Offloading Schemes in Mobile Cloud

Monika Dudeja
M.Tech Scholar,
Dept. of CSE, MRIU

Kritika Soni
Associate Professor,
Dept. of CSE, MRIU

ABSTRACT

Cloud computing is one of the emerging technology which is finding its usage everywhere. Mobile cloud is one step ahead of Cloud computing. In this paper, we will give a brief an overview of Mobile cloud computing and will discuss some of the application partition schemes for offloading in mobile cloud .The paper also discusses what kind of applications are suitable for offloading and also indicates towards future research directions for offloading.

General Terms

Mobile Cloud, Offloading, Application Partitioning

Keywords

Overview of Mobile Cloud, Offloading, Cloud Computing

1. INTRODUCTION

Mobile cloud Computing can be defined as a technology where the computational intensive tasks from a mobile device are offloaded to nearby cloud to save battery and CPU usage. The Cloud provides the required storage space as well as other resources which are deficient in a mobile device. A mobile device here can be anything which is mobile i.e. a cellphone, or a laptop or a tablet etc. Mobile Cloud computing is finding its use everywhere and there are variety of applications which are based out of this technology

2. ARCHITECTURE OF MOBILE CLOUD COMPUTING

The generic architecture as defined by Dinh et all[1] basically consists of a mobile device connected to mobile networks, The mobile networks then transfer the request from mobile devices to cloud for processing via internet. Figure 1 explains the basic architecture for Mobile cloud.

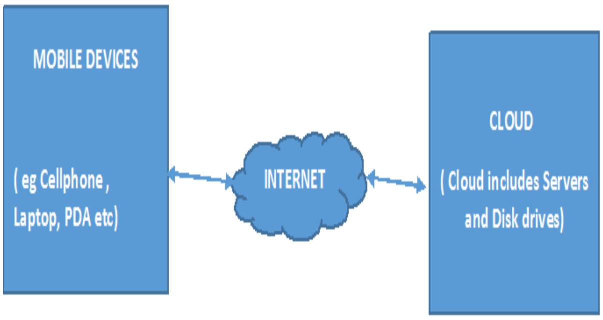


Figure 1: Basic Architecture of Mobile Cloud

3. ADVANTAGES OF MOBILE CLOUD COMPUTING

1. Reduces Battery Consumption: As most of the processor intensive task are offloaded to cloud, it hence can save considerable amount of energy by saving battery life.
2. Provides more data storage. Cloud can offer data storage and mobile devices can use this storage space for storing the data. Amazon Simple Storage Service (Amazon S3)[8] offers such kind of service by providing a database for use.
3. Improves Reliability: The data can be stored and backups are created on multiple servers, hence ensuring that data is not prone to losses, even if the mobile device is stolen..
4. Privacy: By encrypting the data stored on the cloud the privacy of the user is maintained.

4. DISADVANTAGES OF MOBILE CLOUD COMPUTING

1. Low Bandwidth:. Bandwidth and lack of signal or signal errors is one of the main reason that Mobile cloud computing is not becoming as useful for the users as it should be. Some alternate solutions need to be found out to overcome bandwidth limitations
2. Heterogeneous Networks: There are various type of technology used in mobile devices like GSM, CDMA, WiMAX and handling such heterogeneous network is quite a difficult task.
3. Interoperability between Platforms; Different mobile devices use different Operating system for example Phones based out of Android technology like HTC use Android Operating system. Symbian is used by Nokia, Apple Phones use IPhone IOS, hence creating a mobile application which can work in all of those is quite a difficult task.
4. Security and Data Privacy: It is essential to save data during transit as well as data when stored on the cloud servers. Several security attacks and threats on wireless network need to be taken care of.
5. Data Access: It is essential to have an efficient mechanism of data access, which is cost effective as well as saves the communication overhead.
6. Computational Offloading: A tradeoff between communication and calculation costs has to be evaluated before deciding to offload any mobile application and it's a complex process which depend on several entities.
7. Pricing: The pricing model in case of mobile cloud has to be decided among the three stakeholders ie application license provider, mobile network provider and cloud service provider and there could be clashes among 3.

5. OFFLOADING MECHANISM

Offloading in its simple terms can be defined as the mechanism of partitioning an application into offloadable and non offloadable sections considering various parameters and then remotely executing the offlloadble sections. There is no set of guidelines for deciding which section of an application to offload but we can keep in mind the below mentioned points. Figure 2 represents a basic Flow chart for Offloading Process. Offloading can only be triggered considering two things i.e. the benefits the user will get and the availability of resources.

5.1 Good Candidates for Offloading

1. Games like chess and Sudoku Solver which need quick computation and have numerous small datasets
2. File Search applications which takes more than few minutes to search a particular file
3. Image Processing tasks and image rendering tasks are quite computational intensive
4. .Download Applications can save time and energy when offloaded to cloud.
5. Antivirus Applications can be offloaded to cloud since it involves a complete scan of phone and comparison of several virus signatures.

5.2 Bad Candidates for Offloading

1. User interaction related sections as well as interfaces interacting with camera or GPS should not be offloaded because it will increase the communication overhead.
2. Sections which directly access I/o Devices
3. Some native methods (for eg methods declared static in java)
4. Methods accessing Device specific properties

5.3 Entities Effecting Offloading

As per Khan [10] the decision for offloading is a complex process. The following are the entities which impact that decision.

1. Capability of smartphone: If a smartphone has a powerful processor comparable to a server then offload process will never be triggered.
2. Nature of Application: An application requiring native resources will never be fully offloaded.
3. Connection: A 3G connection vs a Wi-Fi connection have their own set of advantages and disadvantages, hence depending upon the connection a user has user may decide not to offload , for eg 3G connection is more expensive than Wi-Fi , so if a user has 3G connection, he may decide not to choose offloading due to cost of bandwidth.
4. Cloud Provider: The Cloud provided should have enough resources as required by mobile application, otherwise user will not get a benefit out of offloading to cloud.

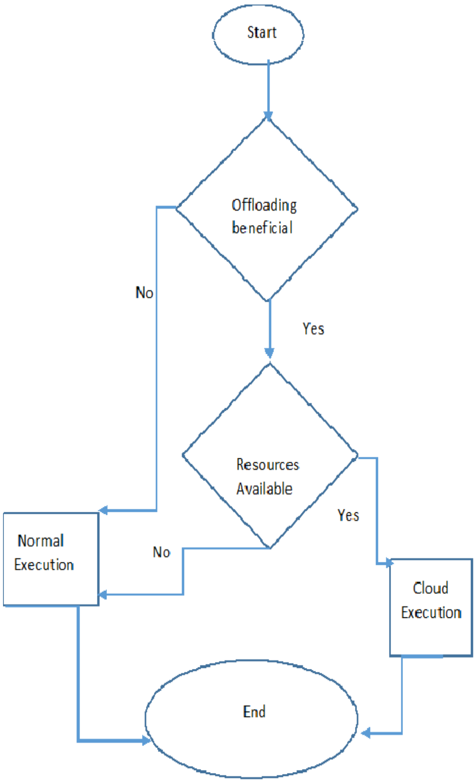


Figure 2 :Flowchart for Offloading Process

5.4 Application partition schemes

Application partitioning is the first step towards offloading process. It can mainly be categorized as static and dynamic.

Static partitioning deals with dividing the application into partitions during the design time. Li Z. et al. [4] have defined a static partitioning scheme by considering offloading at the level of procedure calls (tasks). For a given program, they have profiled the information about computation time and data sharing, then constructed a cost graph and have proposed a task mapping algorithm to statically divide the program into server tasks and client tasks. After constructing the cost graph, they have applied the branch-and-bound algorithm to the cost graph to minimize the total energy consumption of computation and the total data communication cost.

Dynamic Partitioning considers the run time parameters which play an important role in determining resource consumption in pervasive environment. Ou et al [5] have proposed a multiconstraint (k+1) Partitioning scheme, to partition an application into k offloadable and one unoffloadable components. They have created a cost graph by considering the metrics like communication cost, CPU cost and memory cost for each of the node of graph .The run time parameters which are considered are obtained by taking repeated screenshots of resource consumption by each of the node and any increase or decrease in resource consumption is captured or taken into consideration. Another dynamic scheme is adapted in Maui [7] , here the programmer first initially marks initial partition of the application using annotations. Then three kind of profilers i.e. network profilers, device profilers, program profilers are used to evaluate the application performance and energy consumption and then a Maui solver decides which method to offload. The Problem of manual partitioning is that it is an additional overhead on programmers and cannot adapt to changes in the environment. Hence Automatic partitioning Techniques like Spectra [11] have come into existence. Recently several infrastructures like Clonecloud [9] have emerged which migrate the entire OS and its respective applications to cloud by creating a clone. A ‘Clone‘is a mirror image of a Smartphone running on a virtual machine. By contrast with smart phones, such a ‘Clone‘ has more hardware, software, network, energy resources in a virtual machine which provides more suitable environment to process complicated tasks,

6. EXPERIMENTAL ANALYSIS

To understand whether Offloading can save energy, an Android application was created for Sudoku Solver puzzle game and installed on a mobile device. Next a web application using Net Beans version 7.4 was developed which was installed on laptop to act as a Server. Both versions of the application has used the same recursive backtracking algorithm for solving the puzzle which is an existing algorithm. The mobile device used for the setup is HTC explorer having Android version 2.3 and the Server is a laptop having Core I3 processor. The Android application takes the input from the user in the form of puzzle and sends one copy of input to the local mobile device and another copy of input to Server .The local versus Server processing time were calculated and displayed on the mobile device. Local Processing time(TLocal) is the time taken to solve the puzzle when the application runs locally on the mobile device while the Server processing time(TServer) is the sum of time taken by the server to parse the input coming from mobile device(TInput) ,time taken to compute the result(TProcessing) and time taken to deliver the output back to mobile device(TOutput.Hence we can write the expression for Local Processing time(TLocal)and Server processing time(TServer).

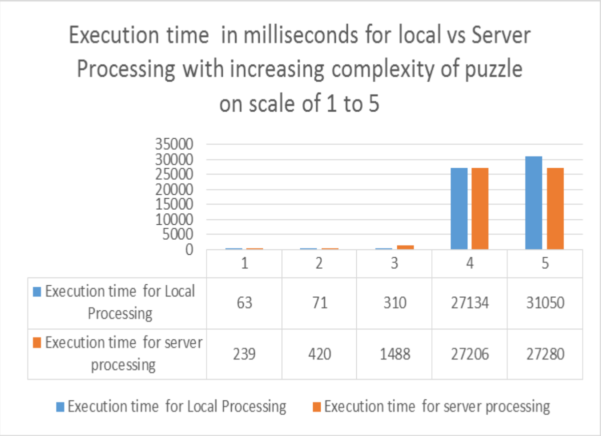
TLocal =TProcessing (ignore TInput and TOutput since it will be negligible when application executes locally}

TServer= TInput+ TProcessing+ TOutput

For offloading to be beneficial **TServer<=T Local**

7. RESULTS

The comparison for local vs Server Processing Time have shown that initially when the input puzzle is an easy or has medium complexity the local processing takes less time than Server. But as the puzzle becomes more complex especially when there are 17 or less than 17 clues then Server time becomes less than or equal to local processing time as shown in Figure 3. Hence this indicates that if such a process is offloaded to a nearby Server or Cloud then definitely, there is a gain in terms of energy savings and low battery consumption.



	1	2	3	4	5
■ Execution time for Local Processing	63	71	310	27134	31050
■ Execution time for server processing	239	420	1488	27206	27280

Figure 3: Sudoku solver execution time for device-only vs. offloaded execution in milliseconds

8. CONCLUSION AND FUTURE SCOPE

Mobile Cloud Computing is gaining its momentum and is encouraging the users to leverage the benefits that Cloud provides in order to ease the burden on resource constraint mobile devices. Offloading can become beneficial for computational intensive and mathematical applications, provided entities impacting the offloading are understood and analyzed thoroughly. Future research work is needed in developing efficient Offloading frameworks and optimal partitioning schemes which are portable across all the mobile execution platforms.

9. ACKNOWLEDGMENTS

Our thanks to the Professors, experts and other faculty Members who provided useful resources and background to complete this research paper

10. REFERENCES

[1] H. T. Dinh, C. Lee, D. Niyato and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches", Wireless Communications and Mobile Computing - Wiley, (2011) October.

[2] Kumar K, Lu Y. Cloud computing for mobile users: can offloading computation save energy. IEEE Computer Society 2010

[3] G. Karypis and V. Kumar, “Parallel Multilevel k-way Partitioning Scheme for Irregular Graphs,” Technical Report TR 96-036, Department of Computer Science, University of Minnesota, 1996

[4] Z. Li, C. Wang, and R. Xu, “Computation Offloading to Save Energy on Handheld Devices: A Partition Scheme, ”Proc. Int’l Conf. Compilers, Architectures and Synthesis for Embedded Systems, Nov. 2001.

[5] Ou S, Yang K, Liotta A (2006) “An adaptive multi-constraint partitioning algorithm for offloading in pervasive systems.” In:IEEE international conference on pervasive computing and communications, pp 116–125

[6] .Soleimanian Gharehchopogh Farhad, Raheleh Rezaei, Isa Maleki”Mobile Cloud Computing: Security Challenges forThreats Reduction”. International Journal of Scientific & Engineering Research, Volume 4, Issue 3, March-2013

[7] 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 Proc. 8th international conference on Mobile systems, applications, and services. ACM, 2010, pp. 49–62.

[8] Amazon simple storage service [Online]. Accessed May 5 2014 Available: http://aws.amazon.com/s3/

[9] B.-G. Chun, S. Ihm, P. Maniatis, and M. Naik, “Clone cloud: boosting mobile device applications through cloud clone execution,” arXiv preprint arXiv: 1009.3088, 2010.

[10] A Survey of Mobile Cloud Computing Application Models Atta ur Rehman Khan, Mazliza Othman, Sajjad Ahmad Madani, IEEE Member, and Samee Ullah Khan, IEEE Senior Member IEEE communications, Surveys and tutorials, vol.16, No.1 First quarter 2014

[11] J. Flinn, D. Narayanan, and M. Satyanarayanan, “Self-tuned remote execution for pervasive computing,” in Hot Topics in Operating Systems,2001. Proceedings of the Eighth Workshop on. IEEE, 2001, pp. 61–66.