International Journal of Engineering & Technology, 7 (2.33) (2018) 722-725

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Computation offloading in smart mobile devices and comparison study on frameworks

SK. Piramu Preethika [1] *, Dr. R. Gobinath [2]

[1] Research Scholar, VISTAS, Pallavaram, Chennai, India
[2] Assistant Professor, VISTAS, Pallavaram, Chennai, India
**Corresponding author E-mail: preethikamanikandan@gmail.com.*

Abstract

Technology in the hands of young professionals leads to many superficial inventions among that smart mobile device are now with all human beings who may be in the village or in a modern town always busy looking at their mobile devices. People interacting with social media or many online games through smart mobile device leads the battery power down and slower the performance. Mobile Cloud Computation offloading is the best strategy to overcome this pitfall. Binding of Mobile devices to the powerful cloud with the side support from tele-communication is called Mobile cloud computing. Offloading takes the heavier computation to the cloud through the network and resolves the problem there and come back to smart mobile device. This paper compares and contrasts the features of offloading types and frame-works.

Keywords: *Smart Mobile Device (SMD) Mobile Cloud Computing; Offloading*.

1. Introduction

Mobile devices are becoming more powerful gadget and easy to use; they are replacing traditional Personal Computers for both personal use and work purpose. Rapid advancement in Wireless Mobile Networks (WMN) and mobile handsets facilitate enormous infrastructure and support a wide range of mobile persistence and services in addition to Mobile internet access. The user wants the service from anywhere and anytime without much effort.

Mobile devices are handy computing device which equipped with multiple sensors; they enable many useful applications which were unrealistic before. With the advancement in mobile hardware technology, evolution has been achieved from basic operations to complicated ones, ranging from three dimensional 3Dgames to image processing, speech recognition and bigger reality applications. Mobile applications are becoming energy hungry.

The statistical results of mobile phone users in India from 2013 to 2019 increased in millions. For 2017 the number of mobile phone users in India is expected to rise to 730.7 million. In this same year the number of SMD users in India is predicted to reach 340 million and could reach almost 468 million by 2021 [24].

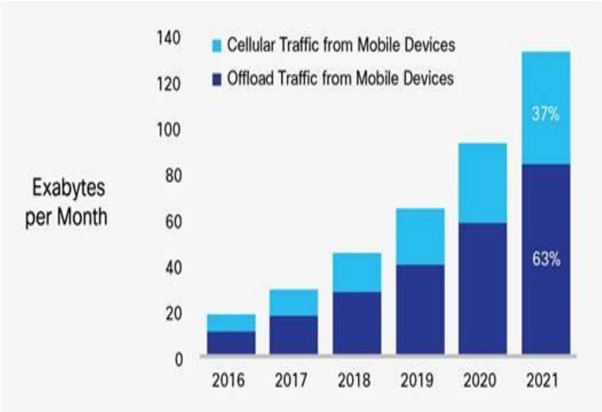


Fig. 1: Offload Traffic from Mobile Device.

Source: Cisco VNI Mobile, 2017

The amount of traffic offloaded from smart phones will be 64 percent by 2021. The amount of traffic offloaded from 4G was 63 percent at the end of 2016, and it will be 66 percent by 2021. The amount of traffic offloaded from 3G will be 55 percent by 2021, and the amount of traffic offloaded from 2G will be 69 percent. As 5G is being introduced, plans will be generous with data caps and speeds will be high enough to encourage traffic to stay on the mobile network instead of being offloaded, so the offload percentage will be less than 50 percent. As the 5G network matures, we may see higher offload rates [25].

Most of the people use SMD for E-Commerce, Online transactions which include net banking, travel transactions, online Advertisements, online bookings, bill payments, online shopping of all household items, online grocery, and online purchase from Amazon, Flip Cart. As the Smart phone users access the internet for various purposes the load balancing of SMD got affected. So the execution in SMD decreases the battery power. At this time the user needs an external support which reduces the work of mobile device here comes the cloud which supports the processing from outside the device on demand. This binding of Mobile Devices to the Cloud through the internet is called as Mobile Cloud Offloading which reduces the hunger of power consumption.

2. Literature review

2.1. Overview of computation offloading

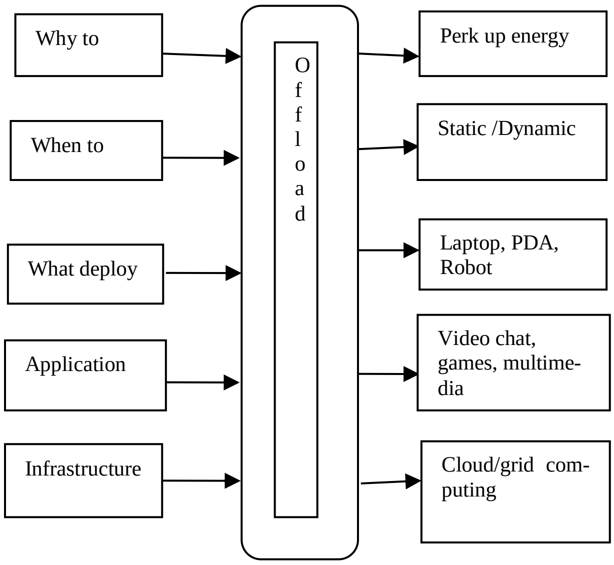


Fig. 2: Overview of Computation Offloading.

Researchers are concentrating on feasibility in late 90's, after that attention changed to designing algorithms for offloading decision whether it is needed or not. Further the direction moved towards Virtualization, bandwidth utilization, and cloud infrastructure. Now the offloading reaches its highs as unexpected [1].

Enablers of Offloading Wireless Mobile Network and Mobile agent, Virtualization and Cloud computing. A mobile agent is a program that can migrate from mobile to some other system like cloud within the network. Features of mobile agent reduce the network load, overcome network latency, adapt dynamically, encapsulate protocols, executes asynchronously and automatically. Cloud computing offers virtualization as a service on demand to the user.

2.2. Approaches of offloading

This approach is Fine-grained – applications only offload the subparts that benefit from remote execution, thereby it leads to large energy savings. For example, a video player decodes and plays video. Offloading its decoder which is the Energy-intensive part can reduce its energy consumption on the mobile device.

The second approach is to offload full process or full VM to the infrastructure. For example, in Android, each app runs in a process which runs in a separate VM. The downside of this approach is that the migration cost is high because all program states need to be transferred.

2.3. Parameter of offloading

User preferences are data security and constancy. Type of network used might be 3G, 4G, WIFI, and speed and bandwidth latency to be taken into account. In SMD its CPU type and speed, Memory size, storage, battery size, load capacity and location must be kept in mind. And in application side input size, average processing time, type of the application, untransferable parts are the metrics of offloading.

Parameter decides offloading is bandwidth and how much data to be exchanged through wireless mobile network, access speed of and memory storage of SMD, availability and execution cost of the cloud. Offloading must either reduce the cost or reduce the computational process or time consuming might be decreased according to the user requirements.

Table 1: Comparative Study of Frameworks Based on Metrics of Offload

Frame Work	Offload Type	Granularity	Partitioning	Metric
Spectra [14], [15]	Static	Fine Grained	Entire Application	Energy, Latency
Chroma [16], [17]	Static	Fine Grained	Entire Application	Latency
Maui [20]	Dynamic	Fine Grained	Application Level	Energy
Slingshot [19]	Static	Coarse Grained	Task Level	Latency
Scavenger [22]	Dynamic	Coarse Grained	Application Level	Energy
Clone Cloud [23]	Static	Coarse Grained	Entire Application	Energy

2.4. Goals of computation offloading

a) Enhance application performance
b) Reduce energy consumption for mobile device

2.4.1. Enhance application performance

For systems target rising application performance, computation offloading is an attractive solution for shortening response time of mobile applications as applications become more and more complicated.

- In context-aware computing, input data stream comes from various originations like GPS, sensors, maps, temperature need to be verified before they are utilized.
- A navigating app for a moving golem might have to acknowledge associate obstacle before the robot smash with the obstacle. Computation may have to be offloaded if the golem doesn't have quick enough processors.

To enhance the performance for mobile applications, we need to partition a program into two parts: 1) Execution on smart mobile device 2) The part to be offloaded. We can formulate the condition for offloading to maximize the performance as below. The speed of the mobile system be sm, w is the amount of computation for the second part is

$$\frac{w}{s_m} \quad (1)$$

To consider the execution time of the second part on the server, we ignore initial setup time of next work and the size of the program (server can download the program before offloading). If the second part is offloaded to a server, forwarding the input data di takes d_i/B seconds at bandwidth B. Offloading can improve application performance if and only if remote execution can be performed faster than local execution. Let ss be the speed of the server. The time to offload and run the second part is

$$\frac{d_i}{B} + \frac{w}{s_s} \quad (2)$$

We can see that computation offloading improve the performance of application when eq. 1 > eq.2

$$\frac{w}{s_m} > \frac{d_i}{B} + \frac{w}{s_s} \Rightarrow w \times \left(\frac{1}{s_m} - \frac{1}{s_s}\right) > \frac{d_i}{B} \quad (3)$$

Table 2: Measurements Taken from Frameworks for Mobile Cloud Offloading

S. No.	Year	Author	Topic	Measurements	Challenges
1	2017	Chen, Xing, et al [4]	Framework for context-aware computation offloading in mobile cloud computing	Execution time Power consumption	a)Reduces execution time by 6-96% b)Reduce power consumption by 60-96%
2	2016	Zhu, Chunsheng, et al.[5]	A novel sensory data processing framework to integrate sensor networks with mobile cloud	Traffic load Storage requirement of the sensor	a)data compressed to 70-75% b) to get better data performance a combination of attributes set.(e.g.)temperature and humidity threshold is .49
3	2015	Abolfazli S, Gani,et al[6]	A hybrid mobile cloud computing framework exploiting heterogeneous resources	Round trip time Energy efficiency	a)energy saved to 83-96% b)using round trip time performance evaluation gains is 80%-96%
4	2015	Shiraz, Muhammad, et al [7]	Energy Efficient Computational Offloading Framework for Mobile Cloud Computing	energy consumption cost	a)data transmission reduced by 84% b)energy consumption cost reduced to 69.9%
5	2015	Shiraz, Muhammad, et al [8]	A study on the critical analysis of computational offloading frameworks for mobile cloud computing	Turnaround time Energy consumption cost Data transmission cost	a)additional energy consumed 31.6 % b)39% additional time is taken and 13241.2 KB data is transmitted in offloading
6	2015	Mora Mora, Higinio, et al [9]	Flexible framework for real-time embedded systems based on mobile cloud computing paradigm	Throughput Delay	A computational model based on imprecise computation method is proposed a)utilization of cloud is 25%
7	2014	Abolfazli, Saeid, et al [10]	Rmcc: Restful mobile cloud computing framework for exploiting adjacent service-based mobile cloudlets	Mean time Energy saving	a)Rmcc conserves energy by 71.45% b)time conserved by 87%
8	2014	Shiraz M, Gani A [11]	A lightweight active service migration framework for computational offloading in mobile cloud computing	9	2014 Shiraz, Muhammad, et al [12] A lightweight distributed framework for computational offloading in mobile cloud computing Turnaround time Size of data transmission a)data transmission reduced by 91% b)energy consumption cost minimized by 81% c)turnaround time decreased by 83.5%

Descriptions of the variables explained below:

- Heavy computation indicated as - large w
- When the server is fast - large ss
- If the size of data transfer is small - di
- Bandwidth of network is high - large B

This inequality also shows that if $\frac{w}{s_m} < \frac{d_i}{B}$, performance can't be improved by offloading, even if the server is infinitely fast (i.e., ss→∞). Hence, when analyzing the program to be offloaded, only tasks that require heavy computation (large w) with light data exchange (small di) should be considered.

Moreover, if we define $w\ (\frac{1}{s_m} - \frac{1}{s_s}) - \frac{d_i}{B}$ as the performance gain of offloading, the server's speed has diminishing return: doubling ss will not double the gain.

2.4.2. Reduce energy consumption

Some computation offloading system helps reduce the energy consumption of the mobile device by delegating energy intensive computation to remote server. The following analysis explains which task is suitable for computation offloading. Let pm be the power of the mobile device, the energy cost of executing the task is

$$p_m \times \frac{w}{s_m} \quad (4)$$

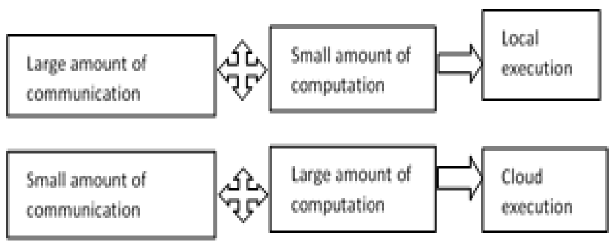
Let pc be the power of network interface of mobile device in working state, pi is the power in idle state. So the energy consumed by network interface of the mobile device

$$p_c \times \frac{d_i}{B} + p_i \times \frac{w}{s_s} \quad (5)$$

Computation offloading saves energy when eq. 4 > eq. 5, such that [1]

$$p_m \times \frac{w}{s_m} > p_c \times \frac{d_i}{B} + p_i \times \frac{w}{s_s} \Rightarrow w \times \left(\frac{p_m}{s_m} - \frac{p_i}{s_s}\right) > p_c \times \frac{d_i}{B} \quad (6)$$

From eq. 6, we can see that tasks with heavy computation (large w) and light data size (small di) are suitable for computation offloading.



By this study, whether offloading decision to cloud necessary or not implied from the above constraint.

3. Existing works

3.1. Frameworks of mobile cloud computation offload

The above table describes the list of measurements taken from frameworks for mobile cloud computing. These works specifies the methods and techniques used to offload the data from SMD to cloud. The table also mentions how different Authors used various frameworks to solve the energy constraint issues.

4. Conclusion

In this paper, we concentrates on the offloading techniques, parameters, types, goals .Further the study extends the measurements of mobile cloud offloading frameworks. We describe why computation offloading can become progressively vital for resource strained devices within the future.

References

[1] Othman, Mazliza, Abdul Nasir Khan, Shahbaz Akhtar Abid, and Sajjad Ahmad Madani. "MobiByte: an application development model for mobile cloud computing." Journal of Grid Computing 13, no. 4 (2015): 605-628.

[2] Gupta, Abhishek Kumar. "Challenges in mobile computing." In Proceedings of second National Conference on Challenges and Opportunities in Information Technology (COIT-2008). Mandi Gobindgarh, India: RIMT-IET, pp. 86-90. 2008.

[3] Kumar, Karthik, et al. "A survey of computation offloading for mobile systems." Mobile Networks and Applications 18.1 (2013): 129-140.

[4] Chen, Xing, et al. "Framework for context-aware computation offloading in mobile cloud computing." Journal of Cloud Computing 6.1 (2017) 1.

[5] Zhu, Chunsheng, and et al. "A novel sensory data processing framework to integrate sensor networks with mobile cloud." IEEE Systems Journal 10.3 (2016): 1125-1136.

[6] Abolfazli, Saeid, Abdullah Gani, and Min Chen. "Hmcc: A hybrid mobile cloud computing framework exploiting heterogeneous resources." Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 third IEEE International Conference on. IEEE, 2015.

[7] Mora Mora, Higinio, et al. "Flexible framework for real-time embedded systems based on mobile cloud computing paradigm." Mobile Information Systems 2015.

[8] Shiraz, Muhammad, et al. "Energy Efficient Computational Offloading Framework for Mobile Cloud Computing." J. Grid Comput. 13.1 (2015): 1-18.

[9] Shiraz, Muhammad, et al. "A study on the critical analysis of computational offloading frameworks for mobile cloud computing." Journal of Network and Computer Applications 47 (2015): 47-60.

[10] Abolfazli, Saeid, et al. "Rmcc: Restful mobile cloud computing framework for exploiting adjacent service-based mobile cloudlets." Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on. IEEE, 2014.

[11] Shiraz, Muhammad, and Abdullah Gani. "A lightweight active service migration framework for computational offloading in mobile cloud computing." The Journal of Supercomputing 68.2 (2014): 978-995.

[12] Shiraz, Muhammad, Abdullah Gani, Raja Wasim Ahmad, Syed Adeel Ali Shah, Ahmad Karim, and Zulkanain Abdul Rahman. "A lightweight distributed framework for computational offloading in mobile cloud computing." PloS one9, no. 8 (2014): e102270.

[13] Sharifi, Mohsen, Somayeh Kafaie, and Omid Kashefi. "A survey and taxonomy of cyber foraging of mobile devices." IEEE Communications Surveys & Tutorials 14.4 (2012): 1232-1243.

[14] J. Flinn, S. Park, and M. Satyanarayanan, “Balancing performance, energy, and quality in pervasive computing,” in 22nd International Conference on Distributed Computing Systems (ICDCS02), Vienna, Austria, 2002, pp. 217–226.

[15] J. Flinn, D. Narayanan, and M. Satyanarayanan, “Self-tuned remote execution for pervasive computing,” in eighth IEEE Workshop Hot Topics in Operating Systems, Schloss Elmau, Germany, 2001, pp. 61–66.

[16] R. K. Balan, D. Gergle, M. Satyanarayanan, and J. Herbsleb, “Simplifying cyber foraging for mobile devices,” in 5th USENIX International Conference on Mobile Systems, Applications and Services (MobiSys), San Juan, Puerto Rico, 2007, pp. 272–285.

[17] R. K. Balan, M. Satyanarayanan, S. Park, and T. Okoshi, “Tactics-based remote execution formobile computing,” in 1stInternational Conference on Mobile Systems, Applications and Services, San Francisco, 2003, pp. 273–286.

[18] S. Goyal and J. Carter, “A lightweight secure cyber foraging infrastructure for resource-constrained devices,” in 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA ’04), English Lake District, UK, 2004, pp. 186–195.

[19] Y. Y. Su and J. Flinn, “Slingshot: Deploying stateful services in wireless hotspots,” in 3rd International Conference on Mobile Systems, Applications, and Services, New York, NY, USA, 2005, pp. 79–92.

[20] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “MAUI: Making smartphones last longer with code offload,” in eighth international conference on Mobile systems, applications, and services (ACM MobiSys’10), San Francisco, USA, 2010, pp. 49–62.

[21] M. D. Kristensen, “Scavenger: Transparent development of efficient cyber foraging applications,” in IEEE International Conference on Pervasive Computing and Communications (PerCom), Mannheim, Germany, 2010, pp. 217–226.

[22] M. D. Kristensen and N. O. Bouvin, “Scheduling and development support in the scavenger cyber foraging system,” Pervasive and Mobile Computing, vol. 1, no. 6, pp. 677–692, 2010.

[23] B. Chun and P. Maniatis, “Augmented smartphone applications through clone cloud execution,” in Proc. 8th Workshop on Hot Topics in Operating Systems (HotOS), Monte Verita, Switzerland, 2009.

[24] http://www.statista.com.

[25] www.cisco.com.