The Smartphone and the Cloud: Power to the User

Roelof Kemp, Nicholas Palmer, Thilo Kielmann, and Henri Bal

Vrije Universiteit, De Boelelaan 1081A, Amsterdam, The Netherlands
{rkemp,palmer,kielmann,bal}@cs.vu.nl

Abstract. In this paper we study how smartphones can benefit from the resources available in clouds. Unfortunately, integrating smartphones with cloud resources is challenging and comes with dangers for the user in terms of loss of control of applications and data as portions move into the cloud. In this paper we outline our work on the Interdroid project, where we are building a framework for smart applications for smartphones which includes components for integration with the cloud.

Key words: Mobile Computing, Cloud Computing

1 Introduction

In the last decade we have seen a shift in the use of computers. Our personal computers no longer stand on the desks in our homes, but are located in the pockets of our trousers. We use these pocket devices, called *smartphones*, to access our multimedia data, to communicate with our social networks, to navigate to places of interest, to search for online information, and much more.

These smartphones, by nature, are tightly coupled to a person and therefore are usually within arms reach of their owners. They contain personal information, such as the owners social relations, and can detect the context of their owners using various sensors. Being highly interactive for the user and able to communicate with all sorts of other devices makes the smartphone a compelling platform for many personal applications. Unfortunately, smartphones, being small and battery powered, have limitations in computational power, both in terms of processor speed and memory size, as well as space for data storage.

Next to the shift in our personal use of computers, companies have also changed their way of using computers. Techniques such as web services, virtualization, and grid computing stimulated companies to transform their desktop applications into scalable services running on a flexible number of servers. Big companies realized that they could sell some of their compute capacity when their own demand was low[1]. Startups and companies with an irregular compute demand became their customers and a new paradigm was born: *cloud computing*.

In contrast to smartphones, clouds offer a flexible and almost infinite amount of computational power and data storage, not being hindered by batteries, but connected to the power grid. In addition, clouds are permanently online at a fixed address, while smartphones, due to mobility, may suffer from connectivity problems and changing addresses.

Since the shortcomings of the smartphone are the strong points of the cloud, it seems to be a logical step to enhance smartphone applications with cloud computing to reduce the problems related to limited battery, processor speed, memory size, data storage, and changing addresses. Thus, we see the present and future of cloud computing and mobile computing intertwined: today, already many of the rapidly increasing number of smartphones applications are tightly coupled to commercial cloud services [7, 3]. However, while the use of cloud computing reduces the shortcomings of smartphones, the combination of cloud computing and smartphones also introduces new problems for both developers and users, which we detail in the next section. Furthermore, we describe how we address these problems, in order to avoid that the cloud as a solution is worse than the original problem.

2 Cloud and Phone Integration

In order to give a more intuitive understanding of the benefits of cloud computing as well as how problems with such integration arise, we illustrate a typical integration in the following scenario:

2.1 Navigation on a Smartphone

Tom uses his smartphone for navigation. He installs an application which contains map data and an algorithm to find a route. Calculating a route is compute intensive and therefore takes a long time and consumes much battery power. Furthermore, he notices that his map data is soon getting out of date. Therefore, Tom decides to install a new navigation application that uses the cloud. Now, the route calculation runs very fast in the cloud and Tom gets accurate map data on the fly from the cloud. Tom rarely experiences wrong map information, but when he does, he corrects it using his application, which forwards his corrections to the cloud to be shared with other users. Through the use of cloud computing Tom's user experience improves.

For a holiday, Tom decides to go for a trip to a foreign country. While driving through a rural area, he misses a turn. Normally the navigation application would calculate a new route, however, it cannot since there is no network coverage in this area.He finds his way back to the correct road on his own and then enters the foreign country. He immediately switches off his 3G connection, since roaming costs are too expensive. However, this causes his cloud based navigation application to stop working. After coming back from his holiday, Tom finds out that his client application is no longer compatible with the cloud service. He buys an update but then discovers that the company offering the cloud navigation service changed policy and now charges users to get map updates submitted by other users, including those he submitted in the past. Because of cloud computing, Tom's experience on his holiday trip was worse than if he had used his original local application.

2.2 Integration Problems

As shown in the above scenario, cloud computing can both improve and degrade the smartphone's user experience and introduces new problems related to connectivity, application distribution and data ownership. Nevertheless, we believe that cloud computing, when properly applied, will be beneficial to the smartphone users. In this section we will describe which problems cloud computing introduces and our solutions for these problems.

First of all, we note that by using the cloud, applications are transformed into distributed applications running partly on the smartphone and partly in a remote cloud. As a consequence the different components need to connect to each other and to communicate with each other. However, connectivity is not always available and communication costs energy, time and money. Thus, we believe that every application which is a candidate to be distributed, should remain configurable to also run locally to properly handle those situations where the user cannot or does not want to use the cloud. Then, smartphone users can run compute intensive applications fast and energy efficient when cloud resources are available and affordable *and* use slower and less efficient implementations when offline or effectively offline due to roaming.

Another consequence of partitioning applications into a smartphone client and a cloud service is that where users used to own a complete application, now they only own the smartphone client, whereas the service provider owns the cloud service. Compared to a local application, the user now becomes vulnerable to vendor lock in, version mismatches and, more important, no longer has the same guarantees about the applications service level. The user has to trust the service provider, which may turn out to be untrustworthy by stopping or degrading the service, or increasing the price.

We propose to prevent such problems by bundling client and service code together in the smartphone application. Then, the user is protected against version mismatches since he owns both the client and the service code. Although the service code does not run on the smartphone, it can be stored there and, when needed, be installed on a cloud resource controlled by the user. In the above scenario, Tom would have stored the routing algorithm on his smartphone, but could choose to execute it on his own computer or on rented cloud resources.

For using the data services of the cloud, users should have the ability to select which data will be stored on the cloud, and how this data should be secured, in order to protect their own privacy. Programming models dealing with data exchange with a cloud service should always offer both the means to get data on and off the cloud in combination with local replication, so that the user is free to move from one cloud computing provider to another, without being locked into a service. Furthermore, local replication may be required in order for applications to work properly when the smartphone is not connected to the network. Users might even run cloud services on their own home server in order to maintain the privacy of their data, or reduce the cost associated with the service as provided by the cloud vendor.

Finally, we note that not all applications that need to be distributed require a centralized cloud service. Such services are problematic because they introduce a single point of failure in the cloud service. Applications which do not have large computational requirements or data stores associated with them should be implemented in a decentralized way, as we have done in [5], thereby avoiding the centralized point of failure in the cloud. Unfortunately, doing all of these things properly is complicated and so what is needed is a framework which makes

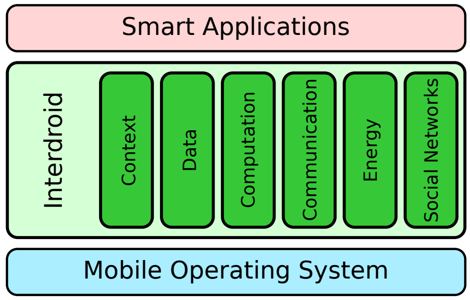

Fig. 1. Abstract overview of the Interdroid project. We strive towards a layer that enables application developers to use social and context information, intensive computing and communication, and data services while keeping energy usage to a minimum.

developing such applications easier. We detail our work on such a framework in the next section.

3 Interdroid: Smart Applications

We have seen that the shortcomings of smartphones can in principle be improved, by the cloud. We believe that a coherent toolkit that guides developers in applying cloud techniques for mobile applications is of key importance in enabling the synthesis of mobile and cloud platforms.

In our research project, called Interdroid, we aim at providing a broad toolkit for application developers to build really smart applications (see Figure 1). We believe that many applications will run distributed and/or use cloud resources to improve their computational speed, battery life, and data distribution. Therefore, part of the Interdroid project focuses on integrating cloud computing and mobile computing. In the next sections we will detail our ongoing work on the components of Interdroid which involve cloud computing.

3.1 Computation in the Cloud

The most obvious combination of mobile computing and cloud computing is probably moving computations from the mobile device to the cloud, as we are working on in the Cuckoo[4] component. Using Cuckoo, compute intensive parts of applications can be executed on cloud resources, a technique known as *computation offloading* [2].

We implemented our computation offloading framework in such a way that both the code that runs on the smartphone and the code that runs on the cloud are bundled together, so that the user owns the full application, instead of just a client. This protects the user from vendor lock-in and forced paid upgrades. Furthermore, application developers using our framework can write two implementations of the same functionality: one that runs on the cloud when connected to it and one local implementation that runs on the phone when the smartphone is offline or roaming. Finally, we added a runtime decision point in our framework, so that intelligence, based on the current context of the user, can be plugged into the system to avoid offloading costs being larger than its gains.

Computation offloading can speedup smartphone applications significantly, particularly if the compute intensive part can be run parallelized in the cloud as we demonstrated with the eyeDentify[6] object-recognition application. Furthermore, it also enables memory intensive applications on smartphones.

While memory size and processor speed on smartphones are still increasing rapidly, the main shortcoming of smartphones is the relative short battery life. Modern smartphone platforms include many optimizations to improve energy efficiency, but the available sensors, radios, the screen and the use of multiple applications at the same time drains the battery rapidly. Unfortunately, battery densities are not keeping pace with power demands of smartphones. Again, computation offloading can lift the burden on the smartphone by offloading energy hungry parts of an application to the cloud [4, 6], provided that the energy saved is greater than the energy spent on communicating with the cloud.

3.2 Communication in the Cloud

There is a particular class of smartphone applications that interact with online information services. Some of these applications monitor information services, such as weather, traffic, stock markets, etc., by periodically polling for new content. Setting the polling frequency involves a trade off between accuracy and high energy usage.

Maintaining high accuracy, but with a low energy usage, can be realized by having the information service push updates, instead of having the application poll for updates. While this approach seems promising, most developers cannot employ it, because they cannot modify the information service.

An application in between the information service and the phone which runs in the cloud can be used to transform a polling smartphone application into one that receives push notifications, without changing the information service. The application in the cloud polls the information resource at a high rate, and upon an update pushes this notification to the smartphone. This way, much of the communication is offloaded to the cloud and a minimal amount of communication, the push messages, are sent to the smartphone.

We are currently implementing a component for Interdroid specifically for this class of applications, which reuses the technique of code bundling as used in Cuckoo. Instead of providing a standalone service in the cloud with no guarantees for the user, we bundle the service and allow the user to deploy the service on their own cloud resources, including a home server, allowing the user to ensure their quality of service and avoiding problems of vendor lock in. Since monitoring applications always involve communication, the applications on the smartphone will not be able to receive updates while being offline. However, the monitoring application in the cloud is permanently online and can store updates and push them to the phone when it is once again online.

3.3 Data in the Cloud

The cloud not only provides abundant computational power, it also has the opportunity to store large quantities of data, which can be used by smartphone applications. Applications which only store data in the cloud can not work properly when the phone is offline as we saw with the navigation application above. However, applications which only store data on the phone can not take advantage of shared data such as the corrections to map data in our example. What is therefore required is a framework which allows users to replicate portions of the data they are going to use on the smartphone while also sharing that data with the cloud. This raises many issues of data synchronization, replication, and versioning as users share their data with each other and with the cloud.

Thus, one of the components of our Interdroid project is Raven, a framework for synchronizing, versioning and replicating data both between smartphones as well as between smartphones and cloud resources. This framework is designed to make it easy for application designers to add versioning and data sharing features to their application, and returns control over data to the user of the application. Users can replicate locally the data they need when offline and can keep the data even if the cloud service vendor goes offline.

4 Conclusions

In this paper we have discussed the relation between smartphones and computational clouds. There are several possibilities for how smartphones can benefit from the power available in clouds, ranging from enhancements of compute power and battery life through computation and communication offloading to improving data services through synchronization, versioning and replication.

Although the usage of the cloud promises the aforementioned improvements for smartphone users, it also comes with threats, introduced by distributing both applications and data. In particular, the user becomes dependent on the cloud service providers in many cases and applications become unusable when the smartphone is offline. We have discussed our work on Interdroid, which makes it easy for application designers to take advantage of the cloud without causing these problems for the user.

We argue that the integration of cloud computing with mobile computing is the future, but that this must be done in such a way that the power of the cloud enhances the mobile experience, while at the same time, the user maintains control of this power at all times.

References

1. Amazon Elastic Compute Cloud Website. http://aws.amazon.com/ec2.
2. B.-G. Chun and P. Maniatis. Augmented smart phone applications through clone cloud execution. In *Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS XII)*, 2009.
3. Gumiyo. http://aws.amazon.com/solutions/case-studies/gumiyo/.
4. R. Kemp, N. Palmer, T. Kielmann, and H. Bal. Cuckoo: a Computation Offloading Framework for Smartphones. In *MobiCASE '10: Proceedings of The Second International Conference on Mobile Computing, Applications, and Services*, 2010.
5. R. Kemp, N. Palmer, T. Kielmann, and H. Bal. Opportunistic Communication for Multiplayer Mobile Gaming: Lessons Learned from PhotoShoot. In *MobiOpp '10: Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, pages 182–184, 2010.
6. R. Kemp, N. Palmer, T. Kielmann, F. Seinstra, N. Drost, J. Maassen, and H. E. Bal. eyeDentify: Multimedia Cyber Foraging from a Smartphone. In *IEEE International Symposium on Multimedia*, 2009.
7. SnapMyLife. http://aws.amazon.com/solutions/case-studies/snapmylife-interview.