

Performance testing for modern apps

The performance of your application affects your business more than you might think. Top engineering organizations think of performance not as a nice-to-have, but as a crucial feature of their product. Unfortunately, most engineering teams do not regularly test the performance and scalability of their infrastructure. To understand how to test performance in modern applications, you must start by understanding performance is key to a great user experience. The reality is the only performance metric that matters is the user's perceived load time.

The value of performance

A common question is how fast is fast enough for a web application? A quick overview of key performance metrics.

- Under 100ms is perceived as reacting instantaneously
- A 100ms to 300ms delay is perceptible
- 1 second is about the limit for the user's flow of thought to stay uninterrupted
 - Users expect a site to load in 2 seconds
 - After 3 seconds, 40% will abandon your site.
- 10 seconds is about the limit for keeping the user's attention

Most engineering teams under you need to treat performance as a feature. When it comes to performance testing you should understand the baseline performance of your application. The performance of each transaction is unique. For example, in an e-commerce application, a homepage transaction is likely highly cached and very fast whereas a checkout transaction is more complicated and has to talk to a payment service, shipping service, etc. To ensure users have a great experience you need to test the most common flows of your users and understand performance in the browser and on the server.

Understanding server-side performance

Apache Bench and Siege are great for quick load tests of a single end point. If you just need to get a sense of the requests per second for an endpoint, these are a great solution. A more advanced approach and my personal preference is locust.io which is a load testing framework that enables complex transactions and can generate high levels of concurrency with ease.


The screenshot shows the Locust.io website with a green header and a dark green background. The main heading is "An open source load testing tool." Below it, a subheading says "Define user behaviour with Python code, and swarm your system with millions of simultaneous users." There are social media links for Twitter, GitHub, and Facebook. A status bar indicates "STATUS RUNNING 9600 users" and "SLAVES 6". A table titled "Statistics" shows various metrics for different endpoints.

Type	Name	# requests	# fails	Median	Average	Min	Max	Content Size	# reqs/sec
GET	/	1831	0	21	21	4	38	19947	18.3
GET	/blog	608	0	25	26	3	49	19841	6.9
GET	/blog/[post-slug]	612	0	14	15	2	27	19858	7.8
GET	/forum	573	0	26	26	3	49	20209	5.5
GET	/forum/[thread-slug]	596	0	30	30	6	55	20209	5.3
POST	/forum/[thread-slug]	71	0	62	63	13	120	11188	0.6
POST	/forum/new	64	0	59	58	6	108	3272	0.7
GET	/login	3439	0	26	26	3	49	19850	31.3

- Locust.io is a great tool for understanding the performance of the server side.
- **Bees with Machine Guns:** A utility for arming (creating) many bees (micro EC2 instances) to attack (load test) targets (web applications).
- **MultiMechanize:** Multi-Mechanize is an open source framework for performance and load testing. It runs concurrent Python scripts to generate load (synthetic transactions) against a remote site or service. Multi-Mechanize is most commonly used for web performance and scalability testing, but can be used to generate workload against any remote API accessible from Python.
- **Siege:** Siege is an http load testing and benchmarking utility. It was designed to let web developers measure their code under duress, to see how it will stand up to load on the internet. Siege supports basic authentication, cookies, HTTP and HTTPS protocols. It lets its user hit a web server with a configurable number of simulated web browsers. Those browsers place the server “under siege.”
- **Apache Bench:** AB is a tool for benchmarking your Apache HTTP server. It is designed to give you an impression of how Apache performs.
- **HttpPerf:** Httpperf is a tool for measuring web server performance. It provides a flexible facility for generating various HTTP workloads and for measuring server performance. The focus of httpperf is not on implementing one particular benchmark but on providing a robust, high-performance tool that facilitates the construction of both micro- and macro-level benchmarks. The three distinguishing characteristics of httpperf are its robustness, which includes the ability to generate and sustain server overload, support for the HTTP/1.1 and SSL protocols, and its extensibility to new workload generators and performance measurements.
- **JMeter:** Apache JMeter may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, databases and queries, FTP servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. You can use it to make a graphical analysis of performance or to test your server/script/object behavior under heavy concurrent load.


Understanding client-side performance

Modern applications spend more time in the browser than on the server-side. The best tool to get started understanding client-side performance is Google PageSpeed Insights. Google PageSpeed Insights is a service to analyzes the content of a web page, and then generates suggestions to make that page faster.



Products > PageSpeed Insights

PageSpeed Insights

 Make your web pages fast on all devices.

ANALYZE

Web Performance

Learn more about [web performance tools at Google](#), including browser extensions and APIs for Insights, PageSpeed Service, and our optimization libraries.

Give Feedback

Have comments or questions about PageSpeed Insights? [Send feedback](#) or [discuss on our mailing list](#).

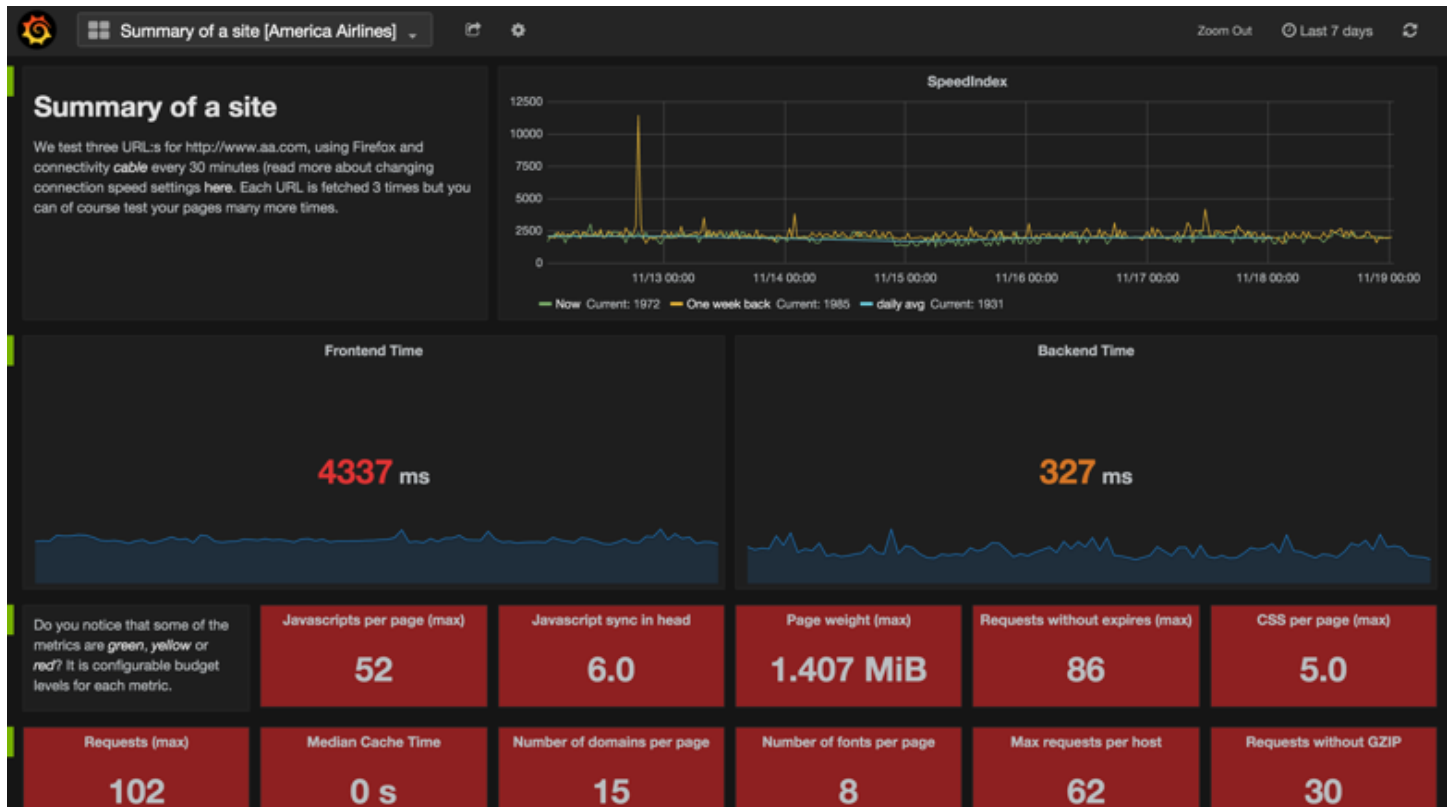
About PageSpeed Insights

PageSpeed Insights analyzes the content of a web page, then generates suggestions to make that page faster. [Learn more](#).

- **Google PageSpeed Insight:** PageSpeed Insights analyzes the content of a web page, then generates suggestions to make that page faster. Reducing page load times can reduce bounce rates and increase conversion rates.

Understanding real-world performance

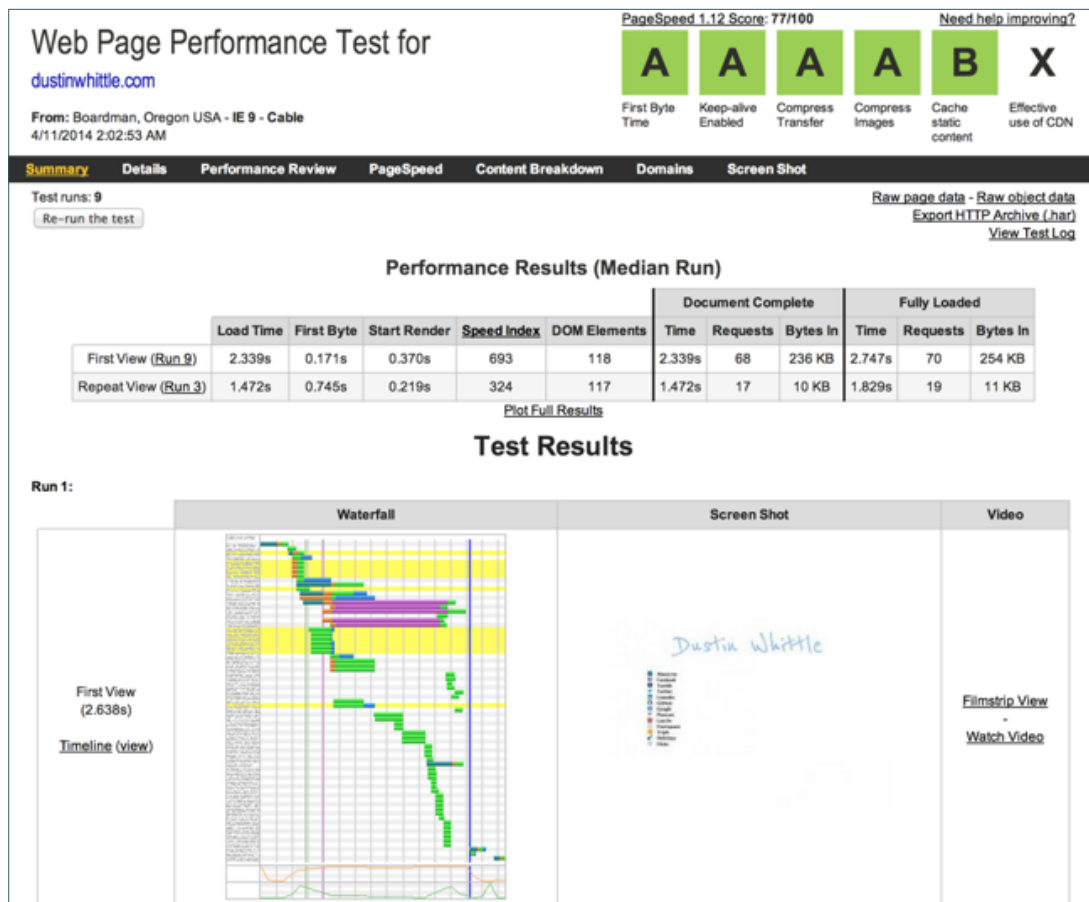
SiteSpeed.io is my favorite tool for evaluating the client-side performance from real browsers. Sitespeed.io is an open source tool that helps you analyze your website speed and performance based on performance best practices and timing metrics. You can analyze one site, analyze and compare multiple sites or let your continuous integration server break your build when your performance budget is exceeded.



It is not always possible for teams to modify the applications to optimize client-side performance. Google has invested in making ngx_pagespeed + mod_pagespeed as web server extensions to automate performance improvements without code changes.

- **Google ngx_pagespeed:** ngx_pagespeed speeds up your site and reduces page load time. This open-source nginx server module automatically applies web performance best practices to pages, and associated assets (CSS, JavaScript, images) without requiring that you modify your existing content or workflow.
- **Google mod_pagespeed:** mod_pagespeed speeds up your site and reduces page load time. This open-source Apache HTTP server module automatically applies web performance best practices to pages, and associated assets (CSS, JavaScript, images) without requiring that you modify your existing content or workflow.
- Cloudflare, Incapsula, Torbit, Visual Website Optimizer are all commercial services that will proxy your website and automatically improve performance without code or infrastructure changes.

WebPageTest.org is an excellent utility for testing a web page in any browser, from any location, over any network condition for free. WebPageTest gives deep insight into the performance of the client-side in a variety of real browsers.

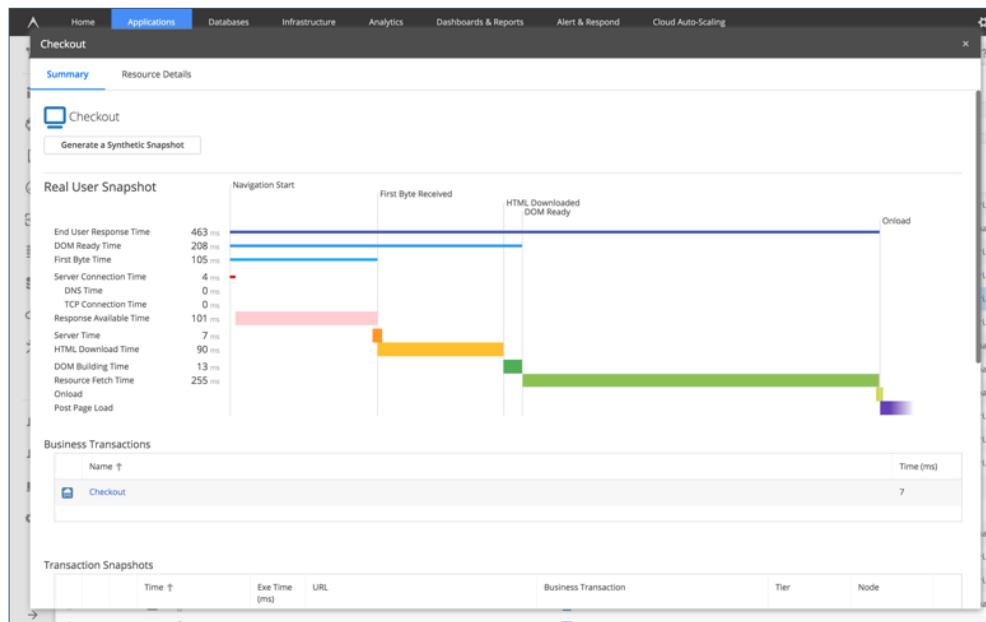


It is not always wise to build and manage your own performance testing tools and infrastructure. Through these services you can build, execute, and analyze performance tests.

- **Soasta**: Build, execute, and analyze performance tests on a single, powerful, intuitive platform.
- **Apica**: Cloud-based load testing for web and mobile applications
- **Blitz.io**: Blitz allows you to continuously monitor your app 24x7 from around the world. You can emulate a single user or hundreds of users all day, every day and be notified immediately if anything goes wrong.
- **Blazemeter**: BlazeMeter is a self- service performance & load testing cloud, 100% JMeter-compatible. Easily run tests of 30k, 50k, 80k or more concurrent users, on demand.

Deriving insights from performance and load testing

The goal of performance testing is to understand how your applications behave under heavy load conditions. Performance testing is only as useful as the intelligence it yields about your application's bottlenecks. When running performance tests, you should always instrument your applications and infrastructure to understand what breaks and why. APM tools enable you to see the performance of your applications and infrastructure in real-time. To completely understand performance modern teams leverage real user monitoring to get visibility into the real performance of your end users across many browsers and platforms.



Everyone knows you should treat performance as a feature. Hopefully, you now have a path to get started on capacity planning and load testing the server-side, optimizing and performance testing the client-side, and how to monitor performance from end to end to derive meaningful insights from performance tests.

Try it FREE at appdynamics.com