

The Advancement of Web Applications for Mobile Devices

ABSTRACT

Nowadays smartphones have been an integral part of human life with us carrying these devices anywhere and everywhere. They were initially built to access people through calls and texts, but the devices powered now are far more capable than any computer built at the beginning of the Computing Era. But, building websites that function well on mobile devices still remains a challenge. Mobile Web applications are essential in both the technological and commercial workspace. There are a variety of methods for creating mobile Web apps, but it's important to comprehend the fundamental technologies given the rapid growth of mobile software and new devices are constantly released.

INTRODUCTION

Web environments originally served as a simple platform for viewing documents, but they quickly developed into a host of complex software systems that can support both human-machine and machine-machine interactions through a network. The growth of mobile computing has caused the most significant change in web development in recent years. The web browser is used as a middle level of abstraction in the web-based application paradigm, which offers a framework for creating applications that can be quickly deployed online.

However, despite these significant advancements, mobile web browsing is frequently annoying. In contrast to other

possibilities, which are still in the development stage, the recent boom of mobile telephony in the Developing World provides a tremendous opportunity to deploy services now. In emerging nations, mobile phones and related services boost social progress and productivity. Creating and successfully promoting an application in a field that is constantly evolving, such as mobile technology, is not an easy task. Numerous platforms and their potential users must be taken into account by developers.

The Internet experience on mobile devices is fairly quick thanks to the iPhone and 3G connection, but the Web was developed at a time when the bandwidth available to desktops increased annually, which led to Web sites and applications getting bigger and utilizing more resources like CSS, JavaScript, images, and video.

THE PROBLEM?

THE LATENCY PROBLEM-

Latency, or the delay between a request and a response, is one of the main problems with mobile Web performance. The term "bandwidth" refers to the amount of data that may be transferred via an Internet connection in a given amount of time. Although consumers' connections are technically capable of managing it, latency is what hinders them from receiving that optimal bandwidth.

Wired Latency:

There is some lag on any Internet connection. Data is received pretty rapidly because wired connections have reduced latency since the cable passes more directly between sites. The electrical resistance of the wire material is the main contributor to the delay in this situation. Unless the cable has been destroyed, that is often insignificant. Otherwise, a cable connection's latency is rather consistent across time. Your wired network connection may be slow because everyone in your area is using the Internet simultaneously or because a large number of family members are consuming a lot of bandwidth.

Wireless Latency:

Wireless Internet connections have a delay and can be interfered with by walls, microwaves, radios, and other electromagnetic or physical obstacles. To find out how much delay impacts a connection's throughput, Tom Hughes-Croucher performed a simulation. He discovered that the throughput was reduced by roughly 67 percent with just a 50 ms increase in delay. The number of requests that are successfully processed during the loading of a typical Web page is crucial since a wireless connection has a high delay. The request must first travel from the phone to the closest cellular tower, then to a GPRS server operated by a mobile service provider, and lastly, it must travel all the way back to the phone after receiving a response from the Internet. A request from Las Vegas to a resource in California really goes to California first before returning to the device since mobile networks only have a small number of GGSNs.

THE SOLUTION:

Improving Web Performance-

When working on mobile projects, developers should account for the reality that latency will always be an issue in wireless connections. A Web site or application should make as few HTTP requests as feasible to reduce latency. The fewer queries made to the Internet, the quicker a page will load since there is a significant expense associated with making a new request over a high-latency connection. Thankfully, there are a lot more technologies accessible today than there were in 1996, when the entire Internet was sluggish, for minimizing requests.

Reduce HTTP Requests:

Using external JavaScript and CSS files as a concatenation, high-performance websites minimize HTTP requests. It is best to limit the number of references to external JavaScript and CSS files per page load to two. When two or more files are present on the server, they ought to be downloaded to the browser as a single file wherever possible. Per page load, there should be a maximum of two references to external JavaScript files and two references to external CSS files. Using a CDN, files are concatenated in real-time. Google even created a `mod_concat` Apache plugin that makes it simple to concatenate files dynamically.

`Mod_concat` enables these files to be merged into a single request rather than being referenced individually by using the following URL:

- <http://www.example.com/assets/js??main.js,utils.js,lang.js>

This URL combines the responses from `main.js`, `utils.js`, and `lang.js` in the provided order. Take note of the two question marks, which tell the server to concatenate this URL's behavior. Better edge caching is achieved for the generated file by installing `mod_concat` on a server and utilizing that server as an origin behind a CDN.

Eliminate Images:

One of the most significant web components on the Internet is images. With CSS3, you can make buttons with rounded edges, a drop shadow, and a gradient backdrop, among other image-free design features. For CSS gradients, there is no need for a vendor prefix in the most recent versions of all major browsers. Certain commands such as `background`, `border-radius`, `box-shadows`, `text-shadows` can be used for a button instead of setting up an image for a button. Therefore, the various pictures that could have been required for this button can be replaced with just four lines of CSS code. Furthermore, compared to utilizing photos, this effect uses a lot fewer data to create. When feasible, it's a good idea to replace pictures using CSS. It decreases the total amount of bytes required for the visual design and lowers the frequency of HTTP requests.

Avoid Redirects:

Redirects must be avoided according to High-Performance Web Sites Rule 11. Redirects come with all of a complete request's latency and overhead. Many websites made the mistake of assuming that consumers would input the whole domain name for their site depending on the desired version, whereas in fact they often only enter the hostname, such as `bing.com`. Here since we search `bing.com`, the server

running that application automatically redirects it to the `www` version of the domain. The application further recognizes the user agent string and notes that it is a mobile device, and then redirects the application to the `m` version of the domain. Two redirects—one from `bing.com` to `www.bing.com` and the other from `www.bing.com` to `m.bing.com`—are shown in the Web Inspector window. After the initial request, the actual page doesn't start downloading until 1,448 milliseconds later. Redirects must be avoided while developing mobile websites. Users of the website benefit from improved performance because the entire experience is served from the same domain.

Mobile Device Limitations-

Web designers didn't have to give any thought to the type of device users were using to access their apps until lately. They must now think about the equipment they are utilizing.

Slow and Expensive Javascript

Although mobile device browsers are generally decent, their JavaScript engines are far slower than those of desktop PCs. Radios, network connectivity, and code execution all have the potential to deplete the batteries of mobile devices. JavaScript use accelerates battery consumption. JavaScript shouldn't be used by developers on mobile devices since it executes a lot of code often and could even skip the CPU. JavaScript should only be used as much as is absolutely necessary on mobile devices.

Less Memory

Compared to desktop and laptop computers, mobile devices have less memory, and it is simple to run out of memory unknowingly. The browser becomes sluggish or unusable and finally crashes when a memory issue gets out of hand. When creating photo-based Web apps for mobile devices, developers frequently encounter issues that result in browser crashes. By regularly eliminating items from the DOM as they became unnecessary, Flickr was able to overcome this issue. Hardware-accelerated graphics, which use the GPU to determine what needs to be rendered on the screen, are what led to Flickr's issue. This uses more memory, and the likelihood of a browser crash increases with the number of composited items on a page. Early on in the creation of mobile websites, developers attempted to impose hardware acceleration even when animations weren't required. They advised utilizing certain CSS attributes to compel hardware acceleration for components. In an effort to speed up the mobile Web experience, hasty developers began applying 3D transformations to the whole page. Instead, it made the experience sluggish and caused the browser to crash.

CONCLUSION

The interesting twist in what has generally been a reasonably easy process is web development for mobile devices. When creating Web sites only for the desktop, developers do not have to take into account the significant constraints of mobile devices. Over-the-air data transmission's

inherent latency results in slower download times and calls for caution in limiting the total number of queries made on any particular page. The same Web page that loads quickly and smoothly on a desktop computer could be rather sluggish on a mobile device due to the slower JavaScript engine and less RAM. Web developers now have to think about the type of device being used when developing a web application, and must pay close attention to how they craft interfaces. As web development for mobile devices progresses, it is increasingly important to take execution time, byte counts, request counts, memory utilization, and other factors into account.

REFERENCES

- https://www.w3.org/2006/12/digital_divide/ajc.pdf
- <https://www.spec-india.com/blog/progression-of-web-applications-what-next-all-that-you-need-to-know>
- https://www.academia.edu/en/9501005/Evolution_of_Mobile_Software_Development_from_Platform_Specific_to_Web_Based_Multiplatform_Paradigm
- Walsh, D. 2012. Force hardware acceleration in WebKit with translate3d; <http://davidwalsh.name/>

- translate3d.
- modconcat.

<http://code.google.com/p/modconcat>

/