impossible. In this case, system simulation is very helpful.

2. Related work

F. Fakhfakh et al. (2019) Tablets, smartphones, and other portable electronics are becoming a big part of modern life very quickly. These gadgets give us good ways to talk to each other that aren't limited by time or place. Even so, it's hard for them to keep supplies on hand and communicate with each other. To get around these problems, a new way of doing things called Mobile Cloud Computing (MCC) has come into being. MCC makes Cloud Computing available on mobile devices. When this paradigm is used, it brings up a lot of problems, especially in terms of performance and accuracy. But MCC makes it hard to get the most out of performance for a number of reasons. One of these is having goals that are at odds with each other. But because the apps being looked at were so complicated, it was more likely that mistakes would be made when they were being modelled. In this essay, we will look at where MCC came from, define it, and then look at the many ways it can be used. Also, we describe the approaches that have been suggested in the literature, with a focus on the verification and performance of MCC systems.

Then, we give a detailed evaluation of the different available options based on a number of factors. In the end, we talk about the most important questions that the study raised and make some suggestions for future research.

F. A. Silva et al. (2018) Mobile Cloud Computing (MCC) could help improve the performance of mobile apps that use a lot of resources by sending labor-intensive tasks to cloud computing infrastructures. To finish this process, the first step is to break the programme down into its parts and figure out which ones can be given to other people or groups. The code is broken up into a number of method calls, which are then sent to different places on the Internet. It is common to use a lot of servers to handle hard computing jobs that need to process a lot of applications at once. It's not easy to guess how apps that can run at the same time will behave. How well the apps work and how much it costs to use the cloud depend on how many remote servers are used. Users want to use as many servers as possible so that applications run faster. But users also want to save money, so they'd rather not use as much cloud space. In this study, we suggest using stochastic Petri Nets (SPN) to model how mobile cloud system method calls are carried out. The method lets a designer plan and optimise MCC configurations by modelling system behaviour with SPNs and then estimating how long it will take to run applications that can be run in parallel.

Francisco Airton Silva et al. (2017) Mobile Cloud Computing (MCC) could help improve the performance of mobile apps that use a lot of resources by sending labor-intensive tasks to cloud computing infrastructures. To finish this process, the first step is to break the programme down into its parts and figure out which ones can be given to other people or groups. The code is broken up into a number of method calls, which are then sent to different places on the Internet. It is common to use a lot of servers to handle hard computing jobs that need to process a lot of applications at once. It's not easy to guess how apps that can run at the same time will behave. How well the apps work and how much it costs to use the cloud depend on how many remote servers are used. Users want to use as many servers as possible so that applications run faster. But users also want to save money, so they'd rather not use as much cloud space. In this study, we suggest using stochastic Petri Nets (SPN) to model how mobile cloud system method calls are carried out. The method lets a designer plan and optimise MCC configurations by modelling system behaviour with SPNs and then estimating how long it will take to run applications that can be run in parallel.

F. Silva, et al (2018) Mobile Cloud Computing (MCC) could help improve the performance of mobile apps that use a lot of resources by sending labor-intensive tasks to cloud computing infrastructures. To finish this process, the first step is to break the programme down into its parts and figure out which ones can be given to other people or groups. The code is broken up into a number of method calls, which are then sent to different places on the Internet. It is common to use a lot of servers to handle hard computing jobs that need to process a lot of applications at once. It's not easy to guess how apps that can run at the same time will behave. How well the apps work and how much it costs to use the cloud depend on how many remote servers are used. Users want to use as many servers as possible so that applications run faster. But users also want to save money, so they'd rather not use as much cloud space. In this study, we suggest using stochastic Petri Nets (SPN) to model how mobile cloud system method calls are carried out. The method lets a designer plan and optimise MCC configurations by modelling system behaviour with SPNs and then estimating how long it will take to run applications that can be run in parallel.

V. Vijayalakshmi D et al. (2015) Mobile cloud computing, or MC2 for short, is a new way of thinking about computing that aims to improve the performance of limited-resource devices running apps that use a lot of data and computation. But mobile devices only have a certain amount of memory, processing power, and battery life. It's also possible that these devices can't run software that's too hard for them. The problem could be fixed by moving some tasks to servers with more resources. As a result, performance will get better and the amount of resources used will go down. When people try to improve the quality and performance of mobile apps, they don't think much about how to take care of the servers that the apps use to do their jobs. In this study, we come up with two approaches that work well together. In one, data processing for a mobile app happens on the device itself. In the other, data processing happens on a remote server. In the next sections, we'll look at these two approaches in more depth. These methods are used so that complicated compression offloading techniques don't have to be set up. By comparing the two strategies described above, you can find out which one works better. Depending on how big the application is and what it needs to do, one method may be better than

ISSN- 2394-5125 VOL 8, ISSUE 04, 2021

the other. The number of resources used by an application may be related to how well it works. We also talk about a method called Dot Net Nuke (DNN) that can be used to manage and frequently update the mobile server according to the user's needs without the user having to know how to code. Because of this, the quality of work done with mobile computing has gone up in general.

3. Methodology

First, requesting mobile devices are put into groups based on where they can work. Then, the fuzzy K-nearest neighbour method is used to put the mobile devices close to the cloud resources they need. Here, we'll figure out how each gadget can move after allocating[4] resources and figuring out the best routes. Since mobile devices are always moving, the connection they made to the cloud or cloudlet will be broken as soon as they move to a new area. So, the proposed method uses the HMM to predict where the mobile device will be in the future based on how it has moved in the past. Assuming that the location of the mobile device has already been figured out, the next step is to find the shortest route between the device and the cloud. This is done with the help of an ACO, which uses the most recent information about the animal's pheromones to find the quickest way to its food source.

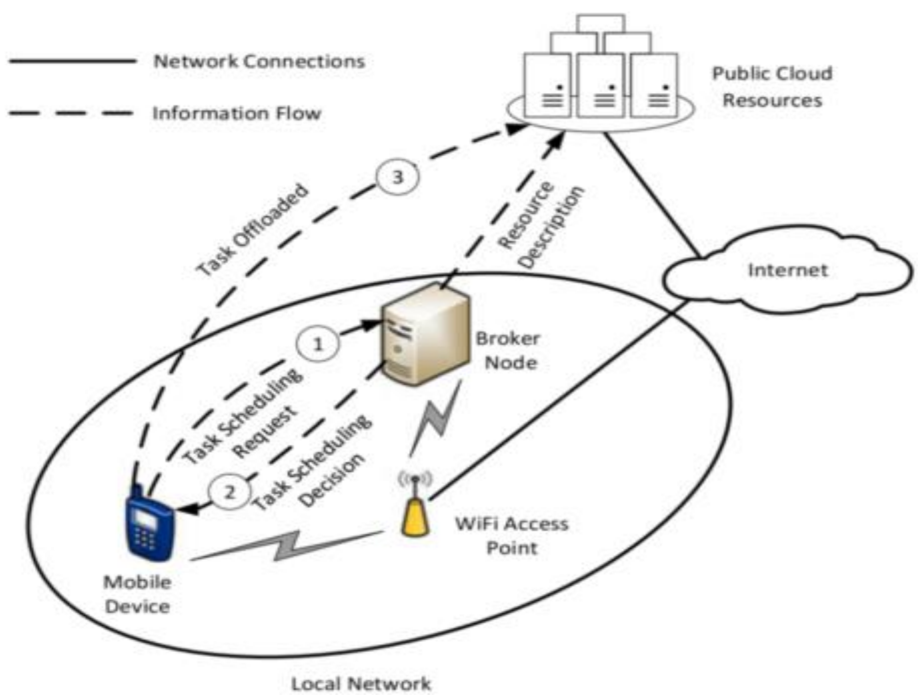

Figure 2; Simple working structure[13]

ISSN- 2394-5125 VOL 8, ISSUE 04, 2021

The proposed compute offloading is shown in more detail in the block diagram[5] below. The number of mobile users who ask for cloud services is used to show how much energy the method could use, how long it would take to process requests, and how much it would cost. Table.1 shows the settings that were used for the simulation.

Table .1 Simulation Parameters

Parameter	Values
Number of Mobile Users	50 - 100
Simulation area	200 m
Total Power of Transmission	100Mw
Channel Bandwidth	2Mhz
Processing Speed of ER	1100 Mega Cycles
Processing Speed of CS	1300 Mega Cycles

Speeding up the time it takes to process mobile applications is an important goal. In the past, data from mobile apps were sent to a remote cloud using a compression method[5]. This may help applications that need to process large amounts of data because it cut down on the amount of data that had to be sent to the cloud. But programmes that work with small amounts of data don't have to go through the compression offloading process. Because of this activity, there will be more complexity. Instead of suggesting a compression algorithm, the paper suggests two ways to figure out and shorten the time it takes for a mobile app to do what it needs to do. We also have a lot of faith that putting limits on how long a mobile app can run will lead to less power being used by the device.

Web Server Application Development

Dot Net Nuke made the software that runs on the server side of the building management system. With this programme, the app on the phone can talk to each other.

DNN's features, like the fact that the server was made, make it easy to change and often updated. We've talked about the skills listed above. In the end, the facilities management system was put together with a mobile app made with Android Studio and a server app[6] made with DNN. The FMS is a set of tools that are centred around the server. It has modules like Resource Manager, Resource rate management, and Booking reporting.

In the list below, you can see what each of these sub-modules does[7]. With their help, it only takes a few minutes to update and fix the server.

1. Resource Manager: The Resource management module takes care of all of this information, like where the room, hall, or other space is and what kind of resources are in it. The manager of resources puts each building into a category[8] based on its location and type to make maintenance easier.
2. Resource Rate Management: This is because the rates of use of different resources (facilities) can't be directly compared because each resource (facility) is different. When deciding how often to consume, the price and the needs of the many organisations are taken into account. The information about how much the item will cost
3. Booking Reports: The booking reports module has all the information about the services that mobile customers have booked or reserved. Keeping track[9] of bookings and reservations helps make sure that resources aren't booked twice and that two people don't use the same space at the same time.

4. Performance Evaluation

Android Application Studio was used to make the FMS mobile app. In order to compare how well the two methods work, the programme needs to be made so that it can be run in either mode. We can say that one method is better for making large, resource-intensive programmes and the other is better for making smaller apps because either method can be used to figure out how long an app takes to run. Taking screen size[10] into account when making mobile apps in the future could lead to more nuanced requirements for how they are used. When making an FMS mobile app, developers often have to work with two different modules.

1. Display Resources: The "display resource" client-side module[11] shows a list of all the facilities and resources that mobile users can use and book. The server-side execution strategy's display resource module will first get all[12] of the data from the

server before showing the user the accessible resource. So, the mobile client and the server are always sending and receiving data.

But when the Client Side execution method is used, the display resource module does not talk to the server to see if resources are available[13]. Instead, it does this check locally on the mobile device. This lets it show which resources are currently available.

2. Book Resource: The Book resource module works with the server-side execution modules so that any changes that need to be made can be made before[14] a disabled guest makes a reservation to use the facility.

5. Conclusion

The use of cloud computing on mobile devices is being looked into as a possible way to improve the speed and battery life of mobile devices. Even so, designing and building MCC infrastructures isn't easy due to the complexity of the systems. Choosing how many remote servers to use, for example, can affect how fast the apps run and how much it costs to use the cloud. If service providers knew how much cloud computing cost, they could give their customers better service and save money at the same time. In this research, we suggest that application partitioning be shown using an SPN modelling approach. Charts showing the Throughput, MTTE, and CDFs are made by the method. Based on a number of different model evaluation techniques, our group made the MCC-Adviser tool and put it through a lot of tests. The goal of making this kind of tool is to make it easier for software engineers to plan the mobile cloud architecture of their companies. We think that ours is the first automatic implementation of SPN in MCC that we know of. After testing MCC-Adviser on a local area network (LAN), we made it available to the public as an interactive online application that can be seen at http://cin.ufpe.br/faps/mcc-adv/. Combine MCC-Adviser with other ways to improve mobilecloud performance for even more gains.

Reference

[1].F. Fakhfakh, "Performance and Correctness of Mobile Cloud Computing Systems: Taxonomy and Open Challenges," 2019 15th International Wireless Communications &

Mobile Computing Conference (IWCMC), 2019, pp. 1019-1024, doi: 10.1109/IWCMC.2019.8766588.

[2]. F. A. Silva et al., "Mobile Cloud Performance Evaluation Using Stochastic Models," in IEEE Transactions on Mobile Computing, vol. 17, no. 5, pp. 1134-1147, 1 May 2018, doi: 10.1109/TMC.2017.2749577

[3]. **Francisco Airton Silva**, **Sokol Kosta**,**Matheus Dornelas Rodrigues**,**Danilo Oliveira**, Mobile Cloud Performance Evaluation Using Stochastic Models**,** September 2017IEEE Transactions on Mobile Computing PP(99):1-1DOI:10.1109/TMC.2017.2749577

[4]. F. Silva, et al.,"Mobile Cloud Performance Evaluation Using Stochastic Models" in IEEE Transactions on Mobile Computing, vol. 17, no. 05, pp. 1134-1147, 2018. doi: 10.1109/TMC.2017.2749577

[5]. V. Vijayalakshmi D. Sundaranarayana I. Shamili, 2015, Performance Efficient Application Development in Mobile Cloud Computing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) TITCON – 2015 (Volume 3 – Issue 16),

[6]. J. Liu, E. Ahmed, M. Shiraz, A. Gani, R. Buyya, and A. Qureshi,“Application partitioning algorithms in mobile cloud computing:Taxonomy, review and future directions,” Journal of Network andComputer Applications, vol. 48, pp. 99 – 117, 2015

[7]. F. A. Silva, M. Rodrigues, P. Maciel, S. Kosta, and A. Mei, “Plan-ning mobile cloud infrastructures using stochastic petri nets andgraphic processing units,” in 2015 IEEE 7th International Conferenceon Cloud Computing Technology and Science (CloudCom), Nov 2015,pp. 471–474.

[8]. J. Mendonc¸a, R. Lima, E. Andrade, and G. Callou, “Assessingperformance and energy consumption in mobile applications,”in Systems, Man, and Cybernetics (SMC), 2015 IEEE InternationalConference on, Oct 2015, pp. 74–79

[9]. CCS-Insight-Forecast, “Smartphone sales to peak in western markets in 2017 as they enter new phase of maturity,” 2015. [Online]. Available: http://tiny.cc/ih0y6x

[10]. F. A. Silva, M. Rodrigues, P. Maciel, S. Kosta, and A. Mei, “Planning mobile cloud infrastructures using stochastic petri nets and graphic processing units,” in Proc. IEEE 7th Int. Conf. Cloud Comput. Technol. Sci., Nov. 2015, pp. 471–474.

[11]. Akherfi, Khadija, et al. “Mobile Cloud Computing for Computation Offloading: Issues and Challenges.” Applied Computing and Informatics, vol. 14, no. 1, Jan. 2018, pp. 1–16, 10.1016/j.aci.2016.11.002.

[12]. RakanAldmoura,SufianYouse,fbTharBakerc,ElhadjBenkhelif. An approach for offloading in mobile cloud computing to optimize power consumption and processing time. https://doi.org/10.1016/j.suscom.2021.100562

[13]. M. Nir, A. Matrawy, and M. St-Hilaire, “Challenges of Task Offloading in Mobile Cloud Computing Environments”, A presentation (refereed on abstract) to the 56th Canadian Operational Research Society (CORS) Annual Conference, Ottawa, ON, Canada, May 2014

[14]. S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, R. Buyya, An autonomic cloud environment for hosting ecg data analysis services, Future Generation Computer Systems, 2012, pp. 147 154.

[15]. S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, R. Buyya, An autonomic cloud environment for hosting ecg data analysis services, Future Generation Computer Systems, 2012, pp. 147 154.

[16]. S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, R. Buyya, An autonomic cloud environment for hosting ecg data analysis services, Future Generation Computer Systems, 2012, pp. 147 154.

[17]. Dr. Jennifer S. Raj, Improved Response Time and Energy Management for Mobile Cloud Computing Using Computational Offloading, Journal of ISMAC (2020), http://irojournals.com/iroismac/ DOI: https://doi.org/10.36548/jismac.2020.1.004

[18]. Zheng, Jianchao, Yueming Cai, Yuan Wu, and Xuemin Shen. "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach." IEEE Transactions on Mobile Computing 18, no. 4 (2018): 771-786.

[19]. Chen, Xu, Lei Jiao, Wenzhong Li, and Xiaoming Fu. "Efficient multi-user computation offloading for mobile-edge cloud computing." IEEE/ACM Transactions on Networking 24, no. 5 (2015): 2795-2808.

[20]. Bestak, Robert, and S. Smys. "big data analytics for smart cloud-fog based applications." Journal of trends in Computer Science and Smart technology (TCSST) 1, no. 02 (2019): 74-83.

[21]. S. Srivastava and R. Kumar, "Indirect method to measure software quality using CK-OO suite," 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), 2013, pp. 47-51, doi: 10.1109/ISSP.2013.6526872.

[22]. Ram Kumar, Gunja Varshney , Tourism Crisis Evaluation Using Fuzzy Artificial Neural network, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-NCAI2011, June 2011

[23]. Ram Kumar, Jasvinder Pal Singh, Gaurav Srivastava, "A Survey Paper on Altered Fingerprint Identification & Classification" International Journal of Electronics Communication and Computer Engineering Volume 3, Issue 5, ISSN (Online): 2249–071X, ISSN (Print): 2278– 4209

[24]. Kumar, R., Singh, J.P., Srivastava, G. (2014). Altered Fingerprint Identification and Classification Using SP Detection and Fuzzy Classification. In: , et al. Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012. Advances in Intelligent Systems and Computing, vol 236. Springer, New Delhi. https://doi.org/10.1007/978-81-322-1602-5_139

[25]. Gite S.N, Dharmadhikari D.D, Ram Kumar," Educational Decision Making Based On GIS" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-1, Issue-1, April 2012.

[26]. Ram Kumar, Sarvesh Kumar, Kolte V. S.," A Model for Intrusion Detection Based on Undefined Distance", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1 Issue-5, November 2011

[27]. Rashidi, Shima, and Saeed Sharifian. "Cloudlet dynamic server selection policy for mobile task off-loading in mobile cloud computing using soft computing techniques." The Journal of Supercomputing 73, no. 9 (2017): 3796-3820

[28]. Rashidi, Shima, and Saeed Sharifian. "Cloudlet dynamic server selection policy for mobile task off-loading in mobile cloud computing using soft computing techniques." The Journal of Supercomputing 73, no. 9 (2017): 3796-3820