Adaptive Computation Offloading in Mobile Cloud Computing

Vibha Tripathi
MIO, NYC, U.S.A.

Keywords: Mobile Cloud Computing, Computation Offloading, Data as a Service (DaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Infrastructure as a Service (IaaS), Machine Learning, Artificial Intelligence, Augmented Reality, Internet of Things (IoT), Nash Equilibrium.

Abstract: Mobile Computing has been in use for a while now. A mobile device is a concise tool with limited computational resources like battery, CPU and memory. Although these resources suffice the immediate traditional needs of its user, as the mobile devices are fast turning into personal computing devices, with the rapid development in Cloud-Based technologies like Machine Learning in the Cloud, Data as a Service, Software as a Service, and so on there is an emergent need to implement iteratively more effective ways to offload mobile computation to the Cloud in an on-demand, adaptable and opportunistic way. The major issue in implementing this requirement lies in the very fact that mobile devices are location and context sensitive, limited in battery capacity and need to be constantly reconnecting with their provider's Base Transceivers while still providing efficient response time to its user. In this paper, we survey this issue and a few proposed solutions in this area and in the end; propose a model for adaptive computation offloading.

1 INTRODUCTION

Mobile devices have been around for some time now. Mobile Computing has come of age. Cloud Computing is rapidly growing in its own space. For Mobile devices to be able to leverage on Cloud Computing resources available today, there are some constraints to consider. We cannot simply merge the two worlds of Mobile and Cloud Computing due to the difference in nature of a handheld or wearable device versus a desktop machine virtually always connected on the same Local Area Network (LAN).

Mobile devices often need to connect and reconnect to their provider's transceivers owing to possible location changes.

The functional collaboration required between a Cloud and a Mobile Network, to make a mobile user's Quality of Experience (QoE) seamless is complex.

Computation offloading involves methods and means to decide the optimal nature and amount of computation to be delegated from a Mobile device to the Cloud on an on-demand basis and on-the-go.

In this paper, we study the nature of some of the complexities in Computation Offloading, proposed solutions and their limitations and finally propose a possible adaptive computation offloading model for Mobile Cloud Computing.

2 CLOUD ARCHITECTURE

Cloud Computing offers 'as a Service' frameworks that are used in collaborative computations. For Mobile Computing, the service components that are of more significance can be shown as in Fig. 1.

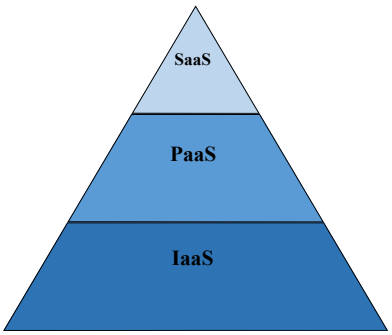

Figure 1: Cloud Service Layers.

The three layers are Software as a Service (SaaS) e.g. email service, data mining and machine learning applications, Platform as a Service (PaaS) e.g.

Tripathi, V.
Adaptive Computation Offloading in Mobile Cloud Computing.
DOI: 10.5220/0006348505520557
In *Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)*, pages 524-529
ISBN: 978-989-758-243-1

operating systems, and Infrastructure as a Service (IaaS) e.g. storage and virtual machines.

At the core of Cloud Computing is the concept of Hardware Virtualization – many virtual servers participate independently with their own operating environment but with the same physical layer.

Mobile Networks are laid out architecturally to provide an illusion of pervasive uninterrupted network connection to the end user while making location and reconnections appear seamless most of the times. The architecture is described in the Section 2.1.

2.1 Mobile Cloud Computing and Offloading Computations

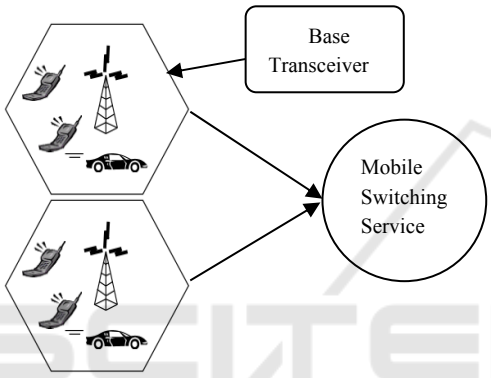

Figure 2: The figure shows how handheld devices switch connections to their base transceivers based on their location in a cell.

Computation and Data Intensive Apps: Some mobile applications may need to perform computation and data-intensive tasks like voice and speech analysis, facial expression analysis (Google Cloud Vision API), Augmented Reality etc.

Cloud based mobile apps can allow the end users to leverage the vast resources of the Cloud and run in a virtual environment, by offloading the data-intensive computation components to the Cloud and provide a better response time from the Cloud as opposed to local device-based computations when communication is light (Jesus Zambrano, 2015).

Mobile Battery Life: With the rapid increase in the multimedia content and computation intensive games being made available for mobile system, one of the primary constraints while considering computation offloading is the mobile battery life.

Battery technology has until now been unable to catch up with the fast growing battery consumption on mobile devices.

(Karthik Kumar et al, 2012) propose a formula to dynamically decide on computation offloading of certain energy-intensive computations with a view to benefit from offloading as opposed to local mobile based computation which can drain out the entire battery.

Along with the computational components, however, we also need to consider the location and context awareness of these mobile devices.

Location and Context Awareness of Mobile Devices: Mobile devices have sensors that can collect information about the location of the device, some context metrics about the users (e.g. *Fitbit*) for example to recommend services, customize answers to user queries, advertise local businesses, provide individualized search results etc. (Gabriel Orsini et al, 2015) discuss the requirements, design and a pre-phasing or partitioning for computational offloading to arrive at an optimal context aware solution.

A typical Cloud-enabled Mobile device would have the below minimum requirements:

- An uninterrupted connection to a Cloud service provider
- An efficient computation offloading model that takes context awareness as well as battery usage efficiency into account
- An effective computation component distribution model

2.2 Offloading Model

Fig. 3 shows how the current offloading process appears considering the major participants. However the components involved in a typical mobile application computation offloading to Cloud can be categorized as below:

- *Surrogate* – a computing node or a virtual environment made available by the Cloud service provider where the offloaded code can run
- *Partitioner* – divides and sorts out the application components to be offloaded for Cloud-based computing, this could be static or dynamic (Porras, J. Riva, 2009)
- *Context Monitor* – Monitors and provides context related information about the device, the available surrogates in the local area, battery status, connectivity etc.
- *Solver* – uses the information from Context Monitor to decide which surrogate to offload the computation the Partitioner decides to offload

- *Coordinator* – authenticates and securely sends computation over to surrogate, receives and presents the results from the surrogate

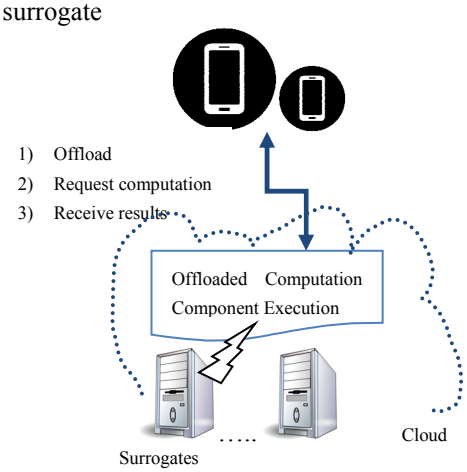

Figure 3: Offloading Mobile Computation.

2.2.1 Offloading Granularity and Surrogate Selection

Several approaches are in use at the moment, with each having its own pros and cons. Traditional remote method invocation (RMI) and client-server model for offloaded computation keep the object-oriented-ness for mobile app designers but require synchronization of variables which may mean error-prone context-insensitive computation results.

Virtual Machine based offloading can be a language based VM for the mobile app or a mirrored mobile computing environment with the ability to run any app in a given mobile environment.

(Xu, Chen et al, 2015) propose a game theoretic approach to decision making problem in computation offloading amongst multiple mobile device users. They formulate the problem as a multi-user computation offloading game and prove that it always admits Nash equilibrium. They also propose a distributed offloading algorithm that can achieve this equilibrium. However, the dynamicity of mobile users frequently joining and leaving the game on-the-fly has yet to be considered in this approach.

None of the above solutions however, make it easy for the Partitioner to decide the granularity of offloading computation components on-the-fly or for the Coordinator to decide which Surrogate to offload to.

2.2.2 Opportunistic Partitioning

For an opportunistic Partitioner the application should be able to compute the cost of local versus cloud-based computation based on the context information e.g. battery usage, available bandwidth and quality of network connection. Derived from the report by (S. Pachamuthu & Kumar, 2016) a simple formula to compute the cost can be as below:

$$\alpha\,(Tcloud - Tlocal)/Tlocal + (1-\alpha)(Ecloud - Elocal)/Elocal$$

Where α denotes a weight between 0 and 1 towards computing time T or energy consumed E being assigned that weight.

Mobile Apps that can run optimizations to decide what parts of their computation should be offloaded and what should be run locally can either come with pre-defined static partitioning of the program into components while some others can run this optimization itself in the cloud.

2.2.3 Dynamic Surrogate Selection

In Section 2.2.1, we mentioned the complexity with Coordinator to decide which Surrogate to connect to for offloading. Considering the mobility of a handheld, we acknowledge another level of complexity to the decision making – the mobile device should potentially be able to connect and offload to multiple Surrogates and able to receive and reconstruct results from them dynamically.

2.3 Current Solutions

2.3.1 Native Mobile Cloud Computing

Some solutions in use today have been designed specifically to solve the offloading from mobile devices. These solutions fall in two broad categories. The first category is, where the entire mobile application or parts of it are run in Virtual Machines mimicking the mobile device on servers to decide which threads of methods should be offloaded when running from a mobile device. *CloneCloud* (Chun, B.G, 2011) and *MAUI* (Cuervo, E., 2010) are such examples. The application running on the server's VM is profiled for its cost of computation.

In the second category, the computation components are analysed a-priori and decisions are made to offload more computation intensive tasks to Surrogates. This solution doesn't take into account the context sensitivity of the mobile device and hence results can be unsatisfactory.

2.3.2 Frameworks and Domain Specific Languages

There are many programming languages in market today, specifically designed to solve problems in domains like mobile commerce etc. These languages are often required to run in their specialized run time environment. While this solution follows an agile approach as the developers are focussed in the domain, it does require developers to niche their expertise for a specific language and run time environment.

Various frameworks and platforms like Java Remote Method Invocation provide proven methodologies for distributed execution of applications, thereby letting the developer focus on the application functionalities. However, these conventional standard frameworks are not flexible enough to support mobile computation offloading in rapidly varying contexts.

2.3.3 Distributed Virtual Machines

In this approach multiple VMs run on multiple nodes in a network and run computation components with a common global state of the application (Coulouris, G., Dollimore et al, 2012). However this solution requires the mobile device to support the distribution of the tasks. There are proposals for using embedded devices in routinely used objects like shoes, watches and so on for computation offloading on personal distributed systems (Niroshinie Fernando. et al, 2013)

(Verbelen, T., 2012) propose a new architectural paradigm by converging mobile computing, Internet of Things (IoT) and Cloud Computing. A *Cloudlet* sits between the Cloud and the Mobile or IoT device and can serve as a small data centre bringing the cloud closer to the device by providing a PaaS layer which can support wearable cognitive assistance, making possible new flurry of applications based on Artificial Intelligence and Augmented Reality for the Mobile or IoT user.

2.3.4 Other Solutions

Some solutions take the application partitioning and component execution to the system level. Mobile applications in this approach are *designed* to assure that computationally intensive parts of the app can be run in distributed environment. These solutions do not however do part by part offloading from the mobile device and instead consider the application in entirety as per the design.

3 TOWARD AN ADAPTIVE OFFLOADING MODEL

We propose that an effective cloud-enabled Mobile App should be elastic in its decision-making of what and where to offload its computation.

Additionally, the Mobile Cloud Computing infrastructure should be able to provide a dynamic selection of Surrogates that can pass on partial or completed results from offloaded computation based on the changing context and location of the device.

The benefit in our model arises from the Surrogates providing a virtual PaaS by partially or completely participating in the computation of offloaded requests. The Mobile device may not be aware of this virtual network of Surrogates and may receive the results of its offloaded computation from any one of the participating Surrogates.

3.1 Virtual Network of Surrogates

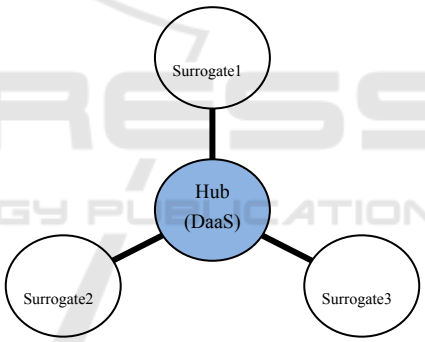

Figure 4: Virtual Networks of Surrogates.

We propose a model for surrogates that connect amongst themselves via a cloud-based hub that provides Data as a Service. When a mobile device selects for example Surrogate1 in the Fig.4, based on the application's requirements, it either precomputes partitioning or offloads the execution components entirely to that Surrogate, any computation results and context information is sent to the *DaaS* Hub, which upon request provides either partially computed or completed results to another Surrogate which the mobile device may have selected upon location change.

Context information and the cost of offloading may also have changed with the location, and hence that should be taken into consideration when selecting the next Surrogate.

Having a Hub providing DaaS, resolves several issues like *availability*, since some computation results may always be available for the mobile device via the Hub once at least the first Surrogate was selected for offloading. It also provides data *security* as most of the data intensive transactions would be limited within the Surrogate network.

The above proposed model also eases *maintainability* of mobile applications as developers can focus on the application itself while most of the intensive computation, including the optimization for Partitioner to decide what components to offload, can be run in selected Surrogate once and all that is exchanged with subsequently selected Surrogates could be Context Sensitive information.

3.2 Multi-Surrogate Distributed Offload

The model proposed in Section 3.1 would also allow for more *scalability* as multiple Surrogates can be employed in an elastic manner to compute a distributed task intensive application. The selected Surrogates can, not only compete for their candidacies based on the Virtual Environment they can provide to run parts of computation from a mobile app, they can also utilize results from participating Surrogates.

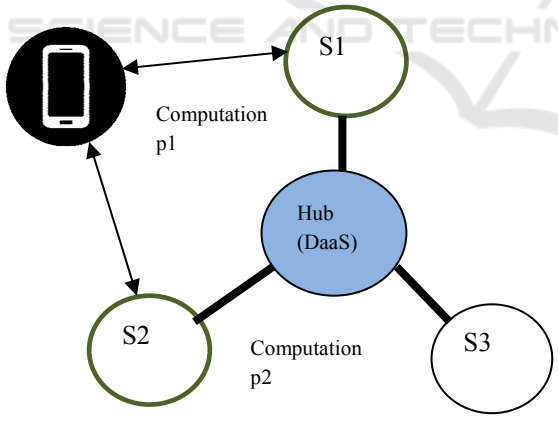

Figure 5: Distributed Computation Offloading.

This way we can apply heterogeneously specialized VMs, for example to provide Cloud Based Machine Learning routines, Video Games and so on.

3.3 Centralized Offload and Distribution

Another approach with the abovementioned virtual network of surrogates can be as shown in Fig. 6 for the mobile application to offload centrally to the first selected Surrogate which acts as a master and decides to distribute tasks to other Surrogates based on the resource availability. Big Data solutions like Hadoop internally employ such distribution of tasks.

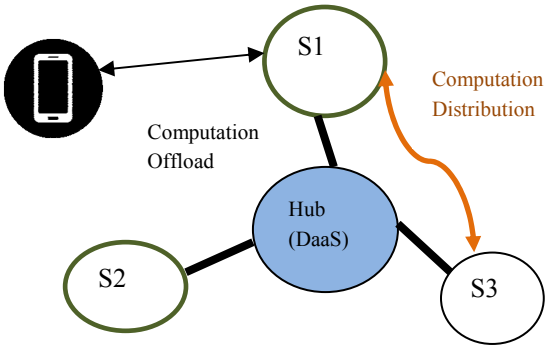

Figure 6: Centralized Computation Offloading.

4 CONCLUSION

The main challenges in the field of Mobile Cloud Computing include Context Sensitive Computation Offloading. In order for the Mobile users to leverage from the vast resources and functionalities offered by rapidly advancing Cloud Computing technologies, much further research is required in the area of computation offloading.

We studied state of the art Mobile Cloud Computing, current challenges and a few proposed solutions. Based on the study we proposed a model for opportunistic and adaptive offloading with a focus on the context awareness of computation intensive mobile applications. Further work is required to run evaluations with computation intensive mobile applications using the model and fine tune the model.

In conclusion, Mobile Cloud Computing is still in its nascent state and while much of the traditional distributed application approaches can be utilized to some extent in the context of MCC, certain areas like adaptive context sensitive computation offloading require further research.

REFERENCES

Porras, J. Riva, O., Kristensen, M.D. Dynamic resource management and cyber foraging. In: Middleware for Network Eccentric and Mobile Applications. Springer; 2009, p. 349–368.

Coulouris, G., Dollimore, J., Kindberg, T.. Distributed Systems: Concepts and Design. Boston, USA: Addison-Wesley; 5 ed.; 2012.

Verbelen, T., Simoens, P., De Turck, F., Dhoedt, B.. Cloudlets: Bringing the cloud to the mobile user. In: Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services; MCS '12. New York, NY, USA: ACM. ISBN 978-1-4503-1319-3; 2012, p. 29–36.

Gabriel Orsini, Dirk Bade, Winfried Lamersdorf Context-Aware Computation Offloading for Mobile Cloud Computing: Requirements Analysis, Survey and Design Guideline, MobiSPC 2015

Karthik Kumar, Jibang Liu, Yung-Hsiang Lu ,

Bharat Bhargava: A Survey of Computation Offloading for Mobile Systems, Mobile Netw Appl DOI 10.1007/s11036-012-0368-0, 2012

Jesus Zambrano, Mobile Cloud Computing: Offloading Mobile Processing to the Cloud, University of North Florida 2015

Niroshinie Fernando, Seng W. Loke, Wenny Rahayu; Mobile cloud computing: A survey, Elsevier: Future Generation Computer Systems Volume 29, Issue 1, January 2013, Pages 84–106

Chun, B.G., Ihm, S., Maniatis, P., Naik, M., Patti, A.. CloneCloud: elastic execution between mobile device and cloud. In: Proceedings of the 6. European Conference on Computer Systems. 2011, p. 301–314.

Cuervo, E., Balasubramanian, A., Cho, D., Wolman, A., Saroiu, S., Chandra, R., et al. MAUI: Making smartphones last longer with code offload. In: ACM MobiSys 2010

Xu Chen, Lei Jiao, Wenzhong Li, and Xiaoming Fu Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing, IEEE/ACM Transactions on Networking 2015

S. Pachamuthu & Kumar Rochester Institute of Technology, Rochester, New York Cost Evaluation of Computation Offloading on Mobile Devices, 2016