Biba's mission is to get today's screen-savvy kids back outdoors and active. To that end, they have created a suite of mobile games that are specifically designed to work with the equipment found in neighborhood playgrounds.

More than 2000 Biba playgrounds are already active in cities across North America and beyond.

Environment:

- Google Cloud, App Engine, Container Engine, Cloud Functions, Pub/Sub
- Postgres
- Go
- Node.js

Need:

- A way to rapidly instrument their new code for timings and traces so they could see and understand the benefit of their new infrastructure and what still needed tweaking

- An observability service that allowed them to investigate performance issues and drill down rapidly to identify sources of slowness

Love:

"Almost immediately after we had Honeycomb set up and getting traces, we solved our first problem.

We are ecstatic about it; we're looking forward to showing off how Honeycomb has raised the level of performance."

*- **Sean Hagen**,*
Tech Lead, Biba

Challenge:

Biba's engineering team faces a complex set of challenges in developing and delivering an augmented-reality (AR) app for kids. Speed is of the utmost importance; kids don't have a lot of patience for a slow experience (neither do their parents) and mobile apps already suffer the latency associated with the delivery network, so optimizing back-end performance is critical.

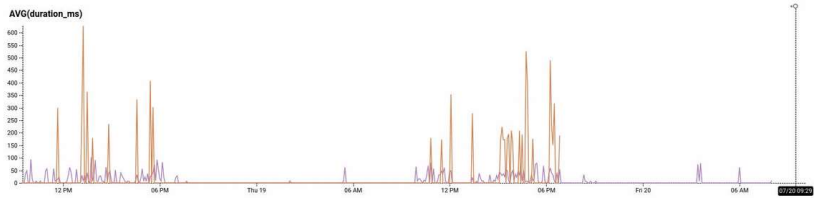
The Biba engineering team recently undertook a complete rewrite of their back-end services to take advantage of modern technologies, but the need for speed and clear visibility into user behaviors and experience remained at the top of their list of requirements. New services mean new instrumentation, and that can take significant time investment. Additionally, the lack of direct access to logs that comes along with hosted infrastructure was an additional barrier to understanding what might be impacting their service.

Solution:

The engineers at Biba got started fast, with the help of the Honeycomb Beelines for Go and Node.js, getting their code **instrumented automatically** to send wide, context-rich events and traces into Honeycomb.

One thing they were able to debug immediately was the service that provides information to users about nearby playgrounds that support the Biba service. The route that provides the information had an average response time over a second, compared to the average of 100 ms for the other routes.

"Being able to highlight the routes that the games are dealing with has been very powerful and helpful. With Honeycomb, we could look at exactly what part of the route was taking the most time."



Graph showing the performance difference between the non-optimized and optimized database query

Results:

They were able to identify that the database query was not optimized, and were then able to improve response time from one second down to 200ms, as well as improve other routes along the way.