Stop worrying about Time To First Byte (TTFB)

07/05/2012

John Graham-Cumming

Time To First Byte is often used as a measure of how quickly a web server responds to a request and common web testing services report it. The faster it is the better the web server (in theory). But the theory isn't very good.

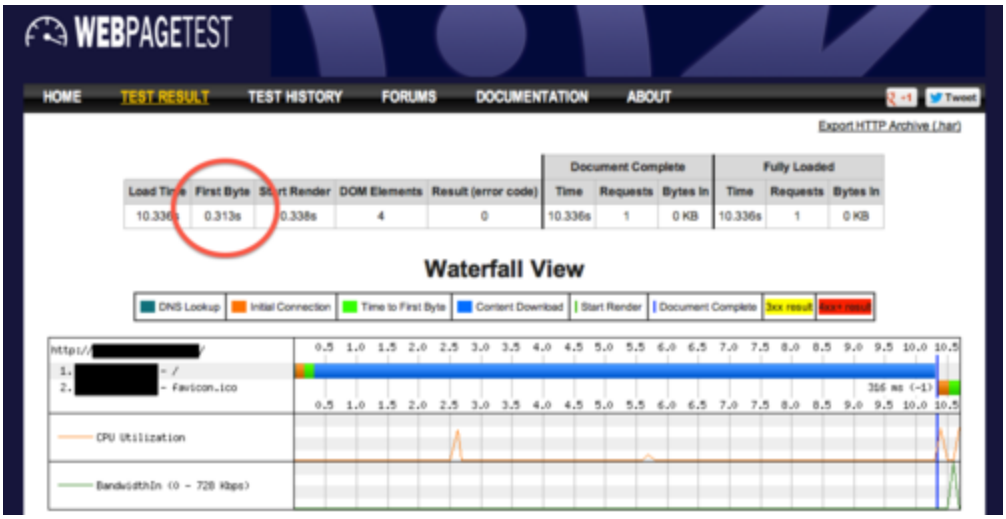
Wikipedia defines Time To First Byte as "the duration from the virtual user making an HTTP request to the first byte of the page being received by the browser." But what do popular web page testing sites actually report? To find out we created a test server that inserts delays into the HTTP response to find out what's really being measured. The answer was a big surprise and showed that TTFB isn't a helpful measure.

When a web browser requests a page from a web server it sends the request itself and some headers that specify things like the acceptable formats for the response. The server responds with a status line (which is typically HTTP/1.1 200 OK indicating that the page was available) followed by more headers (containing information about the page) and finally the content of the page.

CloudFlare's TTFB test server behaves a little differently. When it receives a request it sends the first letter of HTTP/1.1 200 OK (the H) and then waits

for 10 seconds before sending the rest of the headers and page itself. (You can grab the code for the TTFB server here; it's written in Go).

If you ask WebPageTest to download a page from the CloudFlare TTFB server you get the following surprise. WebPageTest reported the Time To First Byte as the time the H was received (and not the time the page itself was actually sent). The 10 second wait makes this obvious.



Exactly the same number is reported by gomez.

The TTFB being reported is not the time of the first data byte of the page, but the first byte of the HTTP response. These are very different things because the response headers can be generated very quickly, but it's the data that will affect the most important metric of all: how fast the user gets to see the page.

At CloudFlare we make extensive use of nginx and while investigating TTFB came across a significant difference in TTFB from nginx when compression is or is not used. Gzip compression of web pages greatly reduces the time it takes a web page to download, but the compression itself has a cost. That cost causes TTFB to be greater even though the complete download is quicker.

To illustrate that we took the largest Wikipedia page (List of Advanced Dungeons and Dragons 2nd Edition Monsters) and served it using nginx with and without gzip compression enabled. The table below shows the TTFB and total download time with compression on and off.

	TTFB	Page loaded
No compression (gzip off)	213us	43ms
Compressed (gzip on)	1.7ms	8ms

Notice how with gzip compression on, the page was downloaded 5x faster, but the TTFB was 8x greater. That's because nginx waits until compression has started before sending the HTTP headers; when compression is turned off it sends the headers straight away. So if you look at TTFB it looks as if compression is a bad idea. But it you look at the download time you see the opposite.

From the end user perspective TTFB is almost useless. In this (real) example it's actually negatively correlated with the download time: the worse the TTFB the better the download time. Peering into the nginx source code we realized we could cheat and send the headers quickly so that it looked like our TTFB was fantastic even with compression, but ultimately we decided not to: that too would have negatively impacted the end user experience because we would have wasted a valuable packet right when TCP is going through slow start. It would have made CloudFlare look good in some tests, but actually hurt the end user.

Probably the only time TTFB is useful is as a trend. And it's best measured at the server itself so that network latency is eliminated. By examining a trend it's possible to spot whether there's a problem on the web server (such as it becoming overloaded).

Measuring TTFB remotely means you're also measuring the network latency at the same time which obscures the thing TTFB is actually measuring: how fast the web server is able to respond to a request.

At CloudFlare TTFB is not a significant metric. We're interested in optimizing the experience for end users and that means the real end-user page being visible time. We'll be rolling out tools specifically to monitor end-user experience so that all our publishers get to see and measure what their visitors are experiencing.

Speed & Reliability

Follow on Twitter

John Graham-Cumming | @jgrahamc
Cloudflare | Cloudflare