Adoption Trends for Chromium Browsers

Testing for performance differences in Android browsers
Mobile performance is essential to engage today’s consumers. It’s critical to optimize websites to meet user expectations for consistent, fast, secure digital experiences. Testing mobile websites in as many different browsers as possible is an important part of that optimization process. Yet there are many complications involved in testing for Android browsers. This white paper will examine some of those complications in detail to determine if there are notable performance differences between common Chromium-based Android browsers, and to outline how you can optimize your testing process.

On Android, Chrome may not be Chrome. That is, there are several browsers that use the Chromium open source browser but nonetheless are not Google Chrome. We discussed this state of affairs in the “When Chrome Isn’t Chrome” white paper. Since we’re going to continue our study of Chromium-based browsers in this white paper, it’s useful to repeat the main conclusions of the previous one:

- There are many Android browsers based on Chromium, Google’s open-source browser that anyone can use. Google Chrome is the best-known one, but far from the only one.
- About 70% of Google Chrome users update their browsers within three weeks of a new version being released. On the other hand, 30% do not, and frequently stay on an older version for a long time.

On Android, Chrome may not be Chrome. That is, there are several browsers that use the Chromium open source browser but nonetheless are not Google Chrome.
CHROMIUM BROWSERS

• About one in four Samsung device users do not use Google Chrome — and about 23% use Samsung Internet.

• Samsung Internet is also based on Chromium, but is not exactly the same as Google Chrome.

• Other Chromium-based browsers include the Xiaomi default browser, the UC browser, discontinued browsers on older HTC and LG phones, and Microsoft Edge for Android.

• These browsers resemble one another but are not exactly the same. In fact, this white paper studies one way in which they’re different.

• In addition to all these browsers, there are also WebViews. These are separate browsers used by native apps such as Twitter for displaying HTML content without leaving the app. WebViews are overwhelmingly Google Chrome-like, but they might be slightly behind the latest Google Chrome version.

• Browser detection scripts often cannot handle this browser diversity and count all these browsers as Chrome. This artificially inflates Google Chrome’s market share somewhat. We recommend the site WhichBrowser, which powers the HTML5 Test site.

In this white paper, we're going to concentrate on the high-level performance data of three Chromium-based browsers: Google Chrome, Samsung Internet, and the MIUI Browser created by Xiaomi for its devices. Is there a notable performance difference between the three browsers, or do they roughly perform the same? What consequences does this have for your testing setup?
We subdivided the 61, 67, and 73 buckets into MIUI, Samsung Internet, and Other.

THE BROWSER DATA

We used about 5.5 billion hits from Chromium 61, 67, and 73 Android browsers between February and May 2019 for testing. We selected 61 and 67 because those are the versions that MIUI Browser and Samsung Internet use (though shortly after the test period, Samsung Internet started its upgrade to 73). As for 73, it was the most recent Google Chrome version in the test period, and we used it for comparison.

We subdivided the 61, 67, and 73 buckets into MIUI, Samsung Internet, and Other: 15.6% of hits came from Samsung Internet 67 and 2.4% from MIUI Browser 61. The Other category, good for 82% of the hits, is overwhelmingly Google Chrome, though it likely contains trace elements of other Chromium-based browsers.

The Xiaomi 67 and 73 and the Samsung Internet 61 buckets were too tiny for robust measurements with fewer than 10,000 hits each, and we excluded them. We did use the Samsung Internet 73 bucket, even though it only contains 500,000 hits and may thus not be entirely reliable, either.
4 KEY PERFORMANCE METRICS

For all these browsers, we looked at four performance metrics:

1. Total loading time
2. The DOMContentLoaded event (DCL)
3. Time to interactive (TTI)
4. Longtasks

The total loading time is just that: How long did it take for the page to be completely loaded? This is an easy metric to gather but potentially also a misleading one. For instance, it could be that one script or image is very slow in loading and thus drives up the total loading time, or that such script or image is not required to start using the page. Thus, users will start interacting with the page long before loading has officially finished; to users, the page appears to be considerably faster – and the customer experience much better – than total loading time would suggest.

That’s why we need the three additional metrics.

The DCL event fires when the page has been loaded completely, but external assets such as images or scripts may still be coming in. Technically speaking, it fires when the HTML parser is done parsing the page. From this moment on, changes to the DOM structure can be made. Thus, it is an excellent measurement for determining whether a page is technically ready for interaction.

To users, the page appears to be considerably faster – and the customer experience much better – than total loading time would suggest.
However, it could be that there is a script not yet loaded that makes important changes to the page; so important, that the page is not usable without these changes. That’s where the TTI measurement comes in. It measures the moment at which the user can safely start interacting with the page; for instance, clicking on a button, which fires up a script that does something useful to the page.

Finally, we use longtasks in our measurements. Longtasks are exactly that: long tasks that the browser executes and that tie up its resources. More important, while these longtasks are running, browsers can do nothing else. If a user clicks on a link during a longtask, the browser finishes the longtask first and then reacts to the click. This might have a jarring effect on the customer experience. mPulse only measures longtasks that take place up to TTI, so longtasks that take place after page load are not part of this survey.

One more note before we look at the results: Because all Akamai clients do not have TTI and longtask measurements configured, TTI and longtask data are available for only about 40–45% of the pages in our sample. Still, they sketch roughly the same picture as total loading time and DCL.

**This might have a jarring effect on the customer experience. mPulse only measures longtasks that take place up to TTI, so longtasks that take place after page load are not part of this survey.**
BROWSER PERFORMANCE RESULTS

With this methodology, what did we find?

The main conclusion is easy: Of the six browsers, Samsung Internet 67 and Google Chrome 73 are the fastest across all four metrics. The others, Google Chrome 61 and 67, Samsung Internet 73, and MIUI 61, are noticeably slower.

The total load time data clearly shows this. About 20% of Google Chrome 73s and Samsung Internet 67s manage to load pages within two seconds, while only about 10% of the other four browsers manage to do so. Conversely, between 40% and 45% of the other four browsers take more than eight seconds to load a page, while only 20% of Google Chrome 73s and Samsung Internet 67s take that much time.
The TTI and DCL metrics show the same: Google Chrome 73 and Samsung Internet 67 perform significantly better than the other four browsers.
Only the longtasks metrics are slightly less clear. Although, again, Google Chrome 73 and Samsung Internet 67 perform better than the other four, the difference is smaller than in the other three measurements.
WHAT DO THE RESULTS MEAN?

We find that, across the board, Samsung Internet 67 is roughly as fast as Google Chrome 73. Consumers won’t notice much difference between the two browsers — and neither will web developers. More interestingly, Samsung Internet 67 is significantly faster than Google Chrome 67, which theoretically should be the most comparable browser.

Why is that? They should be the same browser, right? The most interesting answer to this question is that the remaining Google Chrome 67 users are not unwilling to upgrade to a newer version; rather, they are unable to do so because of the limitations of their device. Thus, the cause of the performance difference would be the device, not the browser.

This line of reasoning also explains why Samsung Internet 67 and Google Chrome 73 are roughly similar: both run on similar hardware since they are the latest versions of these browsers.

Another reason might be that Samsung Internet is optimized for Samsung devices. In this theory,

In this theory, Samsung Internet cooperates just a little bit better with the underlying hardware, software, and networking stacks. Even a slight improvement across the board in this area could lead to much better performance results.

Unfortunately, all this remains somewhat-educated guesswork; we do not actually know why some people don’t upgrade their browsers. We also do not know everything about Samsung Internet’s integration with Samsung software and hardware. Therefore, we will have to leave it at this.
As one would expect, the higher the Google Chrome version, the better the performance. This is likely due to a combination of running on newer devices and the performance tweaks Google builds into each new version. Samsung Internet bucks the trend here; the new 73 version, which was on the verge of being rolled out when this research took place, had distinctly worse performance than the 67. However, the Samsung Internet 73 sample is really small (about 0.01% of the total sample), so we should be careful not to draw too many conclusions from such a narrow base.

The only clear outlier is the relation between MIUI 61 and Google Chrome 61. On average, they have roughly the same performance, which is odd if we assume that Google Chrome 61 runs on outdated devices while MIUI does not, as we theorized for 67. Maybe it’s genuinely a worse browser from a performance perspective?

There is one other interesting point, and that is that the differences are much less pronounced when we look at longtasks. We think that this is due to device fragmentation – some devices have worse processors than others, and it seems that the browser used has fairly little to do with longtask time.

Still, it’s clear that performance differences can be large during page load. A good performance strategy addresses those loading problems first. An occasional glitch while a customer is using your website can be mildly annoying, but otherwise-happy customers may forgive it, especially if it’s not recurring.

Conversely, when a new visitor comes to your site, the first impression they get is the load time. If it takes too long, they leave, never to return. Thus the cost of long load times is higher than the cost of occasionally slow interactions coupled with decent load times.
TESTING RECOMMENDATIONS

It’s obvious that testing your websites on Samsung Internet in addition to Google Chrome is a good idea. While similar, these browsers are not the same, and although Samsung Internet has the edge in performance, it could be that Google Chrome is better at other tasks. The only way of finding this out is being aware of differences in browsers and performing rigorous tests of your websites.

In order to do so, you should establish, or expand, your device lab. We’d like to close this paper with some recommendations in this area:

1. Add a state-of-the-art Samsung Galaxy, with both Samsung Internet and Google Chrome.

2. Add a much cheaper Samsung. Buy a popular model for about $200, and you will test on a device that people actually have in their pockets.

3. Add a non-Samsung device such as a Huawei, Xiaomi, Sony, or Motorola. As we saw in this article, Xiaomi phones have their own default browser, so you should pick that brand if you go for browser diversity. Note: Any Google-branded phone does not count, because they’re not actually being used all that much by consumers. You should test on what’s in consumers’ pockets; not what’s in your developers’ pockets.

4. If you can afford another non-Samsung phone, buy one. Maybe make it a cheaper model in addition to the flagship model you already have.

5. Then, install browsers on these phones. Google Chrome and the vendor default browser will already be present, but you should install Firefox, Opera Mobile, Opera Mini, and UC, at the minimum. Make sure each browser runs on a high-end and a mid-range device; you might find interesting differences.

6. Make a list. That is, if a website is being formally tested, developers should go through all devices and all browsers and not forget to test in one or two. A list helps here; it will tell even new developers and interns what your testing process looks like. They’ll appreciate the clarity (though not the extra browser bugs they’ll find).

Data for this research was collected using mPulse. Start monitoring real user performance today.

Get Started: www.akamai.com/trympulse