

A Context Aware Mechanism for Mobile Devices to Efficiently Acquire Resources Remotely

Kashif Tasneem

Department of Computer Science
and Engineering
University of Engineering and
Technology Lahore

Ayesha Siddiqui

Department of Computer Science
and Engineering
University of Engineering and
Technology Lahore

Anum Liaquat

Department of Computer Science
Virtual University of Pakistan

ABSTRACT

Mobile devices are becoming very popular due to their portability and small size, but they are still not comparable to that of PC. Computational and network intensive applications cannot run efficiently on mobile devices due to their limited batteries and energy deficient systems. Recent works to improve mobile device efficiency include mobile cloud computing, mobile edge computing, and device to device communication. Shifting the computations on the cloud will reduce the burden of mobile device but at the same time, this implementation presents many challenges. In this paper, an algorithm is proposed which can deal with the resource acquisition from different sources by using a context aware system. Implementation of that algorithm is left for future work.

General Terms

Cloud Computing, Mobile Edge Computing, Resource Management.

Keywords

Cloud Computing, Cache Computing, Fog Computing, Mobile Edge, Offloading

1. INTRODUCTION

With the increase in mobile devices especially smart phones, different types of software applications are gaining popularity which has let us to the very advanced technologies in term of mobile software. Applications like games, streaming and social media are very commonly and frequently used to remain updated but unfortunately hardware technology for a mobile device is still lacking behind. Even though, latest smartphones use multi core processors, high end graphics cards, and amounts of RAM which were unthinkable in the past but there is still a big gap in mobile software and hardware technologies which in turn present many challenges.

Mobile users want a more thorough and advanced experience just like a PC in their hands which currently can't be implemented because of limitations such as limited battery, storage, computations capabilities and weak backhaul network resource. Demands of users seem to be increased in upcoming years because of the introduction to the augmented and virtual reality applications which in turn will burden our networks with bulk of requests. Our network technologies are also changing fast, but they still are unable to meet mobile user requirements when multiple users are requesting similar type of resource.

There are different augmentation techniques which put storage and computing capabilities on cloud to enhance user experience on smartphones. Mobile Cloud Computing (MCC) is very useful to avail services of cloud instead of using local mobile resources but at the same time it presents some issues

like longer round-trip times and congestion of wireless network due to increased requests [30]. Researchers have proposed many solutions for MCC from which light is being shed on some popular techniques.

1.1 Offloading

Computational tasks are offloaded to the host through network [7]. It is only viable if its efficient in every aspect like power consumption, network speed and mobile batteries. Before offloading any task to the cloud, one must properly compute the power efficiency for different components otherwise it may result in performance degradation. Internet connection must be reliable for task offloading which might not be the case with current network technologies as internet might not be available in all situations or is of very limited connectivity. The increase in number of mobile devices is giving rise to the revolution of mobile applications from which most are poor in resource handling. Shifting computation to the cloud consumes mobile batteries which also makes offloading less efficient.

One other aspect of offloading is concerned with multi-users. Single user offloading might pose less challenges but what happens when multiple users want to utilize cloud computation on mobile device? [6] A solution is also needed to the problem of when and how a user should select a cloud computational resource instead of using his local mobile based services and through which network channel a user can get better efficiency.

1.2 Mobile Edge Computing (MEC)

It provides computation resources for mobile users on the network edge. As discussed in [9] many techniques of caching the content at the edge are proposed which reduces network traffic by caching the most requested content either at the radio access network or core network. This technique is useful to reduce load from mobile device and to shift it to the cloud by using nearest radio access network [11]. Some techniques like cloudlets [30] and foglets [23] provides services like storage, computation etc. at the intermediate levels of cloud and mobile but they are not practically implemented yet. Main purpose of edge or fog computing is to provide the needed resources nearest to the users to enhance performance. In case, network edge is not capable of entertaining the request, only then the requests should be forwarded to the higher level of cloud servers. It is also a question of interest that how to determine that which requests should be forwarded to the cloud.

1.3 Device-to-Device Communication (D2DC)

One other method to reduce load from the mobile device is to create a cloud of mobile devices in which a device to device (D2D) communication is performed. In this method, offloading benefit is achieved by sharing one device resources with other available devices. This sharing requires their users to agree upon a mutual decision of how and which resources they would like to share on the cloud. This technique is very efficient if there is no internet connectivity and users are willing to share their resources with each other. Many design frameworks have been proposed [22] [12] on mutual resource sharing of mobile devices on cloud, however, there is a point of concern that why would a user like to share its resources with anyone?

Some of the major key issues faced in using heterogeneous computing resources and wireless networks to implement mobile cloud augmentation are given below:

1. **Outsourcing Computation:** How to outsource computation is one of the main challenges. It means how to select the best augmentation approach.
2. **Unstable Wireless Communication:** Wireless communications are unreliable, they are dependent on too many variables like channel interference, channel capacity and limitations of carrier packages.
3. **Mobile Device Mobility:** Smartphones are portable which is a big plus, but it is also a big negative regarding network connectivity. Network signals are strong in a certain area and are weak in others, a system is needed which can provide consistent computing capabilities across all areas.
4. **Context Aware Augmentation Decision Making Strategies:** Augmentation is affected by heterogeneity of cloud resources and applications. For example, code offloading may consume a high amount of battery as compared to if running locally on the device. Hence, intelligent augmented strategies need to be designed to cater such problems.
5. **Data Security:** Outsourcing computation and data increases security risks. The cloud on which these items are hosted must be secure, reliable and should handle privacy very efficiently.

All the techniques discussed so far focus on one aspect separately, but they don't provide a comprehensive solution to the problem of mobile cloud computation. In this research work, an algorithm is provided which deals with the computational problems of cloud, edge and D2D communication as well. Rest of the paper is organized that section II is about the related work done, section III is about all the augmentation techniques related to the MCC and section IV provides an overview of the design, section V provides detail of our proposed algorithm, section VI concludes our paper with some future directions. At the end acknowledgment and references are presented.

2. RELATED WORK

Mobile cloud computing greatly depends on the network topologies and the availability of internet to use the cloud resources which makes it unreliable for emergency situations. MCC can use three schemes to operate [14]. One traditional approach is to request the cloud directly for any resource

while another is to make a distributed cloud of mobile devices connected through a stable internet connection sharing resources through a pool [15]. There is a third hybrid approach in which a mobile device is connected to the cloud as well as to the pool of local mobile devices [3].

2.1 Power Consumption

A model for different components like CPU, Wi-Fi and RAM is given in [7] to estimate the power consumption and what will be the effect of dynamic offloading by using that model. First, different energy models of various components like CPU, Display Unit, Wi-Fi, and RAM are proposed and then they are tested on two android mobile phones to determine the power consumption of different components. To validate the proposed models, experiment was performed by using two applications i.e. a web browser and an application like web browser but without display and results showed an average of less than 8% of error in the model.

All the proposed power models are used for the estimation of energy consumption which in turn helps in making decision of dynamic offloading for different components. So, application developers can utilize this information to develop energy efficient applications in such a way that energy intensive parts can be offloaded to the cloud to save mobile battery. To calculate power consumption of different components of a mobile device, there are different techniques like PowerTutor [26], background logger [29], PowerProf [19], AppScope[17] and by using an external power meter [18]. All mentioned models are only for single core processor, but models proposed in this paper are supported for multicore mobile devices without using any extra equipment. After estimating power consumption, next task is to offload the computation on the server. Different architectures have been proposed to offload computations like MAUI [27], CloneCloud [24], Cukoo [29], ThinkAir [16], MISCO [28], COSMOS [10], Honeybee [13], Serendipity [20] and AIOLOS [21]. AIOLOS framework is extended to optimize energy consumption both locally and remotely.

2.2 Battery Improving Techniques

Energy Harvesting technology uses different types of renewable energy sources to resolve battery related issues of mobile devices [25]. The design strategies on how to use energy harvesting techniques in mobile edge computing for mobile devices are discussed in [8] where the computation tasks can be executed locally at the mobile device or be offloaded to the server for cloud execution. Failure and delay are the core problems encountered during the execution which are minimized in the proposed algorithm. It is found that greater percentage of battery results in shorter execution time. The proposed algorithm is also suitable for environments which are unpredictable because it does not require any additional information related to computation request, wireless channel and energy harvesting etc. Overall, Lyapunov optimization-based dynamic computation offloading (LODCO) algorithm reduce execution cost and task failure.

2.3 Caching & Edge Computing

In [9] techniques are provided to improve the cloudlet services by minimizing the user perceived latency, service placement transition and resources usage from provider perspective on cloudlets. Different types of caching and edge computing techniques are discussed in [2]. Edge caching can be performed at the evolved packet core (EPC) which might cause a bang in index size of network because packet size is very small. Another way is to cache data at the RAN which is

more complex due to the tunneling and packaging of data. D2D caching is also a viable solution by assuming that user requirements are known in advance. The key concept of mobile edge computing is to provide fast computations on cloud with location awareness. There are two types of cloud fog architectures i.e. fog-fog and fog cloud computing. As battery is the main bottleneck of mobile devices, therefore, edge caching techniques should be energy efficient. At the same time, by using edge caching, load on backend servers can be reduced because number of requests directly received to the server will decrease [4].

2.4 Service Availability Through Replication

If a cloud has multiple replicas, then its service availability will be much higher than a single node cloud, but replication will also cost in terms of energy consumption which is not desired. So, how one can provide reliable service while keeping the energy consumption at its lowest level? Software design can play an important role through which an appropriate decision can be made that when to replicate some service. A context aware software architecture is proposed which configure itself for activation of replicas depending on the context information. The proposed solution is only suitable for small and medium size network because as the network size increases, the factor of efficiency decreases. In the proposed architecture [3], there are four basic components i.e. communication middleware, monitoring subsystem, context manager service, replica manager service. An election algorithm is used to decide which replica to activate/hibernate for service availability.

2.5 Incentive Based D2DC

Tang et al [5] have discussed the development of a mechanism to attract more users to participate in resource sharing architecture. The proposed method is based on a broker which collects information of users who are willing to give their resources on some price and give that information to the needed users if they are willing to accept the bided price. In this way, more users will participate in the resource sharing due to the incentives they get. In case where all the users are price-taking and there is not any supplier than what will happen? An equilibrium must be maintained to keep the mobile cloud sharing possible in a vicinity. A broker is an entity which is responsible for maintaining equilibrium. Although communication between giving users and taking users through broker might pose some extra burden but it is necessary to maintain a market flow.

3. AUGMENTATION TECHNIQUES

Augmentation techniques are mainly divided into two categories i.e. computation augmentation and storage augmentation [1].

3.1 Computational Augmentation

Computation augmentation is further divided into the computational, decision and supporting techniques.

3.1.1 Computational Techniques:

They are further categorized as the models and architectures.

3.1.1.1 Augmentation Models

Popular augmentation models include code offloading and service-oriented task delegation.

Code offloading aims at migrating computationally intensive code from resource limited machines to remote computing resources to accelerate the running process of the computation

and reduce energy consumption on limited battery devices. There are multiple ways of code offloading such as *Partitioned Offloading* in which only sends computation intensive part to the remote servers either statically (hard code partitions) or dynamically (using flow-based programming, .NET common language runtime programming, Java Reflection and distributed shared memory), *VM Migration* which moves computations dynamically among machines in the distributed system without interrupting the ongoing execution mainly focusing to improve energy efficiency while considering network conditions, *Mobile Agent-Based Offloading* in which whenever an application using mobile agents needs to request a service from a remote server, like the cloud, it collects the application execution information and passes it to the execution environment of the agent for offloading and remote execution and *Replication Based Offloading* which benefit the network intensive applications by creating their replicas on remote servers.

Service oriented task delegation augments resource-limited mobile devices by utilizing existing services on remote servers with remote process invocation rather than migrating part of the application to the server for execution. It can be implemented using *Web Service Based Mobile Cloud* in which there is no need to develop native mobile application but only a thin client to relay the service requests to the cloud servers or *Open Service Gateway Initiative* which is a Java module management system enabling applications to dynamically load and unload service bundles at runtime.

3.1.1.2 Augmentation Architectures

Utilizing *parallel execution* to distribute computation can significantly reduce the time overhead of handling big data applications by dividing the computation and related data into sub problems with a chunk of data, which are then processed simultaneously on multiple resources. Different frameworks such as GPU computing, OpenCL and MapReduce have been adopted to cope with the big data application.

The *opportunistic mobile collaboration network* provides a solution for mobile devices to dynamically utilize other peer devices in the vicinity to overcome the issue of unstable wireless network connections. Many consider MANET as a mobile device cloud, which is a network of mobile devices connected wirelessly in either a centralized or distributed manner, but its performance can be affected by network congestion. Proactive and reactive fault tolerance techniques can be used to improve network stability.

3.1.2 Decision Making Techniques:

They help in making decision of when and where to offload the code. For making offloading decisions, three main categories are discussed: stochastic process, analytic model, and resource monitoring and profiling. For making task scheduling decisions, the techniques are classified into four categories: heuristic, combinatorial optimization, metaheuristic, and game theory.

3.1.3 Supporting Techniques:

These include the mobility impact and fault tolerance techniques caused by device's mobility.

3.2 Storage Augmentation

3.2.1 Mobile Storage Augmentation Techniques

Most commonly used mobile storage augmentation solutions involve public cloud storage services. To extend mobile storage, the solutions propose middleware's between cloud

services and mobile devices to provide data offloading and data management functions.

3.2.2 Data Protection

Three major issues related to data protection are data security, accessibility and authentication. Data security can be ensured by using HTTPS, SSL/TLS, MD% and SHA to capture the original data signature and use of encryption techniques. Attribute Based Data Storage (ABDS) can be used for access control schemes and ALILI framework was proposed to solve the group file sharing problems in mobile cloud systems.

3.2.3 Data Interoperability

One solution is to apply standardized service frameworks and message exchanging techniques such as SOA, REST, XML, and JSON to mobile augmentation systems. Some have used a standard service API's for mobile and cloud device while others have developed some uniform mobile application platform to communicate between different devices.

4. PROPOSED DESIGN

A new design paradigm for mobile cloud computing to increase computational capabilities with energy efficient models for mobile batteries is presented here. When this system will be implemented, it will provide variety of options to both smartphone manufacturers and cloud providers. Regarding user's point of view, users will be able to experience any type of service on their own unlike now-a-days, where you need to purchase a new device, when you need to upgrade it specs. If cloud services are used, users can pay via subscription and experience the smartphone environment according to their requirement. Our design caters with all the possible conditions which can occur during a request handling from mobile to cloud. Our model consists four parts mainly as shown in fig 1. First component is the mobile device which is our main concern. Second level components include broker and network edge. At third level there is cloud server which is a main hub for the service provision. In case of broker communication, last level of contact will be other mobile devices available in the vicinity. Details of each level components is given below.

4.1 Users

Mobile device users are the main entity of our system which require resources from remote stores for a better performance. Mobile user can be using any mobile device like phone, PDA's, iPad's etc.

4.2 Resource Manager

Resource manager is the second most important component of our model. There are two types of resource manager in our case. One, which is a broker responsible for the communication and managing of D2D resource sharing and second is the network edge. In case broker, local mobile connections will be used to avoid any delays and network connectivity issues. Network edge is anything between cloud and the mobile users which is managing the resource for fast entertaining of the requests. Network edge basically cache the most received requested data at its end to avoid direct connection to the cloud server. By using these resource managers, computation capabilities are enhanced.

4.3 End Devices

In our case, end devices are the other mobile users who share their resource with the broker in return for some incentive.

4.4 Cloud Server

Cloud server requires connection to the internet to provide any resources to the mobile users according to their demands.

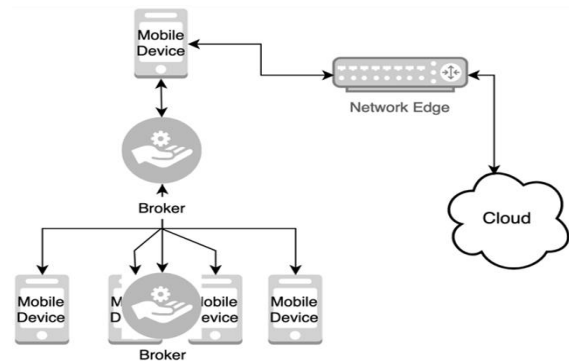


Fig 1: Model for Context Aware Resource Management in Cloud

5. ALGORITHM

An algorithm is provided in fig. 2 through which a mobile device which is running out of resources needs a resource from remote stores. According to our algorithm, when a mobile user requires any resource then first of all it will contact the broker to get some of the available resources. Broker is the manager of all the requests from suppliers as well from providers. Incentive mechanism is managed in such a way that if a user requests for a resource then he must be willing to share a resource whenever the broker will need. For example, a user with high storage capacity is running out of battery and he requests the broker to give him some energy harvesting resource, so that he can use his device without any intervention. In that case, broker will make a deal to take his storage to keep a stock of memory.

Now, comes the second case, when broker is not able to entertain the user request. In that situation, request will be forwarded to the network edge. As network edge is only able to entertain a specific type of requests, therefore, it will see if the request can be entertained and it has available resources in its cache.

Due to limited capacity of network edge devices, if a user requests some different type of request other than usual then that request will be forwarded to the cloud server. Cloud server request might take some time because there can be network connectivity issues, therefore, it is best to keep the cloud a last choice.

```

if (device is running out of resources) {
    contact broker;
    broker will request other devices for required resources
    (mutual sharing);

    if (required resource found) {
        allocate resource;
    }
    else {
        contact network edge;
        if (required resource found) {
            allocate resource;
        }
        else {
            request cloud for resource;
        }
    }
}
}

```

Fig 2: Algorithm for a Mobile Device to Acquire Resource Management in Cloud

Our algorithm is efficient from the previous research that it is taking into account all the possibilities of connections instead of relying on a single source. This may take longer implementation time but once implemented will definitely change the mobile usage future.

6. CONCLUSION

Mobile cloud computing is gaining popularity due to increase in mobile applications. This increase has led to the concept of mobile cloud computing. In this work, different techniques and architectures proposed in the literature for MCC are studied. All the techniques presented are until now being used for one type of resource sharing. By keeping the previous research works in consideration, an algorithm is provided which should be able to deal the requests of different users in different situation. A context awareness is introduced to reduce the delay time and network burden. Context is very important as it is the only thing that determines the point of contact for a mobile to use available remote resource. Our proposed algorithm is efficient in the sense that it tries to fulfil some user request at the lowest possible level. If it does not find the required resource at a certain level, only then the requested should be forwarded which in turn reduce the overall network traffic and increase response time.

7. FUTURE DIRECTIONS

Implementation of this algorithm might present many challenges as it will be complex. It should be further investigated that how to this algorithm will be able to accommodate all the connections. In future, practical implementation of this algorithm on small scale will be presented. Our proposed algorithm is using a broker middleware for the device to device communication but in future, a mechanism can be devised that will enable the devices to contact with each other without any third-party intervention. Some of the points which should be further explored are:

1. **Heterogeneous Mobile Cloud Service:** A variety of smartphones are available in the market having different Operating systems and different types of sensors. The proposed cloud system should be able to handle requests and data from all these types.
2. **Context Awareness:** Additional information such as device mobility, network info, sensors being used can provide additional info on how to handle the cloud system across a variety of different types of devices.
3. **Quality of Service Management:** An efficient system should be designed which should have a small server response time, can scale resource according to current app usage and provide constant wireless communication.
4. **Reliability of Mobile Cloud Augmentation Systems:** As smartphones are mobile devices, the environment always keeps changing. This can affect the availability of computing resources which are dependent on network. So, a reliable system needs to be designed which provide proactive fault tolerant systems.
5. **Security and Privacy:** Due to data transmission between smartphones and servers through wireless network, security and privacy are major concerns.

Security system should be lightweight and should be handle to large amounts of data.

6. **Incentives for Service Users:** It's hard to convince users to adopt a new system which is introduced. So, an incentive system should be introduced so that more users can opt-in and this system be improved.

8. ACKNOWLEDGMENTS

We are greatly thankful to the department of computer science and engineering, UET Lahore for giving us this opportunity to do research in different areas.

9. REFERENCES

- [1] Zhou, Bowen, and Rajkumar Buyya. "Augmentation Techniques for Mobile Cloud Computing: A Taxonomy, Survey, and Future Directions." *ACM Computing Surveys (CSUR)* 51.1 (2018): 13.
- [2] Erol-Kantarci, M., & Sukhmani, S. (2018). Caching and Computing at the Edge for Mobile Augmented Reality and Virtual Reality (AR/VR) in 5G. In *Ad Hoc Networks* (pp. 169-177). Springer, Cham.
- [3] Guerrero-Contreras, G., Garrido, J. L., Balderas-Diaz, S., & Rodríguez-Domínguez, C. (2017). A context-aware architecture supporting service availability in mobile cloud computing. *IEEE Transactions on Services Computing*, 10(6), 956-968.
- [4] Bahmani, K., Argyriou, A., Erol-Kantarci, M.: Backhaul relaxation through caching. In: Imran, M., Raza, S.A., Shakir, M.Z. (eds.) *Access, Fronthaul and Backhaul for 5G Wireless Networks*. IET (2017)
- [5] Tang, L., He, S., & Li, Q. (2017). Double-sided bidding mechanism for resource sharing in mobile cloud. *IEEE Transactions on Vehicular Technology*, 66(2), 1798-1809.
- [6] Chen, X., Jiao, L., Li, W., & Fu, X. (2016). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 24(5), 2795-2808.
- [7] Ali, F. A., Simoens, P., Verbelen, T., Demeester, P., & Dhoedt, B. (2016). Mobile device power models for energy efficient dynamic offloading at runtime. *Journal of Systems and Software*, 113, 173-187.
- [8] Mao, Y., Zhang, J., & Letaief, K. B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*, 34(12), 3590-3605.
- [9] Yang, L., Cao, J., Liang, G., & Han, X. (2016). Cost aware service placement and load dispatching in mobile cloud systems. *IEEE Transactions on Computers*, 65(5), 1440-1452.
- [10] C. Shi, K. Habak, P. Pandurangan, M. Ammar, M. Naik, E. Zegura, Cosmos: Computation offloading as a service for mobile devices, in: *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '14*, ACM, New York, NY, USA, 2014, pp. 287-296.
- [11] European Telecommunications Standards Institute (ETSI), "Mobile-edge computing-Introductory technical white paper," Sep. 2014.

- [12] S. Al Noor, R. Hasan, and M.M. Haque. Cellcloud: a novel cost-effective formation of mobile cloud based on bidding incentives. In *IEEE International Conference on Cloud Computing*, pages 200–207, 2014.
- [13] N. Fernando, S. Loke, W. Rahayu, Honeybee: A programming framework for mobile crowd computing, in: K. Zheng, M. Li, H. Jiang (Eds.), *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Vol. 120 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer Berlin Heidelberg, 2013, pp. 224-236.
- [14] N. Fernando, S. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.
- [15] A. Grazioli, M. Picone, F. Zanichelli, and M. Amoretti, “Code migration in mobile clouds with the NAM4J middleware,” in *Proc. IEEE 14th Int. Conf. Mobile Data Manage.*, Jun. 2013, vol. 2, pp. 194–199.
- [16] M. Kjrgaard, H. Blunck, Unsupervised power profiling for mobile devices, in: A. Puiatti, T. Gu (Eds.), *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Vol. 104 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer Berlin Heidelberg, 2012, pp. 138-149.
- [17] C. Yoon, D. Kim, W. Jung, C. Kang, H. Cha, Appscope: Application energy metering framework for android smartphones using kernel activity monitoring, in: *Proceedings of the 2012 USENIX Conference on Annual Technical Conference, USENIX ATC'12*, USENIX Association, Berkeley, CA, USA, 2012, pp. 36-36.
- [18] M. Kim, J. Kong, S. W. Chung, Enhancing online power estimation accuracy for smartphones, *Consumer Electronics, IEEE Transactions on* 58 (2) (2012) 333-339.
- [19] S. Kosta, A. Aucinas, P. Hui, R. Mortier, X. Zhang, Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading, in: *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 945-953.
- [20] C. Shi, V. Lakafosis, M. H. Ammar, E. W. Zegura, Serendipity: Enabling remote computing among intermittently connected mobile devices, in: *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '12*, ACM, New York, NY, USA, 2012, pp. 145-154.
- [21] T. Verbelen, P. Simoens, F. D. Turck, B. Dhoedt, Aiolos: Middleware for improving mobile application performance through cyber foraging, *Journal of Systems and Software* 85 (11) (2012) 2629-2639.
- [22] E. Miluzzo, R. Caceres, and Y. Chen. Vision: mclouds-computing on clouds of mobile devices. In *Proceedings of ACM Workshop on Mobile Cloud Computing and Services*, 2012.
- [23] F. Bonomi, R. Milito, J. Zhu, S. Addepalli. Fog computing and its role in the Internet of Things. In *Proc. ACM MCC*, pp.13-16, August 2012, Helsinki, Finland.
- [24] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, A. Patti, Clonecloud: elastic execution between mobile device and cloud, in: *Proceedings of the sixth conference on Computer systems, EuroSys '11*, ACM, New York, NY, USA, 2011, pp. 301-314.
- [25] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443-461, Jul. 2011.
- [26] L. Zhang, B. Tiwana, R. Dick, Z. Qian, Z. Mao, Z. Wang, L. Yang, Accurate online power estimation and automatic battery behavior-based power model generation for smartphones, in: *Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2010 *IEEE/ACM/IFIP International Conference on*, 2010, pp. 105-114.
- [27] 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.
- [28] A. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikainen, V. H. Tuulos, Misco: A mapreduce framework for mobile systems, in: *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments, PETRA '10*, ACM, New York, NY, USA, 2010, pp. 32:1-32:8.
- [29] A. Shye, B. Scholbrock, G. Memik, Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures, in: *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 42*, ACM, New York, NY, USA, 2009, pp. 168-178.
- [30] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for VM-based cloudlets in mobile computing. In *IEEE Pervasive Computing*, vol.8, no.4, pp.14-23, October 2009.