Mobile Applications for Tourism

JOÃO ANDRÉ LOPES FERREIRA
IST/University of Lisbon
joao.l.ferreira@tecnico.ulisboa.pt

ALBERTO RODRIGUES DA SILVA
IST/INESC-ID
alberto.silva@tecnico.ulisboa.pt

Abstract

Cloud Computing is a paradigm reaching more and more people and enterprises. When Cloud Computing joins with mobile devices, it becomes ubiquitous and at the same time even more powerful. In this paper, we do an overview on Mobile Cloud Computing and its services, benefits and challenges. We also explain in what consists each service and the differences between Backend as a Service and Platform as a Service. Since applications require the usage of these two services, we explore the differences between native, web, and hybrid applications as well as scenarios in which each of these applications should be used. As a practical example we developed two applications for tourism along with their functionalities and optimisations. The results were obtained by testing the cloud's performance regarding native and web cloud applications.

Keywords: performance, user experience, service model, cloud, platform

1. INTRODUCTION

Tourism is a global leisure activity widely practised. In 2011 there were over 983 million international tourist arrivals worldwide, representing a growth of 4.6% when compared to the 940 million in 2010 [1]. Meanwhile, in this kind of activity mobile devices (e.g. smartphones and tablets) are an highly choice for people to get better experiences. Its powerful and portable capabilities make them greater than any other electronic device when it comes to go outside doors.

Besides that people also tend to do a significant number of activities on their mobile devices (e.g. taking photos, use GPS, listen to music) due of its effectiveness and convenience. Mobile devices’ portability make them more ubiquitous and extremely efficient on the field.

With Cloud Computing being used by mobile devices, it appeared a new way to think and to do things, eliminating some barriers and also leading to the rise of important issues such as privacy, performance or bandwidth of mobile network. So far, we did not find any application for tourism such as TripAdvisor and Foursquare that use service models of Cloud Computing, which are Software as a Service, Platform as a Service and Infrastructure as a Service. They are used in many kind of businesses by various organizations. Recently it arose another type of service, Backend as a Service. Targeting native applications, so they can also benefit from Cloud Computing services. At the same time, Backend as a Service is an alternative to Software as a Service on delivering software business. The goal is to reach every mobile device independently if it is native or a mobile web applications because there is no perfect type of application that can meet all business requirements.

Furthermore, these service models also provide features to make it easier to develop and to allow developers to use either web languages (e.g. HTML5, CSS and javascript) or native languages (e.g. Objective-C, Java and C#) which already have a long struggle. Along this, the issues at stake are “Which model is better to develop a mobile cloud application? Software or Backend as a service? Should the development be done with a web or a native programming language?”. To answer these questions there are important aspects, such as performance and user experience, that need be taken into account.

One of the big challenges that users have to take into account when executing mobile appli-
cations is performance. With the emergence of Mobile Cloud Computing native mobile applications are also facing the same challenges as web mobile applications. And beyond this, performance challenges become even bigger when we have to handle applications targeted for tourism. Because they need to deal with a significant amount of images to meet the users’ expectations. Considering this, it can be hard to know which service model to adopt. A native application can have slowdowns like a web application and by consequence, its performance can no longer be considered as great as before or greater than web application.

Our solution is the development of a native and a web mobile application using services of Microsoft Windows Azure such as Web Site and Mobile Service which correspond to Software as a Service and Backend as a Service. Also, to mitigate network issue we are going to use internal storage or cache of mobile device or browser. And finally, test performance on both applications to check the impact (positive or negative) that Mobile Cloud Computing has.

The structure of this paper is divided as the follow. Chapter 2 provides an overview of Mobile Cloud Computing paradigm including the paradigms of Mobile Computing and Cloud Computing. Chapter 3 shows the services provided by Mobile Cloud Computing as well as the comparison between them. Chapter 4 presents all the benefits that can be explored to improve business and access personal data regardless place and time. Chapter 5 discusses the challenges faced by Mobile Cloud Computing in the subjects of security and network. Then chapter 6 discusses some applications that can be developed in this context and finally chapter 7 describes some related work on the topics of security, HTML5 features, and some other approaches related with Mobile Cloud Computing as well as applications development issues.

2. MOBILE CLOUD COMPUTING

The concept of Mobile Cloud Computing (MCC) comes along with two parts, Cloud Computing (CC) and Mobile Computing (MC). It was due to these two computing paradigms that MCC paradigm has emerged. MC is described as an act of portable devices usage to run stand-alone applications and/or accessing remote applications via wireless networks [8] and CC is defined by National Institute of Standards and Technology (NIST) as “on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service interaction” [13].

The main service models provided by the CC paradigm are: Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS).

Regarding MCC, from our perspective it is just the combination between CC and MC. It refers to an infrastructure where both data storage and (most of) the processing can occur outside of the mobile device or can be distributed between cloud-based servers and mobile devices. Nowadays, MCC is a big bet for organizations due to their willing to adapt to the new technologies and explore their advantages. This trend is changing the way companies work as well as their habits. Next section explore the services brought by CC.

3. SERVICES OF CLOUD COMPUTING

CC can be viewed as a collection of services. The three main service models are SaaS, PaaS and IaaS. These services are built one top of another and thus CC is usually depicted as a layered stack. And with the emergence of a new service designated Backend as a Service (BaaS), sometimes also referred as Mobile Backend as a Service (MBaaS) and which is built on top of PaaS, it leads to a new composition of the layered stack of CC (figure 1).

![Cloud Computing stack]

**Figure 1:** Cloud Computing stack
Although through PaaS we can create and build BaaS and SaaS they are targeted to different end users. BaaS is targeted for application developers and SaaS for non-technical users. In this sense, BaaS and SaaS originate two different layers at the same level (over the PaaS layer).

SaaS model delivers applications through the web [12]. An user does not need to download and install anything onto his personal device once it can access it using a web browser. It brings some benefits to cloud providers (e.g., ability to better monitor and improve their software continuously) and to users (e.g., save disk space, no worries about updates). Popular examples of SaaS are: Google Apps [1], Microsoft Office 365 [2].

BaaS model allows developers to connect/link their native mobile applications to a backend cloud storage and start to enjoy some of the features they provide like push notifications and integration with social networking services [15, 18]. Platforms (such as Google App Engine and Windows Azure) provides these type of service and custom software development kits (SDKs) and application programming interfaces (APIs) that make it easier to develop mobile applications and link them to a backend. Examples of BaaS are: appcelerator [3], appery.io [4].

PaaS model is a CC service that allows developers to create and deploy their web applications quickly and easily using the provided frameworks [25]. This facilitates the complexity of maintaining the software and its infrastructure that supports it and without also doing investments like buying the supporting infrastructure (e.g. operating systems, web servers, database servers). In addition, with PaaS, developers can also create a BaaS and benefit from what BaaS provides. Popular examples of PaaS are: Google App Engine [5], Windows Azure [6].

IaaS model is where we get the fundamental building blocks for cloud services [21, 5]. IaaS is where compute resources such as storage, hardware, servers and network capabilities which are owned and hosted by a service provider are offered to customers on-demand. It is a way to provide the hardware infrastructure (i.e. the equipment) of CC and that is why IaaS is also sometimes referred to as Hardware as a Service (HaaS).

The service provider owns the equipment and is responsible for managing virtualization, servers, storage, and networking while the client has to take care of the data, applications, runtime, and middleware [13, 26, 10, 29].

IaaS is the most flexible Cloud Computing model and allows for automated deployment of servers, processing power, storage, and networking. Popular examples of IaaS are: Amazon EC2 [7], Rackspace [8].

The main difference between the previous services is responsibilities distributed by clients and vendors as we can see in table 1.

BaaS does not appear in the table because BaaS and PaaS have the same responsibilities. BaaS is very similar do PaaS. Both are targeted to developers and provide frameworks to help them developing their application, i.e. it speeds up the application development process. The difference between them is whether an application is deployed into the cloud or connect to a backend cloud storage. Where both diverge are the location of the application and data (text and images). To help to understand, we have figure 2 and which represents the following scenario: a business entity wants a mobile application to, at least, store and get information from the cloud.

5. https://developers.google.com/appengine
<table>
<thead>
<tr>
<th>Responsibility</th>
<th>SaaS</th>
<th>PaaS</th>
<th>IaaS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Runtime</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Data</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Middleware</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OS</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Security</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Virtualization</td>
<td>-</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Servers</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Storage</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Networking</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1: Responsibilities of vendors and clients [3].

4. **Benefits**

CC is changing the way companies work and the people’s habits. MCC helps to break some barriers to mobile users as well as to mobile business companies [7].

From Cloud Computing perspective, companies are now looking for how to take fully advantage of CC technology in order to have more engagement with their customers. Using this technology properly can bring benefits such as (1) cost savings, where they do not need to acquire the right and necessary (amount) equipments to always provide a good service to customers regardless the peaks of load, (2) storage, which is on-demand without worrying about storage space (3) ubiquity in data access, where users can access their data anywhere and whenever they want, and (3) deployment, making it easier for their software.

From Mobile Cloud Computing perspective, apart from the benefits previously mentioned for mobile devices, they can also save energy and offload computation [28, 17]. However, these benefits will depend on the amount of computation and communication required.

5. **Challenges**

CC also brings some challenge related with security [16, 27] and network [31]. On security side, there is a lack of knowledge from end-users who are the most responsible to avoid...
some security risks such as privacy and confidentiality, data integrity, user authentication, communication quality. On network side, there is a long way to run regarding communication quality which strongly depends on bandwidth of the mobile network and connectivity is unpredictable, and 4G network where there is still a need on upgrading technology in order to reach it.

6. MOBILE APPLICATIONS

Regarding mobile applications that can be developed, there are three types: native, web, and hybrid.

Native applications are those which are developed by targeting a single operating system (OS) using the specific languages and SDKs (e.g. Java for Android, Objective-C for iOS, or C#/.NET for Windows Phone) provided by some IDEs such as Eclipse, Xcode, or Visual Studio. The advantages of this type of application are: best performance, full access to the features of a specific device, and better user experience. Mobile Web applications are canned in a way that they look like a mobile application. They are usually developed using PHP, Node.js, ASP.NET, HTML, CSS, or JavaScript. Though being cross-platform, there are some limitations that remain such as the access to the native device functionality (e.g., GPS localization and camera). Hybrid application is defined as a web application and then wrapped inside of a native container that provides access to the native features.

HTML5 has brought a surge of on-mobile features access such as GPS location, audio and video capturing, and accelerometer. Also, HTML5 can give a very similar look and feel of what users would find in a native application [23]. In addition, HTML5 provides features for offline navigation. However native applications are the best for performance and on the access to on-device features.

There are many experts that have the same opinion, such as Telerik that says on its survey that “There are no ‘one-size-fits-all’ solutions for mobile application development” and Todd Anglin said that “Many developers are finding the choice between native and hybrid approaches is dependent on business needs, application requirements, developer skill, development time, and other factors”.

It is important to understand the differences between native and mobile web application in order to better decide which one to use. To support the best choice, developers have to answer the following 5 essential questions [6]: (1) How many mobile devices platforms do you want to support? (2) Do you have a “bring your own device” (BYOD) [4, 22] strategy? (3) How many on-device features do you need to access? (4) How important is security and/or performance? (5) What is the purpose of the application?

Now with MCC, native applications will require network connection as well. Since the network is unpredictable (as we mentioned in section 5) it is hard to measure the performance of one application in these situations. Therefore, performance is not only an issue for web applications but also for native applications using external resources such as backend cloud storage, i.e., native applications using BaaS. From this reasoning, we can describe the following two scenarios:

Without BaaS. If the first priority of a business application is performance or on-device features, then native application is the way to go. But if cross-platform is a concern as it will certainly be in corporations implementing BYOD strategies, then mobile web applications are a better choice.

With BaaS. With BYOD strategies we keep the same thought as we described before. However, performance is more complicated to evaluate since both applications requires network. Therefore, it is necessary to regard another issues such as on-device feature. If this is also a priority, then native application is the option to take otherwise, it is the web application.

7. RELATED WORK

Hoang et al. give an overview regarding MCC, starting with some different definitions for MCC that exist, which are very similar to each other
The paper also gives a notion of the service models of CC. However, they do not show the differences between the different services and do not present BaaS since it is a recent service model. MCC applications are introduced for different business models (e.g. mobile commerce, mobile learning, mobile healthcare and mobile gaming) and the purpose of applying MCC to the applications for those business models. They analyses some approaches to deal with offloading in two different types of environments: static and dynamic (when a user is on the move and the connection or bandwidth of his mobile device is changing). Niroshinie et al. discuss some methods of offloading tasks (e.g. how to tackle issues related to the distance between the device and the cloud and the heterogeneity) [9].

Hoang et al. also analyse security regarding two categories: (1) Security for mobile users, where it shows approaches on how to protect mobile devices from possible threats and how to avoid the leak of user’s personal information and (2) Security data on clouds, where it presents methods to secure data access and solutions proposed to address the integrity and digital right management issues. Khan et al. focus more on security issues. They show numerous challenges in this field and some security frameworks for MCC [14].

The survey presented by Niroshinie et al. explains the need of using MCC nowadays by exploring the different applications (such as image processing, natural language processing, crowd computing, sharing GPS/Internet data, sensor data applications, multimedia searches and social network) and scenarios [9].

Amatya and Kurti present a survey where it shows the trends on the mobile development in cross-platform, mainly the potential of two approaches which are web-based approach and hybrid approach [2].

Appcelerator published a white paper that debates native and HTML5 development. It gives some highlights of statistics of Native and HTML applications in production. It describes some aspects that are important for successful applications such as: user experience, performance, monetization, cross platform deployment costs, fragmentation, availability of programming expertise, importance of immediate updates and distribution control, timeliness of few OS innovations and security. It also introduces a tool that helps developers to deliver cross-platform applications without much effort [30].

Also, a paper of Ogunlolu et al., based on their development costs, cross platform capabilities, application performance and development cycle, shows a comparison between web mobile applications (HTML5) and native applications (Android OS, iOS and Windows Phone OS). It also does a brief look to some features (e.g. offline, geolocation, real time communication and connectivity and multimedia) that come with HTML5 and not with HTML4 which leverage more power and efficiency to web applications [24].

Madaudo and Scandurra present some tools for application development using both approaches cross-platform and native application development [20]. Also, together with Li and Powell, they show some advantages and drawbacks of each approach for each aspect (e.g. user experience, performance, device-specific features) [19].

8. APPLICATIONS DEVELOPED

We developed a native mobile application and a web mobile application, named as NativeMAT and WebMAT respectively, and connected to Windows Azure platform.

Windows Azure is a PaaS from Microsoft. To develop application that interact with this platform, Microsoft provides two tools: WebMatrix and Visual Studio. However, depending on what type of service the developers want to use, there are also other tools. For instance, if they want to use Mobile Service, Eclipse and Xcode are some other possible tools.

Windows Azure Portal helps developers to easily create and manage their services. A good way to test load capacity of Windows Azure is to develop a web mobile application hosted within
its Web Site service. Microsoft provides the tool WebMatrix to facilitate both development and deployment to Windows Azure. The services are categorised in the categories Compute, Data Services, App Services, Networks, and Store. We used Web Site service (for web applications) and Mobile Service (for native applications) from Compute category and the SQL Database service from Data Services category.

WebMatrix tool allows developers to develop their applications (1) locally and then deploy to Windows Azure and (2) remotely, where changes are applied when saved.

Before start developing any application we imported a SQL database through SQL Database 2012 tool once the SQL Database service only accepts BACPAC files. If we did not have this database we could create it from scratch using the Windows Azure database server portal that works as SQL Database tool. Through, SQL Database service we can see details about our database as well as monitor it and do some configurations.

Windows Azure Web Site service was used to host WebMAT. It provides important functionalities such as scaling, monitoring and an overview of service details. Through them, we can control everything we need to observe what is happening to our service and update it (if necessary) to provide better service.

In our project we used HTML, CSS, PHP, and JavaScript languages to develop WebMAT. Plus, jQuery Mobile as the framework to provide a better user experience. To make WebMAT communicating with our database, SQL Database service provides the PHP connection string. But first, it is required to link these two services by using the Web Site service. Regarding JavaScript, we also used it to trigger actions only when a button is clicked. AJAX code was embedded in some JavaScript functions to help us to exchange data with the server and update only a set of the page instead of the whole page. jQuery is one of the most popular frameworks used to built out complex web mobile applications. Therefore, jQuery Mobile was used in our project to make responsive web mobile applications that are accessible on all smartphones, tablets and desktops and to try to give a closer look and feel to the native mobile application environment.

Other than that, HTML5 APIs cover a set of functionalities and features that extend the power of a web application and help developers to improve their web mobile applications in aspects like performance and user experience. It is through some of these features such as App Cache and Web Storage that it is possible to make the web application work offline. Beyond that, App Cache can also improve performance dramatically once it provides (1) offline browsing, (2) speeds up the web application, and (3) reduces servers’ load. Web Storage gives the opportunity to store data locally in the user’s browser. This kind of method of storing data is more secure and faster than the one with cookies that are transmitted through HTTP headers without being duly encrypted. Web Storage is divided into two parts for storing data: localStorage and sessionStorage. The difference between these is how long the data remains stored. We used localStorage because it always keeps the data until the developer think it is the right moment to clean that data.

Mobile Service serves to create a BaaS and afterwards link our Android application to it. One of the features we used was authentication, which can be provided from different providers such as Microsoft, Facebook, Twitter, and Google. Our service requires the linkage to our SQL Database service as well once our database is relational, otherwise, it would not be necessary because Mobile service has already its own storage. To make our service able to communicate with our database, the schema name has to be the same as in our Mobile Service. Another step that is required to detect the tables in our SQL Database service is to create the tables with the same names, which will force to detect them. In addition, a Storage service was needed to save and get images as blobs.

In addition, we optimised our applications for both user experience and performance. Google’s PageSpeed was one of the tools used that gave us some useful suggestions such as
turning external resources into internal, leverage browser caching, and images optimisation.

Images are crucial for tourism applications and responsible for most downloaded bytes. The reason why it is extremely important to optimise them. We did a few experiments with kraken.io, JPEGMini Lite and Preview tools to verify which one was better. And for the reasons of non-limitation on the level of compression and on the size of the input images, we used Preview. Through this tool, we were able to save a significant amount of space from 289 MB to 52 MB, a reduction around 81.97%. We also implemented other methods such as Infinite Scrolling and Asynchronous Task to spare CPU on the amount of processing load of the GUI thread and to reduce the waiting time of the user.

9. Evaluation

The tools used to perform tests on Windows Azure and our applications were Apache JMeter and Eclipse. Through Apache JMeter, it was possible to evaluate the behaviour of our Windows Azure Web Site service. It provided several and important elements to put in our test plan such as (1) thread groups, which specify the number of threads that will send requests, (2) logic controllers, to define the order of requests processing, and (3) listeners, to show the results in a graph or tree.

Still, we added a non-test element to the workbench of our test plan and created a proxy in the network settings of our computer in order to start recording all the actions we do in the browser and make the test plan easier.

Regarding Eclipse, it has a tool called Dalvik Debug Monitor Server (DDMS) for assisting developers with debugging, monitoring, and profiling process. We used Systrace provided by DDMS to track all the actions we did in the application and to specify which action we want to track in order to be easier to analyse and understand the application performance.

10. Results

To verify load capacity of Windows Azure and analyse the performance of WebMAT, we increase the number of users that send a set of requests, i.e., we perform tests on JMeter with 10, 100 and 1000 of users. Each test was performed 5 times to make sure the results were consistent. The scenario used was the same for all the tests: A user has to choose the first PoI (within Castle category) in the list and add it to the list of favourite PoIs.

The first test with 10 users was divided into a test with cache and other without cache. The results, quite different in the duration and number of requests, can be seen in figure 3. As expected, the test with cache (figure 3b) is faster than without cache (figure 3a) once it has less requests to process. This way, we can spare work on both client and server sides. We can also see in both graphs that there is no performance degradation and bottleneck.

Still, tests with 100 and 1000 users (larger scale) present the same behaviour as the previous two tests. The throughput of all tests increased on a logarithmic scale, meaning that there is a point in time that the throughput starts to stagnate and keeps that pace until there is no more requests to process. Moreover, we can think that it is normal that the first test last more than the second once it has more requests, but this is not always true.

<table>
<thead>
<tr>
<th>Test</th>
<th>Duration Time (Average)</th>
<th>Internal Server Errors (Average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Users</td>
<td>2m38s</td>
<td>1</td>
</tr>
<tr>
<td>(without cache)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Users</td>
<td>47s</td>
<td>1</td>
</tr>
<tr>
<td>(with cache)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100 Users</td>
<td>54.8s</td>
<td>17</td>
</tr>
<tr>
<td>1000 Users</td>
<td>3m32s</td>
<td>140</td>
</tr>
</tbody>
</table>

Table 2: Averages of time and errors obtained in each Test.

For example, if we take a look to the table 2, we can see that the test with 100 users is faster
than the test with 10 users without using cache where they process 9’000 and 2’700 requests respectively. The reason of this happening is because of the type of requests we sent to server. The resources we keep in cache are also images. As we mentioned in the section § images represent the most downloaded bytes on a page. Therefore, WebMAT does not need to waste time downloading images that already are in cache. This way, not only app cache spares an heavy work for our desktop computer but also to the server.

Another thing that we can see in the table 2 is the internal server errors that increase when the number of users grows. For example, the last test lasted 3 minutes and 32 seconds and got 140 internal server errors in 46’000 requests. Yet, this ratio of failed requests is acceptable once it is perfectly normal to happen in real-life. Beyond these tests with JMeter, we did tests manually in different browsers and devices to verify the application responsiveness and speed. WebMAT was faster on a desktop computer than on a smartphone with 12,85 seconds and 16,17 seconds respectively on average. The values were not a surprise since mobile devices are powerless and more susceptible to network issues. However, both devices were in the same conditions (place and network signal) and the results we got are with the best scenario (with a high signal of network).

Regarding the developed NativeMAT, we got on average 27,66 seconds on the first experiment (without images optimisation) and 21,25 seconds on the second experiment (with images optimisation). Asynchronous tasks are targeted for long-running or CPU-intensive tasks such as tasks that are responsible to fetch images. Although we use asynchronous tasks, the two experiments only start to diverge when the application starts to draw frames of the PolsâˆŻ images, which run inside of the UI thread’s context. This way, it will compromise the responsiveness of the application. For example, perform-Traversals blocks is where we can see how long the application spent drawing a frame (figure 4). Considering this, we would say that images optimisation is as important as running tasks in
background when we are dealing with an application that has to handle many images.

11. CONCLUSION

Mobile Cloud Computing brings with itself some changes to mobile applications. With it, it is required network connection which compromise mobile devices’ performance once it is unpredictable. This way, both web and native mobile applications may suffer from this network issue. However, in order to understand the impact it can cause to both we developed these two types of applications using Windows Azure.

Windows Azure is a good platform for (mobile) developers to create application towards to Mobile Cloud Computing paradigm. It provides a variety of services according to different needs and requirements as well as the required SDKs for mobile application to be able to communicate with its services. The services we used to communicate were Web Site and Mobile Service that let us to create an SaaS and a BaaS respectively.

In this sense, we developed and deployed an HTML5 application (WebMAT) to our Web Site service using WebMatrix tool, and an android application (NativeMAT) linked to our Mobile Service. Through these applications, we did some tests that showed that there is no performance degradation and bottleneck from Windows Azure when increasing the number of users and requests. Moreover, we obtained times by both WebMAT and NativeMAT that bring another meaning to the though that performance of native mobile applications is better than web mobile applications. Since native applications had all the data locally, they did not depend on external resources, which result in a remarkable performance.

Now with Mobile Cloud Computing native applications have significant performance loss once that they are worse than web applications in the tests we have effectuated. One of the reasons for this is the progress reached with HTML5 that can leverage performance significantly by using app cache (one of its features). Yet, the most important reason is image rendering. As we verified in section 10, NativeMAT spent more time on rendering once this task had to done in the main thread, which can delay the current task and the followings. In regard to mobile applications for tourism (having many images), we conclude that web applications can have a better performance than native applications.
REFERENCES


