



Study and Challenges of Reckonable Offloading in Mobile Cloud Computing

Mrs. Jyotsna P. Gabhane¹, Dr. M.U.Kharat²

¹Assistant Professor, ² Professor & Head

¹Department of Computer Technology, Priyadarshini College of Engineering Nagpur, India

²Department of Computer Engineering MET~BKC, Institute of Engineering, Nashik, India

Abstract: An alluring paradigm for mobile users to enjoy the computation power and service of a network is Mobile cloud computing. A remote server is provided as cloud. As mobile resources have limited resources like memory, power, availability of network, so cloud concept is cooperative flair. The reckoning tasks that are more memory and battery consuming on a mobile are uploaded on to a remote server is called as offloading. Offloading has been vast research area in the past decade. Still many perceptive issues related to offloading are needs to investigate. This survey paper provides background concept of offloading. The systems like CloneCloud, MAUI which has been specially designed for cloud computing are discussed along with their advantages and limitations. We also describe the parameters like energy, security, bandwidth, delay, cost related with the offloading in MCC. We also highlighted directions for future research & challenges.

Keywords: Mobile Cloud Computing (MCC), Reckonable offloading, energy

I. INTRODUCTION

Mobile cloud computing (MCC) is an infrastructure where both the data storage and processing executes outside the mobile device. Mobile cloud applications migrates the computing application away from mobile phones and into the cloud [9]. Battery technology is one of the major hurdles for potential growth of Smartphone's. MCC provides mobile users processing of methods and storage in clouds. As all the complicated computing tasks are migrated into the cloud the mobile devices do not need a powerful configuration. Fast development in processors makes devices equipped with more sensors. No of sensors with more reckonable tasks compose the extra power utilization. Battery down is now remaining the primary bottleneck for all mobile devices [8]. So in the path of finding the solution, uses of remote server for all battery consuming task is introduced called as mobile cloud computing.

The reckoning tasks that are more memory and battery consuming on a mobile are uploaded on to a remote server is called as offloading. Some time consuming tasks can be send to cloud means a resourceful server and accepting the result from it once execution is over. The mobile systems have some limitations such as communication, network bandwidth, processor performance, storage capacity and latency. Offloading decreases the power consumption of the mobile device improving the battery life and duty for longer use. In offloading computational applications are spited and the parts offloaded. It speeds up the application thereby improving performance of the mobile device. The infrastructure needed for offloading is grid or cloud computing. Concept of virtualization is implemented to offer computing as a service. A user can "lease" computing resources based on their requirement. Since offloading migrates computation to a more resourceful computer it involves making a decision regarding "whether" and "what" computation to be migrated. The decision of offloading depends on the task static or dynamic.

The computational intensive applications include context based video analysis, 3D modeling, image retrieval, voice reorganization, gaming, GPS and navigation. Offloading becomes an attractive solution for meeting response time requirements on mobile system; however the goal is to overcome real time constraints.

Example: A navigation robot may need to recognize an object before it collides with the object and if robot's processor is too slow the computation may need to be offloaded.

Another application of offloading is context-aware computing, example; where, multiple streams of data from source like GPS, maps, accelerometers, temperature sensors, etc., needs to be analyzed together in order to obtain real-time information about user's context. The mobile systems that require using offloading are laptops, PDA's, tablets, robots and application sensors.

A. Offloading decisions are of basically two types:

- Full Offloading: all reckoning tasks are moved from mobile device to the remote server i.e. to cloud.
- Partial Offloading: application is partitioned into several tasks and most computation intensive task is loaded onto server.

II. LITERATURE SURVEY OF RECKONABLE OFFLOADING DECISIONS

A. Overview of Energy Saving Model:

Energy is a crucial restraint for mobile devices. Now a day's all Smartphone are used only for voice communication, downloading and watching videos, playing games, chatting and many other purposes. As a

result, these Mobile devices are likely to consume more power and shorten the battery life. Battery technology are playing vital role to improve the performance but it is not been able to keep up with the rapid growth of power consumption of such mobile devices. So the fruitful solution is offloading which extend battery life by migrating the energy-intensive parts of the computation to remote servers.

1) Energy analysis for computation offloading [4]:

Suppose,

C: No. of instructions required by computation

S: Speed of instruction /sec of cloud server

M: Speed of instruction/sec of Mobile Device

The task execution on server takes

C/S sec

And on mobile same task executed in

C/M sec

D: is the exchange of data in bytes between mobile & server

B: Network Bandwidth

So it takes D/B sec to transmit & receive data from MD to cloud server.

MD consumes power like:

Pc: for computing

Pi: for being idle

Ptr: Transmitting & receiving power of MD

If mobile perform computation then

$$\text{Energy consumption} = \left[P_c * \left(\frac{C}{M} \right) \right] \quad (1)$$

If server perform computation

$$\text{Energy consumption} = \left[P_t * \left(\frac{C}{S} \right) + P_{tr} * \left(\frac{D}{B} \right) \right] \quad (2)$$

The amount of energy saved is

C= equation (1) – equation (2)

If the above formula produces positive result then only energy gets saved [4].

Offloading is beneficial only when the amount of computation C needed with relatively small amount of computation D.

E.g. Playing Chess

Mobile Image Processing

2) Challenges:

a) *Privacy and security issues*: Shifting all data to the cloud needs more security. Service providers to the cloud works with many third-party vendors, and there is no guarantee as to how these vendors protect the data. Storage locations are typically unknown to the user; it is difficult to determine what laws apply to protect data. Sometimes users are not ready to stored data in the cloud without considering privacy and security implications. Encryption is the possible solution for such storage. However, encryption will not solve the complete problem. Steganography might be the good solution for the privacy purpose as mentioned in [4]. Still research on this matter is going on.

b) *Unreliability in terms of availability of network* : An impending anxiety with mobile cloud computing is its matter of reliability. A mobile user who wants to perform computation in the cloud completely depends on the wireless network and cloud services. Such dependence on the wireless network implies that connectivity must be available to all the users when required. Suppose a user in a basement of a shopping mall and want to share videos, images or want to do chat then connectivity should not be limited. Such issues are challenging in the development of framework for computational offloading.

c) *Processing the real-time data on the mobile system*: Some real time applications like chess, context aware applications, image processing, mobile surveillance have real-time data. In such scenarios, *D* in Equation B is no longer a pointer to the data but it act as the actual data. When the value of *D* is large, offloading cannot save energy. Performing the computation in such cases on the mobile system may be more power efficient. So better to go with partitioning of task and send it to cloud to reduce energy consumption. Such solution provides partial real time data on mobile.

B. Overview of CloneCloud System:

Clone Cloud [6], a system that transforms mobile applications to the cloud to utilize the benefit of it. In clone cloud right portion of the computation get passed on to cloud. A primary objective of clone cloud is “what to run where”. An additional aim of clone cloud is to keep programmer away from application partitioning. The implementation has been done in application layer where the individual thread migrates from mobile to remote

server. After execution of task thread returns back to mobile device along with the output which merge back to the original process.

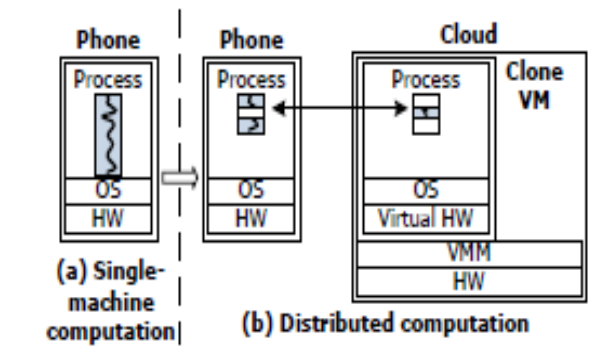


Figure 1. CloneCloud system Model

Conceptually, this system automatically transforms a single-machine execution like computation on a Smartphone into a distributed execution optimized for the network connection to the cloud as shown in figure 1[6]. This paper applies primarily to application layer virtual machines (VMs), such as the Java VM, DalvikVM from the Android Platform, and Microsoft's .NET. Application-layer VMs are widely used in mobile platforms. And the application-layer VM model has the relative ease of manipulating application executables and migrating pieces. VM runtime executes byte codes of input method in the thread. Partitioning framework here combines static program analysis with dynamic program profiling to construct a partition. The Static Analyzer recognize authorized partitions of the application and the Dynamic Profiler profiles the input executable on different platforms with a different set of inputs, and returns a set of executions.

Each thread has its own VM stack, program counter as well native stack. Execution offloading involves two tasks code partitioning and state migration. Partitioning involves by static or dynamic way. In static partitioning task assigned to each machine is fixed so at compilation time speed of processor, power and storage remained constant [5]. In dynamic way code is represented to the cloud for execution. In this paper all stacks and heap objects are migrated on cloud. Here heap is data structure which holds all dynamically allocated data. Static analyzer works for clone cloud based on three legal partition properties [6].

1) Three legal partition properties:

a) A process that access specific features of a device must be attached to the device.

For example,

If a process uses a GPS, Accelerometer or sensor like Proximity Sensor in a mobile device then such method must execute on mobile.

b) Processes that share local state must be collocated at the same device.

For example,

In image processing initialization, detection, fetching methods can access local states then its need to be collocated at the same device.

c) Nested migration must be prevented

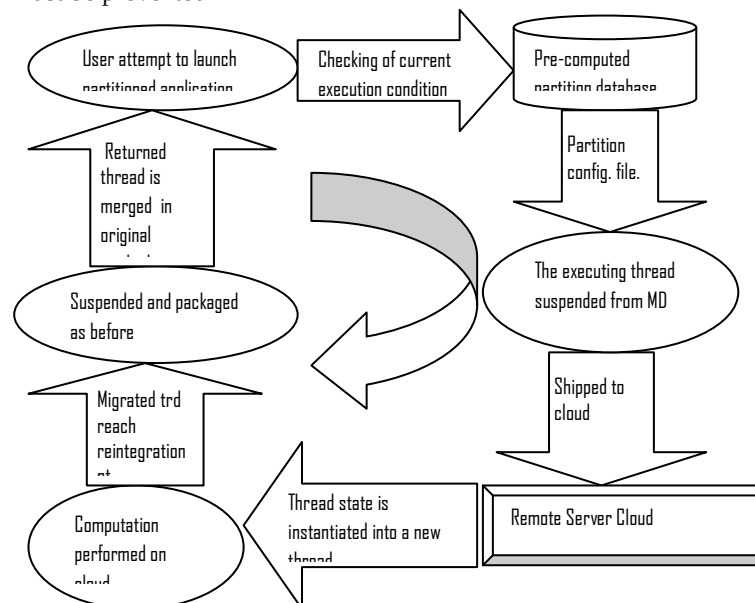


Figure 2. The life cycle of partitioned application

For a one mobile and one cloud there should be no nested resumes and suspends. Once program is suspended for migration at entry point of a method the program should not be suspend again without resumption. Multiple execution of the application on dynamic profiler used to construct a cost model. This profiler measures expected execution time and energy uses of all methods on the device as well as on server. A profile tree is a tree with one node for each method invocation in the execution and it is rooted at the user-defined method invocation of the application like main.

Edge represents method calls in the execution. Each node is interpreted with the cost of its particular invocation in the cost metric. Residual node is one when non-leaf node also has a leaf child. The residual node for node i represents the residual cost of invocation i that is not due to the calls invoked within i ; in other words, node i represents the cost of running the body of code excluding the costs of the methods called by it.

Distributed implementation method in CloneCloud is to perform a precise partition of an application process. The life cycle of application partitioning is as shown in figure 2.

2) Clonecloud Limitations:

- a) Inability to migrate native state & to export unique native resources remotely.
- b) Non virtualized resources cannot get accessed to native resources and are also not available on clone.
- c) Full concurrency between MD and remote server is not get supported by clonecloud system.

C. Overview of MAUI System:

MAUI, a system that enables fine grained Energy-aware offloads of mobile code to the infrastructure. In above discussion it has been mentioned that problem relied heavily on programmer support for partition an application, or sometime full process need to be migrated. MAUI managed code environment to suggest the best of both problem initially MAUI system maximize energy savings by offloading fine-grained code with minimal burden on the programmer and it also decides at runtime which methods should be remotely executed.

A programming environment provided by system where developers interpret which process of an application can be offloaded on cloud. Each time when a method is call upon and a remote server is available, MAUI uses its optimization framework for deciding whether process should be offloaded or not. MAUI implements each method of an application to find the cost of offloading it, i.e. the total states that needs to be transferred for remote execution. MAUI also continuously measures the network connectivity to the infrastructure for estimate its bandwidth as well as latency. All such variables are helpful to plan an optimization problem whose solution states which methods should be offloaded to the remote server and which should execute locally on the devices.

The MAUI system also follows certain rules to offload methods on to server.

1) Following are certain types of code that should not be marked send to the remote server

- a) code with the application's user interface
- b) code which interacts with I/O devices
- c) Code interacting with any outdoor component [7].

2) Offload decisions in MAUI depends on three factors:

- a) Energy consumption characteristics of mobile device
- b) The execution time and resource needs of individual methods
- c) The characteristics of the wireless network [7].

The MAUI profiler measures the characteristics of mobile device at initialization time, and it keeps monitoring the program and frequently changing wireless network characteristics and a fusty measurement may force MAUI to make the wrong decision of offloaded. The next MAUI solver uses data collected by the MAUI profiler as input to a global for optimization problem that determines which methods should execute locally and which should execute remotely. Here the solver's aim is to find a program separation policy that decreases the energy consumption.

3) Application of MAUI System:

MAUI enables: 1) face recognition application that consumes less energy, 2) a latency-sensitive arcade game application that doubles its refresh rate, and 3) a translation application which is voice-based language [7].

III. OFFLOADING IN THE FUTURE SCOPE & RESEARCH DIRECTION

Several studies contribute to the development of MCC by deal with issues as presented in the above section.

There are still some areas which need to be focused. This section we talk about several open issues and probable research guidelines in the development of offloading framework for MCC. Sensors are crucial part of mobiles, now days. These Sensors have limited computing ability. When cameras, microphones, GPS, and many other types of sensors are connected to the device the amount of data produce is overwhelming. Mobile phones, tablets, laptops, and sensors—grow, the demand for increased functionalities will continue. It is becomes difficult to offer long battery lifetimes. Computation Offloading is a natural way out to this problem. The bandwidth limitation is again another concern [9] because the number of mobile and cloud users are increasing. Fourth generation (4G) network and Femto cell are emerging as promising technologies that overcome the limitation of bandwidth [10]. Fourth generation network is a technology that significantly increases bandwidth

capacity for subscribers. Furthermore, 4G networks also promises other advantages such as wider mobile coverage area, smothering quicker handoff, varied services, and so on. An efficient network improves link performance for mobile users and optimizes bandwidth usage. Cognitive radio is a one solution to achieve the wireless access management in mobile communication environment. This increases the efficiency of the spectrum utilization by allowing unlicensed users for accessing the spectrum allocated to the licensed users. Therefore, mobile users in MCC must be able to detect this radio resource availability while ensuring that the traditional services will not be interfered [9]. The users face some problems in offloading such as congestion in the network to the limitation of wireless bandwidths, unavailability of network, and the signal reduction. The problem of delay again occurs when the users want to communicate with the cloud, so QoS is reduced significantly. Two areas CloneCloud [6] and Cloudlets [3] that are expected to reduce the network delay. CloneCloud however have inability to migrate native state & to export unique native resources remotely. Non virtualized resources cannot get accessed to native resources and are also not available on clone. Full concurrency between MD and remote server is not get supported by clone cloud system. A cloudlet is well connected to the Internet and available for use by nearby mobile devices. Thus, when mobile devices do not want to offload to the cloud they can find and use a nearby cloudlet. Real-time interactive response increases by low-latency and high-bandwidth. However, there are some considerations that need to be addressed For example, how to distribute processing, storage, and networking capacity for each cloudlet? How to manage policies for cloudlet providers to maximize user experience while minimizing cost? Also, trust and security for cloudlet are other issues in implementing this idea because adversaries can create a fake cloudlet to steal the user's information. Both mobile service provider (MSP) and cloud service provider (CSP) work with MCC. Both are having different services management, customer's management, way of imbursement, and cost. So how to set cost, how the price segregate among different units and how the customers recompense all these factors are challenging. The web interfaces responsible to form interoperation between user & the cloud. Web interfaces is not specifically designed for mobile devices so it may have more overhead. Again compatibility among devices for web interface is a challenging issue. The services provided by CSPs will be differentiated as cost, availability and quality of network so again more development is needed in this area. The expansion of MCC in the Heterogeneous network is a challenge. Heterogeneous Network (HetNet), has nodes which are having multiple types of low power radio access with macrocell nodes in a wireless network, is widely accepted way to satisfy the traffic demand. Along with the rapid development of HetNet, the cloud services are also used rapidly by mobile users. However, several challenges have to be tackled before the users can actually benefit from the MCC applications. Research on HetNet for MCC will be conducted based on a tight coupling of the unique characteristics of the MCC applications and the wireless heterogeneous networks. The calculation offloading judgment should be made base on the power and QoS from the HetNet. The wireless HetNet, should apply the related radio resource management functions [2].

IV. CONCLUSION

This paper study and organize a research related with reckonable offloading for mobile systems. We observe how Computation Offloading is a natural way out to offer long battery lifetimes. Clonecloud makes seamless interfacing between the mobile and the cloud. Here partitioning, migration with merging, and on-demand instantiation of partitioning are initiative process. CloneCloud however have inability to transfer native state & to export unique native resources remotely. In this paper studied MAUI, a system that enables fine-grained energy-aware offloading. MAUI reduces the burden on programmers with provision of maximizing the energy benefits of offloading code. Finally we describe the importance of computation offloading and its future research direction.

REFERENCES

- [1] Karthik kumar · jibang liu · yung-hsiang lu · bharat bhargava, "a survey of computation offloading for mobile systems", in Springer Science ,Business Media, LLC 2012.
- [2] Lei Lei And Zhangdui Zhong, Kan Zheng, Jiadi Chen, And Hanlin Meng, "Challenges On Wireless Heterogeneous Networks For Mobile Cloud Computing", IEEE Wireless Communications • June 2013,pp 34-44.
- [3] Satyanarayanan M, Bahl P, Caceres R, Davies N. The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing* 2009; 8(4): 14–23.
- [4] K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" *Computer*, vol. 43, no. 4, Apr. 2010, pp. 51–56.
- [5] Seungjun Yang, Yongin Kwon, Yeongpil Cho, Hayoon Yi, Donghyun Kwon, Jonghee Youn, and Yunheung Paek, "Fast Dynamic Execution Offloading for Efficient Mobile Cloud Computing", IEEE International Conference on Pervasive Computing and Communication(PerCom), 2013,pp 20-28.
- [6] B. G. Chun *et al.*, "CloneCloud: Elastic Execution Between Mobile Device and Cloud," *Proc. 6th Conf.Computer Systems (EuroSys)*, Apr. 2011, pp. 301–14.
- [7] E. Cuervo *et al.*, "MAUI: Making Smartphones Last Longer with Code Offload," *Proc. 2010 Int'l. Conf. Mobile Syst., App., Services*, 2010, pp. 49–62
- [8] R. A. Powers., "Batteries for low power electronics". Proceedings of the IEEE april 1995.pp 687–693.
- [9] H. T. Dinh *et al.*, "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches," *Wireless Commun. and Mobile Computing*, Oct 2011.
- [10] Rahman M, Mir FAM. Fourth Generation (4G) mobile networks - features,technologies & issues, In *Proceedings of the 6th IEEE International Conference on 3G and Beyond*, 2007;