ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Study on Computational Offloading Methods in Mobile Cloud Computing.

Meena V [1], Kalpana V[1], Anushia Devi R[2], and Senthil Kumar J[1].

[1]Computer Science and Engineering, School of Computing, SASTRA University, Tirumalaisamudram, Thanjavur - 613401. Tamilnadu, India.
[2]Information and communication, School of Computing, SASTRA University, Tirumalaisamudram, Thanjavur - 613401. Tamilnadu, India.

ABSTRACT

Mobile computing has been an emerging field for many years. A prototype for Mobile computing called Mobile Cloud Computing has evolved from the fusion of robust and potent, yet economically affordable mobile gadgets and cloud computing. The primary goal of this paper is to study prevailing mobile cloud computing applications. This paper also theorizes mobile cloud computing applications that may be emerging in the years to come. Further, perceptions for the equipping technologies and challenges that lie ahead have been furnished, to enable us to progress from mobile computing to mobile cloud computing for constructing the next generation mobile cloud applications. This paper supplies a survey of prevailing solutions, recognize the research gaps and suggest further research areas.

Keywords: Computational offloading, Profiling, Partition

Corresponding author

INTRODUCTION

Usage of mobile computing is very much increased. In end user and endeavour market for cloud based applications which are run on mobile are increased more than $10 billion. Mobile devices are played imperative to everyone. Nowadays developers develop their applications and have intention to deploy on mobile devices. So that, consumers can accesses anything, anywhere, anytime through mobile applications. Mobile devices are trying to perform a pervasive computing. But the grim snag is resource limitation. If high computation intensive jobs are executed in mobile devices there simultaneously battery consumption is also increased. Finally devices are getting conduit in very fast. Usually mobile devices are very being short of execution, battery detainment, CPU hustle when we compare wired devices.

Cloud computing is a very big doorway to all computing and smooth the progress of increasing the computing capabilities of resource constraint mobile devices by giving leased communications and services of cloud. Mobile computing and cloud computing are combined and gives new infrastructure called Mobile Cloud Computing. In that framework data storage and computation are doing in resourceful server (cloud) but not in mobile device.

In mobile applications, there may contain many sub applications of different computation time. When high intensive computations are executed in mobile devices, those devices will easily lose their battery power, automatically performance also getting degraded. In order to avoid that , part of the applications which has low computation are getting executed in mobile devices .And another part which has more computation are offloaded (i.e. migrated)to resourceful server which is called as cloud. This migration process is shown in fig 1. Now we have more questions to be answered. Whether offloading gives energy saving? How to decide which sub part will execute in cloud and which is in mobile device? How to deploy the applications in optimized utilization? This paper mainly focused on to confer in detail ongoing research in computational offloading and also issues which are addressed.

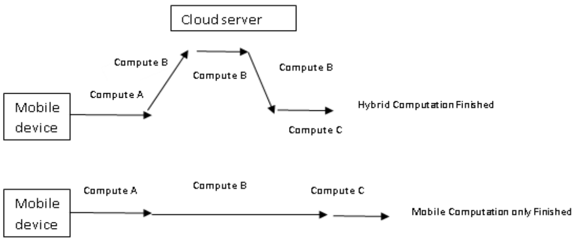

Figure 1: Computational offloading (Migration) process

Classification of Existing methods regarding to develop an Efficient Cost Model for energy saving in Mobile cloud computing

Methodologies applied for computation offloading is broadly classified based on profilers. These methodologies are trying to find the answer for the following questions.

- Whether to offload the computation or not.
- When to offload.
- Which to offload. They are giving answers by analyzing the collected data. Offloading decision is taken based on profiled data. Profiler is of two types static and dynamic profiling.

Computation Offloading to Coprocessors and Clouds

Ying- Dar Lin *et.al* [10] developed a framework called ternary decision maker to reduce response time and energy consumption .To provide comfort execution environment for applications which are executed in mobile, they have taken targets are cloud, on-board CPU, and an on-board GPU. They have taken matrix multiplication for evaluation. In that evaluations they observed that shorter response time, reduce energy consumption, high accuracy and low false decision rates. But they are not conduct experiment on different smart phones, different mobile applications and different network environment.

Clone Cloud

B.G.Chun *et al.*[5]developed a system intuit the mobile applications are transformed in automatic manner to get benefits of cloud. The system can partitioning the application in flexible manner and also enables frank mobile application which is running in application level VM to flawlessly migrate some part of load of their execution from resource constraint devices on to device clones operating in a computational cloud .For automatic partitioning they used both dynamic profiling and static analysis . Thread migration is also applicable at any point. Migrated thread is submitted to clone in cloud. Remainder part of the thread will be executed in clone. After execution reintegration will be performed to mobile device. For experimental purpose they have taken applications like searching process on image, virus scanner and a privacy-preserving targeted advertising. The results are found using history based profiling and implemented under Dalvik VM on Android. However, they are not given any generalized calculations for real networks and device conditions. And also pre-calculated partition cannot cover all the offloading scenarios.

MAUI

E. Cuervo *et al* [4] express the system by combining both full VM migration and code partitioning. In order to increase energy saving by adopting fine grained code offloading. They have an optimization engine to decide the methods whether to offload or not by considering the current connectivity status. From the experiment evaluation observed that they used history based profiling and also implemented in Microsoft .NET environment. However, their system is not supporting scalability. They are not experimented their results on various mobile computing OS platforms.

Cloud Computing for Mobile Users

Can Offloading Computation Save Energy?:K. Kumar *et al.* [1]They suggested cloud computing is pave the way for mobile user to save energy. They construct cost model based on that they decide whether they offload computation or not. Before offloading, they consider energy overhead related to security, privacy and data communication.

Adaptive Computation Offloading for Energy Conservation on Battery-Powered Systems [2]

They followed the different approach while giving solution for the problem of battery consumption in mobile devices. Initially they allow the program to execute on mobile devices with some timeout. There may be possible to have the computation which is not finished, even after threshold time that computation is recommended to migrate to cloud server. However, this method needs collecting online statistics for execution time for each computation. Based on this evaluation, they computed optimal timeout.

Comparison of existing cost model framework

S.No	Cost Model framework	Observations	Limitations
1	Computation Offloading to Coprocessors and Clouds[10]	Shorter response timeReduce energy consumptionHigh accuracy Low false decision rates	Not concentrate on different smart phones,mobile app,network environment
2	Clone Cloud[5]	Historybased profiling DalvikVM (Android)	real network &device conditions cannot be generalized.Pre-calculated partitions cannot cover all the offloading scenarios.
3	MAUI[4] [Mobile Assistance using Infrastructure]	Historybased profilingMicrosoft .NET	Non-scalable,Not adoptable onvarious mobile computing OS platforms.
4	Cloud Computing for Mobile Users: Can Offloading ComputationSave Energy?[1]	Parametric model Energy consumption low if B is high and D/C is low.NET (C #)	Not suitable for dynamic environment.
5	Adaptive Computation Offloading for Energy Conservation on Battery-Powered Systems.[[2]	Parametric model Time out is calculated Linux	Wastage of energy consumption for complex computation instances in mobile

Table 1: Comparison between existing cost model framework

From the above analysis we found that following are the drawbacks in existing methodologies that are listed here

- Started their work by having assumption for cost of link and the execution time can be obtained earlier.
- Does not persist profiling data across multiple runs of same program
- Does not consider all set of execution conditions
- The developed applications support platform dependent

Optimal cost model framework

The optimal cost model framework should not depend on any predefined platforms. It should work on all platforms. List of features are supported by optimal cost model framework is shown in fig 2

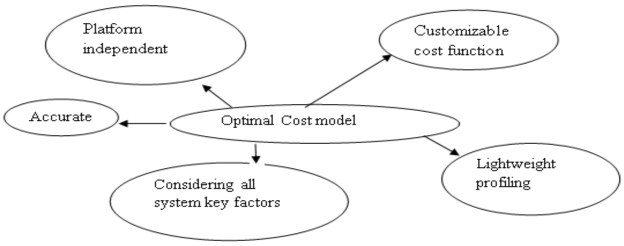

Figure 2: Optimal cost model framework

Existing methods to develop an Enhanced Low Cost , Accurate, Fine Grained and Dynamic Partitioning scheme for mobile cloud computing.

To offload or not to offload [6]: an efficient code partition algorithm for mobile cloud computing:They proposed partitioning algorithm in efficient manner . They assumed that when method is offloaded , the following calls will be offloaded with high probability. Using depth first search algorithm and searching algorithm which has linear complexity, their algorithm finds migrating and integrating points on the successive calls. Throgh their algorithm they 90% partition accuracy.

Comparison between existing application partition algorithm

S.No	Partition scheme used	Observations	Limitations
1	To offload or not to offload: anefficient code partition algorithm for mobilecloud computing[6]	Call graph & call link constructed DFS algorithm Partition accuracy moreFine grained task level ranularity	Not suitable for all type of programming structure
2	Clone cloud[5]	profile as many execution paths as possible by iterating the inputs Profile tree generated	pre-calculated partitions cannot cover all the offloading scenarios
3	MAUI[4] [Mobile Assistance using Infrastructure]	Cost graph constructed Adaptability more Formulate the partition problem into a 0-1 ILP Method level offloading	Programmer annotate methods as Remotable Platform dependent
4	The case for VM-based cloudlets in mobile computing[3]	No code partitioning logic Coarse granularity Virtualbox (linux)	Mobile device serves as a dumb client
5	Partition scheme[11]	Call graph is constructed Apply branch and bound Task level offloading	Adaptability low Accuracy low

Table 2: Comparison between Partition algorithm

The case for VM-based cloudlets in mobile computing [3] : Satyanaranan proposed cloudlet concept in mobile cloud computing. The computation in mobile applications is migrated to local computers named as cloudlet which consists of many multi-core computers. In case of remote cloud server, they suffer from bandwidth and latency issues. But here resource constraint devices can link and behave as dumb terminal to cloudlet which is far better than connecting to remote.

Partition scheme: Li et *al*.[11]proposed migrating methodology based on profiler. Using the profiled information , cost graph is constructed . Then, they are applied the partition scheme on the mobile application. So that, they divide the application into client tasks and server tasks.

From the above analysis we found that following are the drawbacks in existing methodologies that are listed here

- Requirement of skilled programmers
- Program partitioning based on static Inputs
- Lack of thread synchronization to support concurrency between mobile device and cloud
- Lack of data consistency

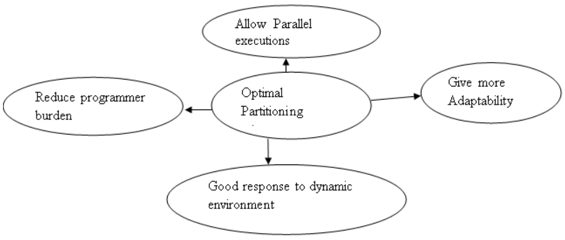

Figure 3: Optimal partitioning framework

Existing methodologies To develop Optimal Deployment model to offload computations in virtual machines in mobile cloud computing

Graph partitioning algorithm: Partitioned applications could be offloaded from resource constraint devices to resourceful server. Important point is how to deploy the applications in optimized manner. Optimized deployment reduces the network usage (bandwidth utilization).Traditional partition algorithm does not support the heterogeneity infrastructure. Verbelen T *et. Al* [7] proposed partitioning algorithm that supports heterogeneity infrastructure.

Different configurations for same application: Developers developed applications with different configuration. They provide the code with differed in quality and software requirements.Their framework[8] selectsthe appropriate configurations based on available resources.Observations and limitations of the above two paper is shown in table 3

S.NO	Deployment scheme used	Observations	Limitations
1	Graph partitioning algorithm[7]	Infrastructure heterogeneity taken into accou -nt Concentrate on minimizing bandwidth	Assume equally weight communicat -ion links between different servers
2	Dynamic deployment and quality adaptation for mobile augmented reality applications[8]	Multiple configurations of an application offering different qualities Target-cloud	During redeployment, performance getting dropped.

Table 3: Different Deployment approach

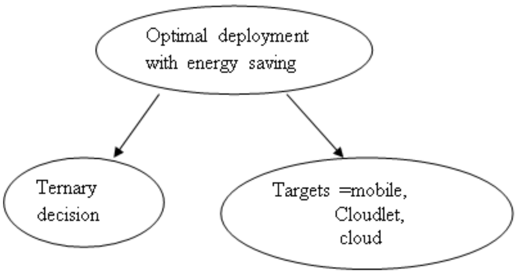
Optimal deployment with energy saving
Ternary decision
Targets =mobile, Cloudlet, cloud

Figure 4: optimal deployment

CONCLUSION

This paper study and classifies a various research associated with computation offloading for mobile systems. We survey various types of algorithm which are used for partition and offloading. We find various targets to deploy the application. Cost model should be optimized to have high energy saving and also to improve the performance of mobile device. Partition algorithm should be optimized to reduce the dependency factors while offloading the computation. Optimized deployment should be necessary to have good resource utilization in cloud.

REFERENCES

[1] K. Kumar, Y.-H. Lu, Cloud computing for mobile users: can offloading computation save energy? Computer 43 (2010) 51–56.

[2] C. Wang, Z. Li, Parametric analysis for adaptive computation offloading,SIGPLAN Notices 39 (2004) 119–130.

[3] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based cloudlets in mobile computing, IEEE Pervasive Computing 8 (2009) 14–23

[4] E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R.Chandra, P. Bahl, Maui: making smartphones last longer with code offload,in: Proceedings of the 8th International Conference on Mobile Systems,Applications, and Services, MobiSys'10, ACM, New York, NY, USA, 2010,pp. 49–62.

[5] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, A. Patti, Clonecloud: elastic executionbetween mobile device and cloud, in: Proceedings of the Sixth Conference onComputer Systems, EuroSys'11, ACM, New York, NY, USA, 2011, pp. 301–314.

[6] L. J. Yuan Zhang, Hao Liu and X. Fu, "To offload or not to offload:an efficient code partition algorithm for mobile cloud computing," inProceedings of the 1st International Conference on Cloud Networking,November 2012.

[7] Verbelen T,StevensT,DeTurckF,DhoedtB. Graph partitioning algorithmsfor optimizing software deployment in mobile cloud computing. Future GenerComput Syst2013;29(2):451–9.

[8] Verbelen T,StevensT, SimoensP, DeTurckF, DhoedtB. Dynamic deployment and quality adaptation for mobile augmented reality applications.JSystSoftw 2011;84(11):1871–82.

[9] Tim Verbelen a, PieterSimoens a,b, FilipDeTurck a, BartDhoedt a Q1. Adaptive deployment and configuration for mobile augmented reality in the cloudlet

[10] Ying-Dar Lin, Edward T.-H. Chu,Yuan-Cheng Lai, and Ting-Jun Huang Time-and-Energy-Aware Computation Offloading inHandheld Devices to Coprocessors and Clouds

[11] Zhiyuan Li,Cheng wang,Rong xu Computation offloading to save energy on Handheld Devices: A Partition scheme.