



ABOUT

Optimizely is the world's leading experimentation platform, empowering marketing, product, and engineering teams to test, learn and deploy winning digital experiences, every time. The platform supports experimentation at every turn, from Full Stack testing, feature management, and Rollouts, a free feature flagging solution, to Optimizely Web, for creating client-side tests with ease.

ENVIRONMENT

JavaScript

Cloudflare

AWS

GOALS

Optimizely's product team is always looking to provide a faster, more lightweight experience for their customers. From the Optimizely blog:

"When you create experiments in Optimizely Web, the experiment instructions are stored in a JavaScript file called a snippet. When a user visits your site, their browser downloads the snippet from Optimizely's CDN and evaluates it to determine what experience to deliver. As the number of experiments grow, the snippet also grows in size because it includes information about all experiments. The larger snippet results in more time to render the experiment which can impact site performance."

This snippet sends telemetry data back to Optimizely for analysis, such as page load and JavaScript execution timings, useragent details, and lots more—most of it of moderate or high cardinality—meaning a great many unique values for a given field. The previous implementation of the Optimizely service also sent everything to a PostgreSQL database, which resulted in lots of trial and error when looking for data to troubleshoot these performance concerns.

"Before we had Honeycomb, you used to have to know the schemas so you could optimize queries; you had to know what you needed to ask ahead of time" –Michael Hood, Senior Staff Performance Engineer, Optimizely

The compound issues of the snippet increasing in size and difficulty in retrieving data when debugging performance issues for the service meant Optimizely's engineering team needed to turn to a new approach to maintain the high quality of service their customers expect.

WHAT THEY NEEDED:

- The ability to locate the exact pain points impacting site performance
- Lightning-fast high-cardinality exploration without having to know what questions to ask ahead of time

HONEYCOMB @ OPTIMIZEZY

With Honeycomb in the mix, the team at Optimizely was able to make rapid progress, to take what Senior Staff Performance Engineer Michael Hood refers to as an "evolutionary step" and build a new, better, more scalable offering with the observability it brings to their development process.

"With Honeycomb, we were able to do something much more radical, high-risk/high reward – with much less risk. Sending the data from the MVP into Honeycomb made it so much easier to know where to focus."

With their Honeycomb observability superpowers, Optimizely built and shipped their new Performance Edge solution for Optimizely Web. This benefits those who use the web editor to create and run experiments at scale, and is up to 20 times faster than their previous implementation – all without breaking the bank.

"It would cost us so much more to run the compute and storage for us to have the explorability Honeycomb gives us. We would need a team of architects and DBAs to do this for ourselves."

And their Customer Support team is feeling the magic as well: customers sometimes reach out for help interpreting the data they get back from their Optimizely experiments and the support engineers often have a clearer picture of their website's performance:

"(The customers) don't have the ability to dive in and slice and dice; it's outside the reach of what they can do with their own data."

The customers have anecdotal evidence from their users, but Honeycomb means Optimizely can provide the real data and confirm what's actually happening—a level of service that customers often take to mean that Optimizely's Performance Engineering team is much larger than it is:

"Often people say 'You must have a big org to do this stuff!' and we tell them 'no, this visibility takes just a few engineers.'"

Go faster, save money, and impress your customers all at the same time, with Honeycomb.