Optimizing the Performance of Mobile Web Application

Kavindra Kumar Singh*, Dr. Praveen Kumar

* Research Scholar, Shri Venkateshwara University, Gajraula, Uttar Pradesh, India

Mail: singhkavindra@rediffmail.com Contact no.:+91-8559874775

Abstract - For any enterprise that uses a website to build brand awareness and sell its products, the explosive growth in mobile devices is impossible to ignore. But while many companies would love to extend their website or e-Commerce application to a mobile audience, they're often uncertain about how to proceed. Does it will affect performance? This paper examines a topic at the heart of any mobile web application i.e. performance optimization. With today's people carrying a vast array of mobile devices that operate across a massively fragmented and shifting market, institutions can find themselves wondering how to deliver content and services specifically designed for mobile use most effectively. The usefulness of mobile devices has increased greatly in recent years allowing users to perform more tasks in a mobile context. This increase in usefulness has come at the expense of the usability of these devices in some contexts.

Keywords— Mobile Web Application, HTML5, Optimization, Execution Efficiency, Hybrid Apps, Native Apps, Smart Phone

INTRODUCTION

There are 6.8 billion people on the planet, 5.1 billion of whom own a cell phone. And today, an ever-growing percentage of these devices are smartphones. According to a recent Pew Research Center Study, the number of users accessing the Internet on their smartphones has more than doubled in the past 5 years, as has the number of users downloading and using mobile apps. Of those who use the Internet or email on their phones, more than a third go online primarily through their handheld devices.

Recent times have seen the field of mobile technology grow exponentially, leading to institutions increasingly recognizing the importance of delivering content and services to users through their mobile devices. In many cases these can simply be delivered using the web, optimizing your websites for use on smaller screens. However, in some cases you may wish to deliver a service that takes advantage of the native capabilities of today's powerful smartphones, such as GPS for location based services for example.

The development of mobile applications that can run across multiple heterogeneous devices is challenging. Not only do mobile devices differ considerably at the hardware level, but the software development environments are also very different. Generally speaking, there are two approaches to mobile app development: Web-based, which involves technologies such as HTML5, CSS, Javascript, and related frameworks; and development on native platforms, such as iOS, Android, and Windows Mobile 8.

One of the main advantages of native app development is the ability to reach hundreds of millions of customers simply by uploading your app to a store. Apps developed in a native platform technology currently outpace Web-based alternatives in both the number of available apps and the time spent by users on the device. That being said, the main advantage of the mobile Web approach is its rapid deployment model and its ability to run immediately on multiple platforms via a Web browser. Many developers see the ability of Web-based apps to circumvent the somewhat formal, and often lengthy, process required to deploy apps in a store as a huge benefit. However, Web-based solutions suffer from browser incompatibilities, an uncertain monetization strategy, and slow evolution of mobile Web development standards.

A good understanding of the advantages and limitations of both development approaches will be important to successful application deployment and a positive user experience. Many developers see hybrid approaches as a natural migration path for developing cross-platform code that can run in a device-independent way across multiple hardware platforms. Companies such as PhoneGap and AppMobi are selling hybrid cross-platform solutions using the HTML5 programming model, thereby leveraging Web technologies that developers already know.

MOBILE WEB APPLICATION

As is true with most technology selections, there's no one-size-fits-all answer when it comes to the type of mobile app to develop. There are numerous web app best practices to consider, not all of which are technical. Who is your target audience? Are they more likely to prefer a mobile web or a native app? What development resources do you have and which mobile technologies are they most familiar with? What is the licensing and sales model that you're envisioning for your product?

Generally speaking (although there are always exceptions), the mobile web route is faster and cheaper than the native app route, especially when the objective is to support a wide range of devices. Conversely, there may be capabilities native to the mobile device (such as the movement sensor and so on) that are essential to your app, but which are only accessible via a native app (which would therefore make the mobile web app choice a non-starter for you).

And beyond the web vs. native question, a hybrid app may be the right answer for you, depending on your requirements and resource constraints. Hybrid apps, like native apps, run on the device itself (as opposed to inside a browser), but are written with web technologies (HTML5, CSS and JavaScript). More specifically, hybrid apps run inside a native container, and leverage the device's browser engine (but not the browser) to render the HTML and process the JavaScript locally. A web-to-native abstraction layer enables access to device capabilities that are not accessible in mobile web applications, such as the accelerometer, camera, and local storage.

But whatever choice you make – whether it be mobile web, native or hybrid app – be careful to adequately research and confirm your assumptions. As an example for the purposes of this mobile web app development concept, you may have decided to develop a native app for e-commerce to sell your products, but according to *Hubspot*, 73% of smartphone users say they use the mobile web more than native apps to do their shopping… so you may have bet on the wrong horse.

But whatever choice you make – whether it be mobile web, native or hybrid app – be careful to adequately research and confirm your assumptions.

And then, of course, there are the practical considerations of time and budget. As one of my favorite sayings goes, "faster, better, cheaper… pick any two". While time-to-market and cost constraints are of paramount importance in web application development, it's crucial not to compromise too heavily on quality in the process. It's quite difficult to recover the confidence of a user who has had a bad first experience.

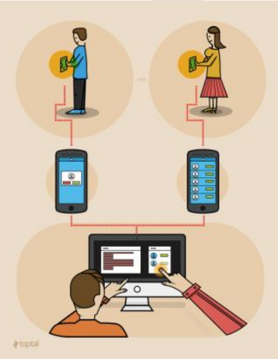

Figure 1.1: Mobile Web Application Used by Users

LIMITING FACTORS FOR MOBILE WEB APPLICATION PERFORMANCE

No matter how interesting, beautiful, or cleverly interactive your Web pages are, if they take more than two or three seconds to render, whether on a desktop or a mobile device, users quickly become impatient. They are measurably less likely to convert from browsing to buying and may even hit the back button or close the browser before the page ever loads.

Even delays of less than one second significantly affect revenues. In 2006 Marissa Mayer, with Google at the time, recounted that, after users indicated they wanted to see more than 10 search results per page, Google experimented with showing 30 instead. To Google's surprise, traffic and revenue dropped by 20 percent in this experiment, apparently because the pages with more results took just an extra half-second to load.[5]

User expectations have only escalated since then. A 2009 study by Forrester Research on behalf of Akamai identified two seconds as the threshold for acceptable Web-page response times and found that 40 percent of consumers abandon a page that takes longer than three seconds to load. Just one year later, another study done for Akamai found that the number of users who abandon a page after three seconds had risen to 57 percent.

Furthermore, users on mobile devices expect performance to be at least as good as if not better than what they experience on their desktops. The Harris Interactive 2011 Mobile Transactions Survey, commissioned by Tealeaf Technology (now part of IBM), reported that 85 percent of adults who had conducted a mobile transaction in the previous year expected the mobile experience to be equal to or better than shopping online using a laptop or desktop computer, and 63 percent said they would be less likely to buy from the same company via other channels if they experienced a problem conducting a transaction on their mobile phones.[10] In other words, poor mobile performance hurts companies on all other platforms, including brick-and-mortar.

Mobile traffic is expanding rapidly. For many consumers, their phone or tablet has become their primary portal to the Internet, but performance is falling short of expectations. A study published by Equation Research on behalf of Compuware in February 2011 found that almost half (46 percent) of mobile users said Web sites load more slowly than expected on their phones. Nearly 60 percent expect pages to load in three seconds or less, and 74 percent report they would leave a site if a single page took five seconds or more to load. A 2012 study of 200 leading e-commerce sites by Strange loop Networks (now part of Radware) found that the median load time was 11.8 seconds over 3G (Figure 1.2); performance over LTE fared only slightly better, at 8.5 seconds.[8]

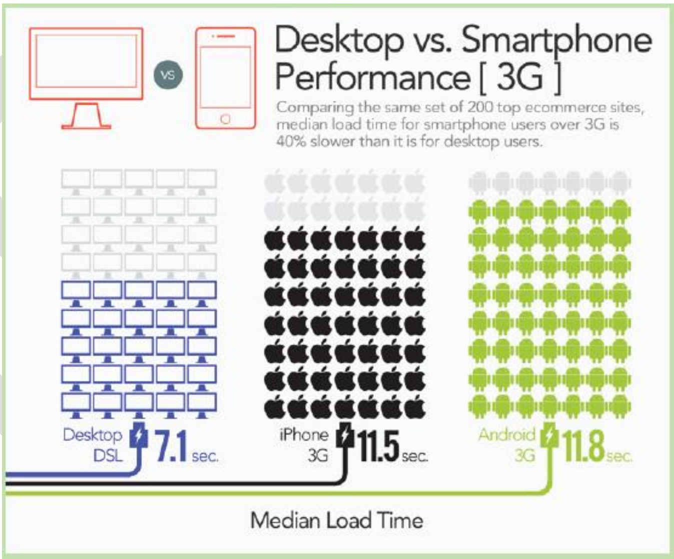

Figure 1.2: Median Load Times for Desktop and Mobile Devices

i. Three Limiting Factors for Mobile Performance

As already mentioned, mobile devices have inherent performance limitations: lower bandwidth, smaller memories, and lower processing power. These challenges are compounded by external issues, notably:

Web pages are bigger than ever. According to the HTTP Archive, the average Web page carries a payload of more than 1 MB and contains at least 80 resources such as images, JavaScript, CSS (Cascading Style Sheets) files, etc. This has a significant impact on desktop performance. Its impact on mobile performance - and particularly on 3G performance is much more dramatic. This impact will be felt even more keenly over the next three years. At the current rate of growth, average page size could surpass 2 MB by 2015.

Latency can vary widely. It can range from as little as 34 ms for LTE to 350 ms or more for 3G. Mobile latency is consistent only in its inconsistency, even when measured at the same location. This is due to a number of variables beyond the amount of data passing through the tower. Factors such as the weather, and even the direction the user is facing, can have a significant impact.

OPTIMIZING THE PERFORMANCE OF MOBILE WEB APPLICATION

Only 20% of the time required to display a typical Web page, whether in a desktop or mobile browser, is consumed by loading the page's HTML. The remaining 80% is spent loading the additional resources needed to render the page - including style sheets, script files, and images - and performing client-side processing.

The three main strategies for improving performance are:

- Reducing the number of HTTP requests required to fetch the resources for each page.
- Reducing the size of the payload needed to fulfill each request.
- Optimizing client-side processing priorities and script execution efficiency.

Because mobile networks are usually slower than those available to desktop machines, reducing requests and payloads takes on huge importance. Mobile browsers are slower to parse HTML and execute JavaScript, so optimizing client-side processing is crucial. In addition, mobile browser caches are much smaller than those of desktop browsers, requiring new approaches to leveraging local storage of reusable resources.

You must think carefully about how to reduce and optimize each byte and server transfer to reduce the user's wait time. Google found that, for every extra 5 seconds of load time, traffic dropped by 20% (and it is also worth noting that search engines look at load times as part of their calculation of page quality score). *60% of mobile web users say they expect a site to load on their mobile phone in 3 seconds or less.* Here are a few tips that can help optimize the performance of your mobile web app and minimize latency:

- ***Image Optimization.*** Image load time is well-known to be one of the biggest performance issues affecting page load on mobile devices. Use of online image optimizers, such as *smushit.com*, can be helpful in addressing this issue.

- ***Code compression.*** Compressing your JavaScript and CSS files, depending on the amount of code you have, can potentially have a significant impact on performance. A useful tool for compressing your code is *refresh-sh.com*.

- ***Database queries.***
  - Some mobile device browsers don't accept as many cookies as desktop browsers do, which can result in the need to execute even more queries than usual. Server-side caching is therefore especially crucial when supporting mobile web app clients.
  - Remember to employ the appropriate filters to preclude SQL query injection that could otherwise compromise the security of your site and server.

- ***Content delivery networks (CDN).*** If you are planning to provide lots of videos, images, audio files, or other types of media, use of a CDN is highly recommended. Some of the more common commercial CDNs include *Amazon S3, Microsoft Windows Azure, and MaxCDN.* The advantages of using a CDN are numerous and include:

- *Improved download performance.* Leveraging a CDN's resources enables you to distribute load, save bandwidth, and boost performance. The better CDNs offer higher availability, lower network latency, and lower packet loss. Moreover, many CDNs provide a globally distributed selection of data centers, enabling downloads to occur from a server closer to the user's location (resulting in fewer network hops and faster downloads).
- *More concurrent downloads.* Browsers typically limit the number of concurrent connections to a single domain, after which additional downloads are blocked until one of the previous downloads has completed. You can often see this limit in action when downloading many large files from the same site. Each additional CDN (on a different domain) allows for additional concurrent downloads.
- *Enhanced analytics.* Many commercial CDNs provide usage reports that can supplement your own website analytics and which may offer a better quantification of video views and downloads. *GTmetrix*, for example, has an excellent website reporting tool for monitoring and optimizing the sources loaded on your site.

If you want to use some of the cool HTML5 stuff, remember to verify in advance that the functionality you're looking for is supported across the device landscape that your customers are likely to be using. For example, in iOS 6 and above, there is no support for the navigator getUserMedia functionality since the camera is only accessible through native apps. Two great resources for checking what's supported on specific devices and browsers are *caniuse.com* and *html5test.com*.

Remember to verify in advance that the functionality you're looking for is supported across the device landscape that your customers are likely to be using.

CSS3 media queries can also help you provide customized content for each device. Here's some example code for capturing different device characteristics, such as pixel density, screen resolution, and orientation:

```
/* For lower than 700px resolutions */
@media ( max-width : 700 px) { ... }
/* Same as last but with the device orientation on land scape */
@media ( max-width : 700 px) and (orientation: landscape) { ... }
/* Including width and orientation you can add a media type clause,
  in this case 'tv' */
@media tv and ( min-width : 700 px) and (orientation: landscape) { ... }
/* for low resolution display with background-image */
. image {
   background -image : url(/path/ to /my/image . png);
   background -size : 200 px 300 px;
   height: 300 px;
   width: 200 px;
}
/* for high resolution (Retina) display with background-image */
@media only screen and ( min-- moz -device-pixel-ratio : 2 ),
only screen and ( -o-min-device-pixel-ratio : 2 / 1 ),
only screen and ( -webkit-min-device-pixel-ratio : 2 ),
only screen and ( min-device-pixel-ratio : 2 ) {
```

```
-repeat ;
  background -size :  200 px  400 px;
/* rest of your styles... */
  }
}
```

CONCLUSION

The paper has discussed the mobile web application with respect to the development of mobile web applications and performances. We identified and discussed some of the limiting factors those affect the performance of mobile applications. Also, this paper suggested how to optimize the performance of mobile web applications. This paper is given some of the examples in terms of source code for capturing difference device characteristics, such as screen resolution, orientation, and pixel density. The final goal of this research paper is identifying major factors those affects the performance of mobile web applications and linked those challenges to improve the quality features.

REFERENCES:

[1] Ali Mesbah, Arie van Deursen, “An Architectural Style for AJAX”, Proceedings of the Working IEEE/IFIP., Conference on Software Architecture (WICSA'07).

[2] Bixby, J. 2012. 2012 state of mobile e-commerce performance;http://www.strangeloopnetworks.com/resources/videos/case-studies-from-the-mobile-frontier-the-relationship-between-faster-mobile-sites-and-business-kpis-video/.

[3] Everts, T. 2013. Case study: how effective are CDNs for mobile visitors. Web Performance Today; http://www.Webperformancetoday.com/2013/05/09/case-study-cdn-content-delivery-network-mobile-3g/.

[4] Fioravanti, R. 2011. Creating fast buttons for mobile Web applications. Google Developers; http://code.google.com/mobile/articles/fast_buttons.html.

[5] http://www.akamai.com/dl/whitepapers/Akamai_PCW_Travel_Perf_Whitepaper.pdf.

[6] mod-spdy; http://code.google.com/p/mod-spdy/.

[7] Prasad Pooja, July – 2013, Choosing the Right HTML 5 Framework: To Build Your Mobile Web Application, A RapidValue Solutions Whitepaper.

[8] PhoCusWright. 2010. PhoCusWright/Akamai study on travel site performance; http://connect.phocuswright.com/2010/06/phocuswrightakamai-study-on-travel-site-performance/

[9] Radware. 2011. Case studies from the mobile frontier: the relationship between faster mobile sites and business KPIS;http://www.strangeloopnetworks.com/resources/research/state-of-mobile-ecommerce-performance/.

[10] Tealeaf. 2011. Report on the Mobile Customer Experience. Based on the Harris Interactive 2011 Mobile Transactions Survey.

[11] W3C. 2012. Network Information API; http://www.w3.org/TR/netinfo-api/.

[12] YUI. ImageLoader. Yahoo! User Interface Library;http://yuilibrary.com/yui/docs/imageloader/