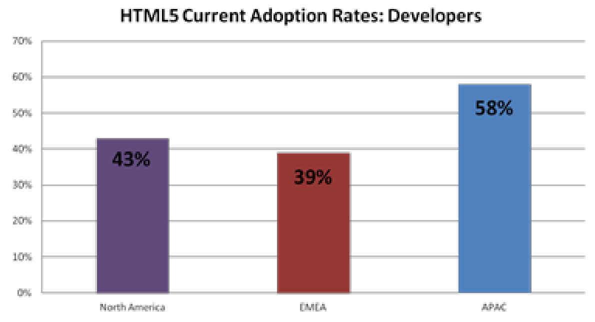
The Need for (HTML5) Speed

Lori MacVittie, 2012-01-08

#mobile #HTML5 #webperf #fasterapp #ado The importance of understanding acceleration techniques in the face of increasing mobile and HTML5 adoption



An old English proverb observes that "Even a broken clock is right twice a day." A more modern idiom involves a blind squirrel and an acorn, and I'm certain there are many other culturally specific nuggets of wisdom that succinctly describe what is essentially blind luck.

The proverb and modern idioms fit well the case of modern acceleration techniques as applied to content delivered to mobile devices. A given configuration of options and solutions may inadvertently be "right" twice a day purely by happenstance, but the rest of the time they may not be doing all that much good.

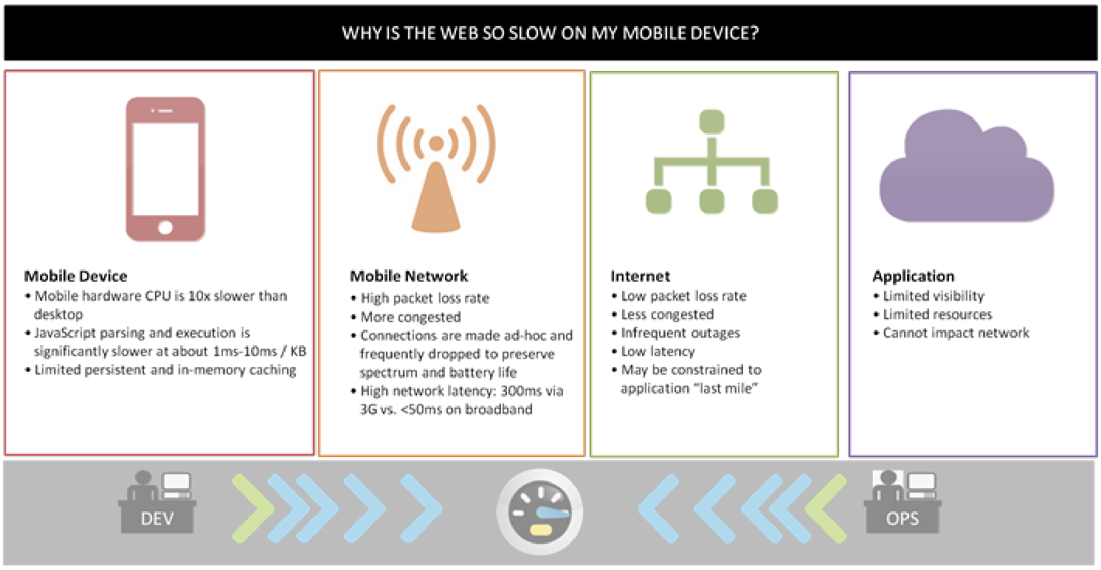
With HTML5 adoption increasing rapidly across the globe, the poor performance of parsing on mobile devices will require more targeted and intense use of acceleration and optimization solutions.

THE MOBILE LAST MILES

One of the reasons content deliver to mobile devices is so challenging is the number of networks and systems through which the content must flow. Unlike WiFi connected devices, which traverse controllable networks as well as the Internet, content delivered to mobile devices connected via carrier networks must also traverse the mobile (carrier) network. Add to that challenge the constrained processing power of mobile devices imposed by carriers and manufacturers alike, and delivering content to these devices in an acceptable timeframe becomes quite challenging. Organizations must contend not only with network conditions across three different networks but also capabilities and innate limitations of the devices themselves. Such limitations include processing capabilities, connection models, and differences in web application support.

Persistence and in-memory caching is far more limited on mobile devices, making reliance on traditional caching strategies as a key component of acceleration techniques less than optimal. Compression and de-duplication of data even over controlled WAN links when mobile devices are in WiFi mode may not be as helpful as they are for desktop and laptop counterparts given mobile hardware limitations.

Difference in connection models – on mobile devices connections are sporadic, shorter-lived, and ad-hoc – render traditional TCP-related enhancements ineffective. TCP slow-start mechanisms, for example, are particularly frustrating under the hood for mobile device connections because connections are constantly being dropped and restarted, forcing TCP to begin again very slowly. TCP, in a nutshell, was designed for fixed-networks, not mobile networks. A good read on this topic is Ben Strong's "Google and Microsoft Cheat on Slow-Start. Should You?" His testing (in 2010) showed both organizations push the limits for the IW (initial window) higher than the RFC allows, with Microsoft nearly ignoring the limitations all together. Proposals to increase the IW in the RFC to 10 have been submitted, but thus far there does not appear to be consensus on whether or not to allow this change. Also not discussed is the impact of changing the IW on fixed (desktop, laptop, LAN) connected devices. The assumption being that IW is specified as it is because it was optimal for fixed end-points and changing that would be detrimental to performance for those devices.



The impact of TCP on mobile performance (and vice-versa) should not be underestimated. CloudFare has a great blog post on the impact of mobility on TCP-related performance concluding that:

> TCP would actually work just fine on a phone except for one small detail: phones don't stay in one location. Because they move around (while using the Internet) the parameters of the network (such as the latency) between the phone and the web server are changing and TCP wasn't designed to detect the sort of change that's happening.
>
> -- CloudFare blog: Why mobile performance is difficult

One answer is more intelligent intermediate acceleration components, capable of detecting not only the type of end-point initiating the connection (mobile or fixed) but actually doing something about it, i.e. manipulating the IW and other TCP-related parameters on the fly. Dynamically and intelligently.

Of course innate parsing and execution performance on mobile devices contributes significantly to the perception of performance on the part of the end-user. While HTML5 may be heralded as a solution to cross-platform, cross-environment compatibility issues, it brings to the table performance challenges that will need to be overcome.

> http://thenextweb.com/dd/2012/05/22/html5-runs-up-to-thousands-of-times-slower-on-mobile-devices-report/
>
> In the latest research by Spaceport.io on the performance of HTML5 on desktop vs smartphones, it appears that there are performance issues for apps and in particular games for mobile devices.
>
> Spaceport.io used its own Perfmarks II benchmarking suite to test HTML rendering techniques across desktop and mobile browsers. Its latest report says:
>
> We found that when comparing top of the line, modern smartphones with modern laptop computers, mobile browsers were, on average, 889 times slower across the various rendering techniques tested. At best the iOS phone was roughly 6 times slower, and the best Android phone 10 times slower. At worst, these devices were thousands of times slower.

Combining the performance impact of parsing HTML5 on mobile devices with mobility-related TCP impacts paints a dim view of performance for mobile clients in the future. Especially as improving the parsing speed of HTML5 is (mostly) out of the hands of operators and developers alike. Very little can be done to impact the parsing speed aside from transformative acceleration techniques, many of which are often not used for fixed client end-points today. Which puts the onus back on operators to use the tools at their disposal (acceleration and optimization) to improve delivery as a means to offset and hopefully improve the overall performance of HTML5-based applications to mobile (and fixed) end-points.

DON'T RELY on BLIND LUCK

Organizations seeking to optimize delivery to mobile and traditional end-points need more dynamic and agile infrastructure solutions capable of recognizing the context in which requests are made and adjusting delivery policies – from TCP to optimization and acceleration – on-demand, as necessary to ensure the best delivery performance possible. Such infrastructure must be able to discern whether the improvements from minification and image optimization will be offset by TCP optimizations designed for fixed end-points interacting with mobile end-points – and do something about it. It's not enough to configure a delivery chain comprised of acceleration and optimization designed for delivery of content to traditional end-points because the very same services that enhance performance for fixed end-points may be degrading performance for mobile end-points. It may be that twice a day, like a broken clock, the network and end-point parameters align in such a way that the same services enhance performance for both fixed and mobile end-points. But relying on such a convergence of conditions as a performance management strategy is akin to relying on blind luck.

Addressing mobile performance requires a more thorough understanding of acceleration techniques – particularly from the perspective of what constraints they best address and under what conditions. Trying to leverage the browser cache, for example, is a great way to improve fixed end-point performance, but may backfire on mobile devices because of limited capabilities for caching. On the other hand, HTML5 introduces client-side cache APIs that may be useful, but are very different from previous HTML caching directives that supporting both will require planning and a flexible infrastructure for execution. In many ways this API will provide opportunities to better leverage client-side caching capabilities, but will require infrastructure support to ensure targeted caching policies can be implemented.

As HTML5 continues to become more widely deployed, it's important to understand the various acceleration and optimization techniques, what each is designed to overcome, and what networks and platforms they are best suited to serve in order to overcome inherent limitations of HTML5 and the challenge of mobile delivery.

- Google and Microsoft Cheat on Slow-Start. Should You?"
- HTML5 runs up to thousands of times slower on mobile devices: Report
- Application Security is a Stack
- Y U No Support SPDY Yet?
- The "All of the Above" Approach to Improving Application Performance
- What Does Mobile Mean, Anyway?
- The HTTP 2.0 War has Just Begun

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com