

LARGE-SCALE PAGE SPEED ANALYSIS:

Methods & Results

BACKLINKO

LARGE-SCALE PAGE SPEED ANALYSIS: METHODS & RESULTS

The goal of this study was twofold.

First, we aimed to establish site speed benchmarks for key metrics, including Time to First Byte, First Contextual Paint, onLoad and Fully Loaded.

Next, we analyzed various factors that impact page loading speed.

To our knowledge this is one of the largest page-speed studies of this kind (in regard to sample size). However, analyzing pages for site speed is far from novel.

In fact, many of the factors that we analyzed were specifically designed to dig deeper into past research findings. [This analysis by Google](#) being one of the most cited.

Considering the impact that loading times can have on [conversions](#) and [SEO](#), we hope that our findings will be valuable to anyone that wants to improve the loading times of their website.

WHAT WE DID— STUDY METHODOLOGY

PART 1

The first part of this study aims to answer the following two research questions:

1. How accurately can speed metrics such as Time to First Byte (TTFB) or Start Render be predicted based on page characteristics?
2. What are the most important page characteristics that influence page speed?

To answer the first question, our approach is based on statistical modeling and machine learning. The method of choice is called “Gradient Boosted Decision Trees”. Gradient boosting is a popular machine learning technique which is highly scalable, efficient, and doesn’t require a lot of data transformations. It’s also considered to have a high prediction performance.

The process of training the model follows the commonly used steps: 1) split the data set to train and test sets, 2) fit the model using training data, 3) make predictions for the unseen test set, and finally 4) evaluate the results.

In this study, the prediction accuracy of the model is measured by calculating the correlation between the observed and predicted values of dependent variables (i.e. different speed metrics). Correlation of 0 would indicate that the model with given independent variables (i.e. a page characteristic like Image Weight) is not able to predict a dependent variable (e.g. TTFB) better than randomly guessing, i.e. independent variables don’t contain any relevant information or at least the model is failing to take that information into account. Whereas a maximum value for a perfect fit is 1; independent variables can perfectly explain the observations.

WHAT WE DID – STUDY METHODOLOGY

PART 1 CONTINUED

For the second question at hand, the choice was to use a recently developed technique called Leave-One-Feature-Out-Importance (LOFO). The idea behind LOFO is to iteratively remove one independent variable at a time from the data set and measure how much predictive power is lost compared to the full model. If the prediction accuracy is not affected at all, then the feature can be considered to be irrelevant for the task. On the other hand, removing important features should cause a large loss of accuracy. The results provide insights into where site owners may need to look for page speed optimizations.

The HTTP Archive database acted as datasource. HTTP Archive tracks how the web is built by crawling some 5 Million Webpages with Web Page Test, including Lighthouse results, and stores the information in BigQuery where it is publicly available.

More information can be found on <https://httparchive.org/> and a getting started guide can be accessed here: https://github.com/HTTPArchive/httparchive.org/blob/master/docs/gettingstarted_bigquery.md

We randomly sampled 100,000 rows from the May to July 2019 crawls. Each row contains details about a single page including timings, # of requests, types of requests and sizes. In addition, data points on the # of domains, redirects, errors, https requests, CDN, etc. are available. In total, we looked at 300,000 rows for both Mobile and Desktop. Please note that adding more data points to the model would not change the overall results. We did run the models with 3x more data, but the results were similar to the reported ones.

WHAT WE DID – STUDY METHODOLOGY

PART 2

We looked at various speed metrics in more detail that contribute to better user experience. We looked at First Contentful Paint (FCP), First Input Delay (FID), various Image Performance Metrics, and Time to First Byte (TTFB) as our core metrics.

We gathered data from browser sessions for the entire month of May 2019. Here we decided to join the May 2019 Http Archive data with the June 2019 Chrome-UX data. Chrome-UX data reflects how Chrome users experienced TTFB's, FCP's and FID's in real-world conditions. For example, TTFB data in the Http Archive has traditionally been measured synthetically in the lab accessed from a single server location, so what makes this dataset unique is that it reflects the real-world server response times experienced by Chrome users as they navigate the web. More information on the Chrome-UX data can be found here: <https://developers.google.com/web/tools/chrome-user-experience-report/>

Information on defining the SQL queries to extract the desired data were partly drawn from discussions on the Http Archive. Most of the time we expanded on those to provide novel insights. See: <https://discuss.httparchive.org>. The SQL queries can be provided on request. One needs to copy-paste those into Google's BigQuery (see getting started guide above).

We aggregated each metric and sliced it according to various categories. We looked at metrics across devices (Desktop vs Mobile), JavaScript Frameworks, Compression, Use of Third-party Scripts, CMS and Hosting Platforms, CDN Usage and CDN Providers.

We also categorized metrics based on existing benchmarks or provisioned standards. A metric can be categorized mainly as Fast, Average, or Slow. We see that exact numbers make sense, but users will most likely view a browsing experience as a good, average, or poor.

FACTOR-BY-FACTOR BREAKDOWN



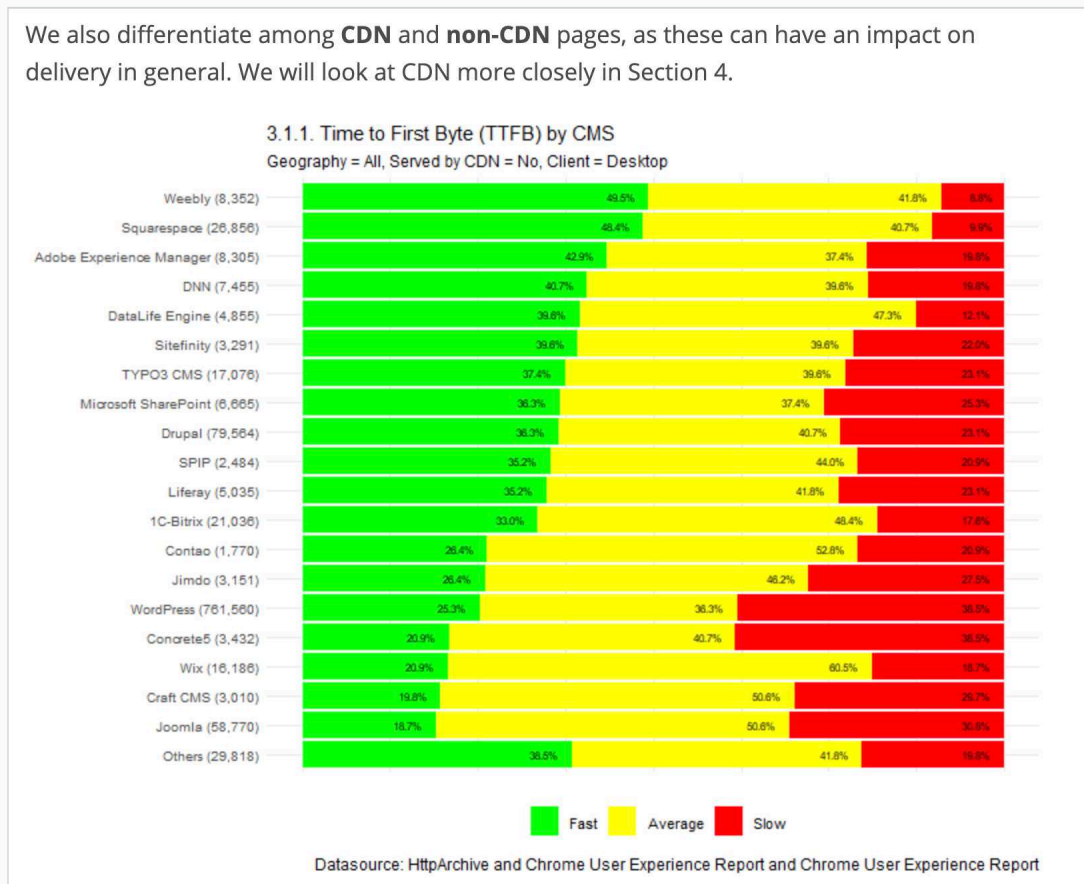
TIME-TO-FIRST-BYTE (TTFB) is measured as the time from the start of navigation request until the time that the client receives the first byte of the response from the server. It includes network setup time (SSL, DNS, TCP) as well as server-side processing. This metric is useful as it ignores the variability of front end performance and focuses only on network setup and backend response time. Available in both Http Archive (lab) and Chrome-UX (field) datasets.

TTFB is measured both with and without a CDN across all regions.



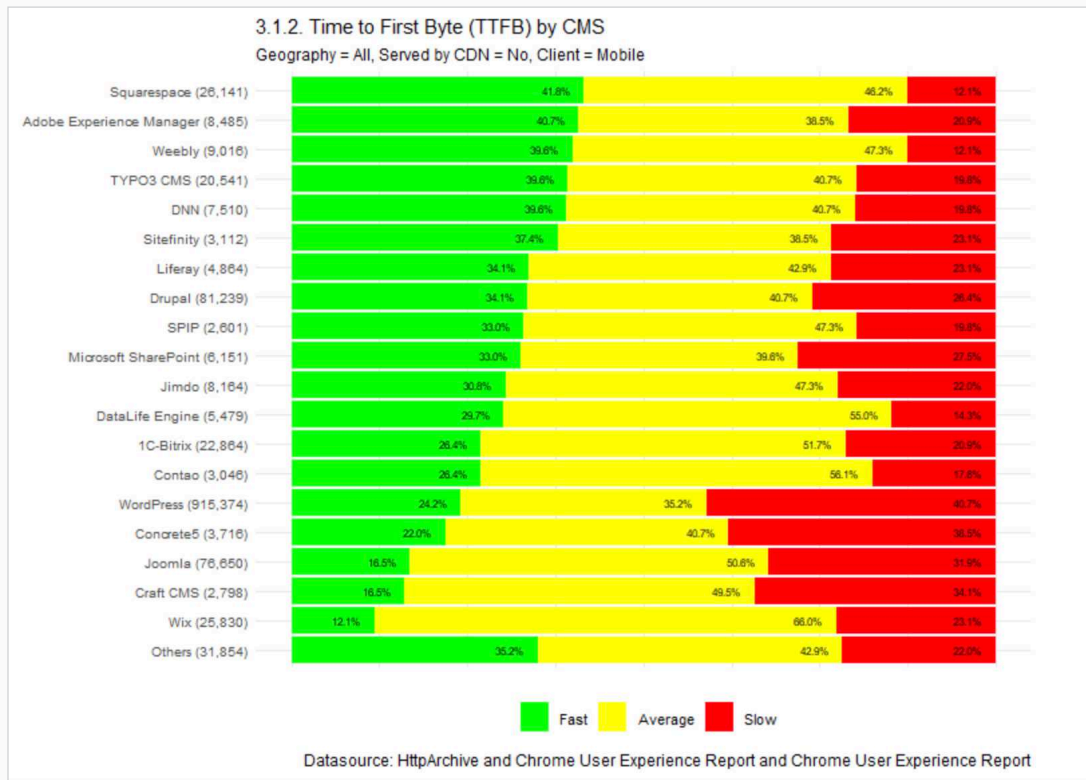
TTFB WITH CDN (DESKTOP) DATA:

We also differentiate among **CDN** and **non-CDN** pages, as these can have an impact on delivery in general. We will look at CDN more closely in Section 4.



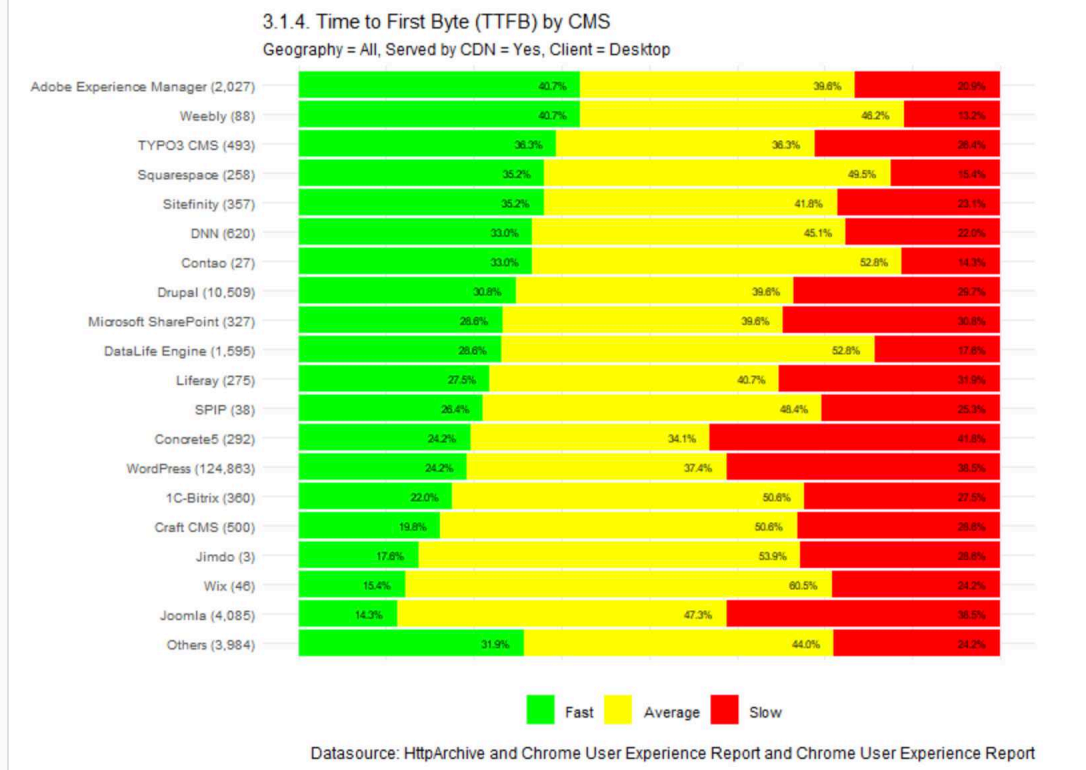


TTFB WITH CDN (MOBILE) DATA:



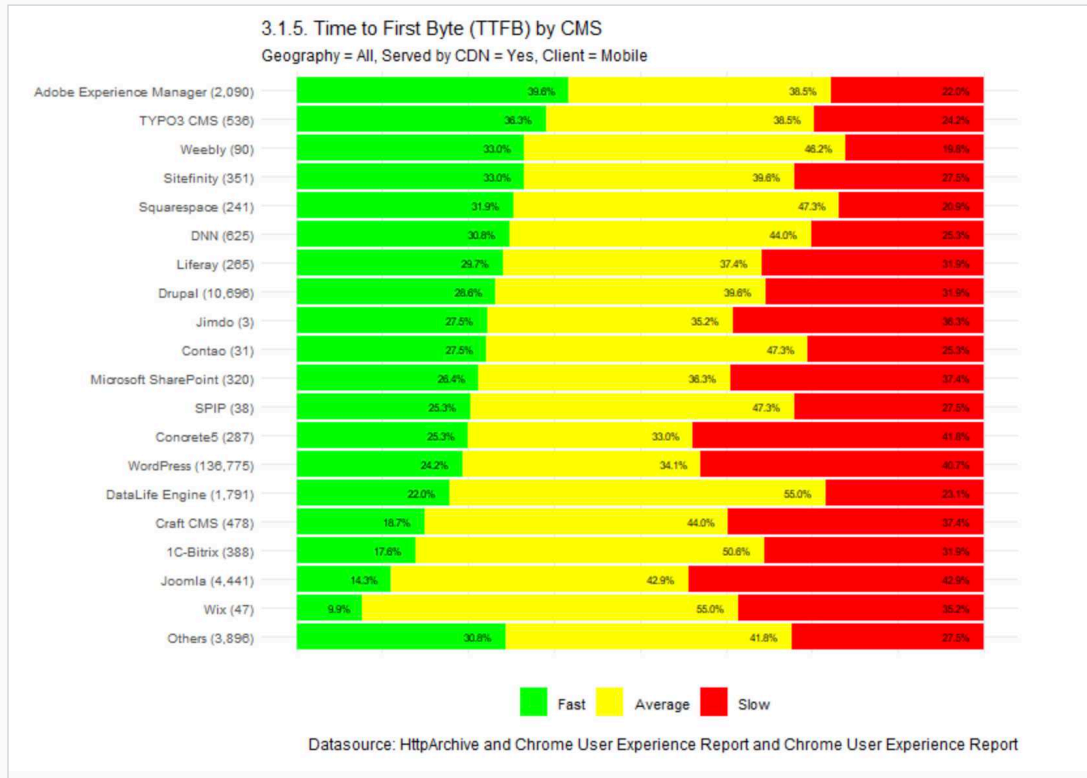
TTFB WITHOUT CDN (DESKTOP) DATA:

- **Mobile:** Fast (6 out of 21), Average (7 out of 21), Slow (16 out of 21)





TTFB WITHOUT CDN (MOBILE) DATA:



STARTRENDER / FIRST PAINT (FP) / FIRST CONTENTFUL PAINT (FCP) mark the

points, immediately after navigation, when the browser renders pixels to the screen. FCP, FP and StartRender are all slightly different but in practice, for the vast majority of sites, they end up being the same (within measurement error). FCP and first paint are when chrome thinks it painted content. StartRender is observed from the outside and is when the viewport actually changed. StartRender is available in Http Archive (lab) while FCP in Chrome-UX (field), while the FCP metric is available in both datasets.



VISUALLY COMPLETE is a user experience metric that identifies the moment in time when users perceive that all the visual elements of a page have completely loaded. Only available in the Http Archive dataset.



SPEED INDEX is a performance metric that measures how quickly a page renders visual elements, from the user's perspective. This is a rate-of-speed metric that is closely related to Visually Complete, which is a moment-in-time measure. Only available in the Http Archive dataset.



ONLOAD calculates the speed when the processing of the page is complete and all the resources on that particular page, such as images, CSS and other functionality have finished downloading. Onload Time is usually integrated with online speed testing tools such as Pingdom. Only available in the Http Archive dataset.



FULLY LOADED adopts the exact same process to page speed as onLoad but will add an additional two seconds after the Onload trigger has fired to make sure there is no further network activity. The reasoning behind this is to ensure more consistency with tests. Only available in the Http Archive dataset.



FIRST INPUT DELAY (FID) measures the time from when a user first interacts with your site (i.e. when they click a link, tap on a button, or use a custom JavaScript-powered control) to the time when the browser is actually able to respond to that interaction. Only available in the Chrome-UX dataset.

RESULTS

Besides the results that were published at Backlinko, all results (including a country-by-country appendix) can be found at: https://frontpagedata.com/projects/backlinko/pagespeed/Analysis_final.html