Aspect BUILD | Sourcegraph

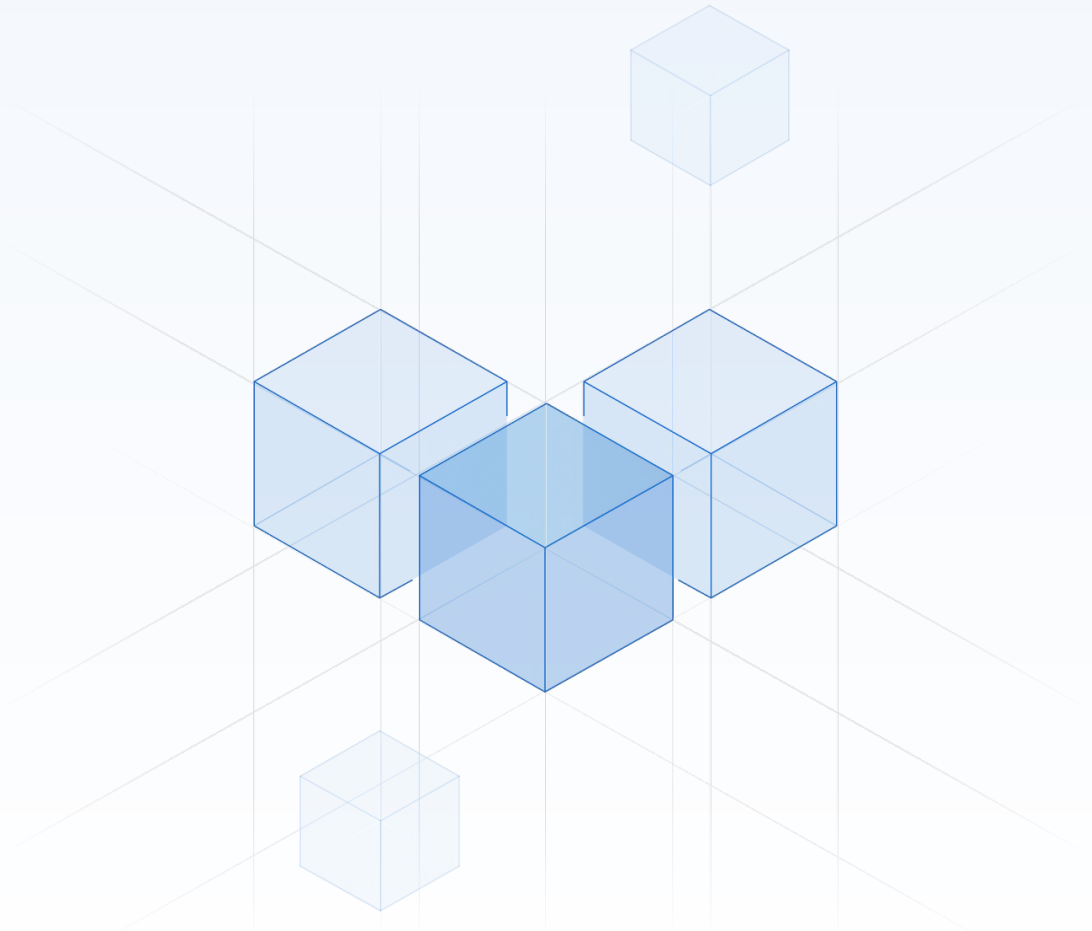
Sourcegraph is a code intelligence platform known for their AI coding assistant, Cody.

Their software is developed in an Open-Source repository by a large team of engineers, primarily in Go, TypeScript, and Rust.

They were building ~50 different docker container images, which were not optimized. Optimizing them individually was considered less attractive than using a build system that naturally produces correct images.

The build for the frontend client bundle was non-incremental and rebuilt on every change. This was a big pain point and made CI very slow. It was also flaky.

Aspect Build Systems is a Bazel product partner and provides a monorepo developer platform called Aspect Workflows. By adopting Workflows Sourcegraph was able to resolve these problems, making CI 2-3x faster while reducing cloud compute costs by 40%.



History

An ex-Googler at Sourcegraph wrote a small internal position paper advocating for a move to Bazel, the open-source Build & Test tool from Google. His experience in Google's monorepo with a giant Go and React app convinced him and the team that "it really works."

Sourcegraph was confident in migrating Go and Rust code to build with Bazel, but the frontend code was perceived as a risk due to limited resources and the complexity of the code and the migration path. This led them to work with Aspect, the author of Bazel's JavaScript rules.

Aspect and Sourcegraph began working together in November 2022, as Sourcegraph started their Bazel migration journey. On December 2, the setup began:

```
commit b1a56385e5659043adc38af53cfd5a131c98b98b
Author: Jean-Hadrien Chabran <jh@chabran.fr>
Date:   Fri Dec 2 12:57:14 2022 +0100
    Initial Bazel Setup (#45052)
    * bazel: initial bazel workspace and rules_js setup
    * bazel: generate types for schema files
    * bazel: generate graphql schema file
    * Fix typo
    Co-authored-by: Derek Cormier <derek@aspect.dev>
```

Migrating to Bazel

Bazel adoption is complex. This is partially due to the inherent difficulty in migrating build systems, which are deeply integrated with the codebase. Also, Bazel is known for a steep learning curve and limited available expertise. Aspect is recognized as the community leader and provides professional services. In this case we began with hourly consulting.

The work was divided into two phases to mitigate risk. The first was a **“Frontend Bazel POC”**. We agreed on the following deliverables:

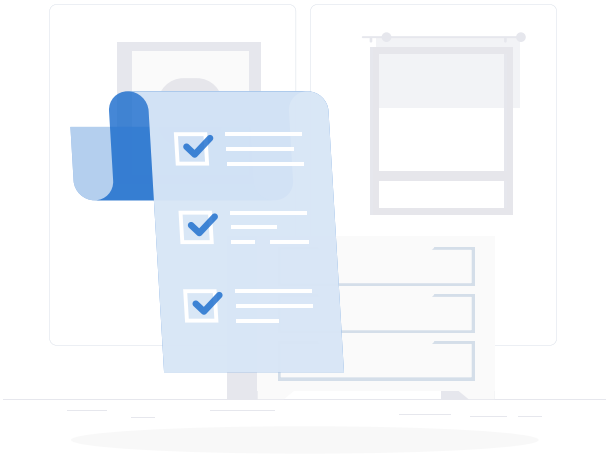
- Prove that Bazel is a viable and performant build system for Sourcegraph’s frontend code.
- Migrate a single React Frontend app to Bazel, using Webpack for bundling.
- Existing Jest tests run under Bazel.
- Demonstrate use of the Aspect CLI to generate BUILD files for TypeScript sources.
- Demonstrate a CI build that is incremental, fast, and robust to network failures.

Following the success of the POC phase, we entered a second phase: **“rules_js Bazel migration”**, scoped to include:

- Fine-grained SASS and Postcss build targets
- Complete @sourgraph/web bundle
- Mocha integration tests
- Review and optimize Sourcegraph’s golang Bazel configuration
- Documentation & Handoff

While Backend engineers were accustomed to maintaining Makefiles, BUILD file generation for JavaScript and TypeScript was particularly critical so that frontend engineers don’t have a new task of manually configuring Bazel as they change source files.

This is integrated into Aspect’s CLI tool



Aspect Workflows

Meanwhile, in February 2023, Aspect presented a demo of our turnkey CI/CD solution, Aspect Workflows. The first reaction from engineers on the call:

> *If we showed this to all the engineers at Sourcegraph there would be a mutiny if we didn't buy it.*

Aspect provided a free trial of Workflows, giving Sourcegraph time to establish target **Key Performance** Indicators:

Simple PRs should spend under **2 min** in build & test.	Median (50%ile) build&test should be **2-3x** faster.	The cost for CI compute should be significantly **reduced**.

Aspect started performing the install for Workflows in July. As part of the trial, we provided guidance to Sourcegraph with graph optimization, non-determinism fixes, and build-without-the-bytes. These all improved the codebase, reducing the workload needed for Bazel.

By November we were reporting excellent results. Even Quinn, Sourcegraph's CEO "warmed up" to Bazel:

> "I'm really warming up to Bazel after we started using it for our monorepo at Sourcegraph. I trusted our team when they decided to set it up, but only recently did I really become a believer. I trust that I can run any build/test steps in a reproducible manner."
>
> 
>
> **Quinn Slack**
> CEO at Sourcegraph

To calculate the return on investment, Sourcegraph ran Aspect Workflows side-by-side with the legacy Bazel build. During the period from November 27 to December 18 2023:

3.7 times as many builds ran in under 2 minutes (172 of build&test jobs (12%) on the legacy build, compared to 649 (44%) on Aspect Workflow

Google Compute Engine costs were reduced **40%** (December cost for legacy build $4607, Aspect Workflows $2810)

Median (50%ile) build&test was **2.4x faster** (12 minutes on legacy build, 5 minutes on Aspect Workflows)

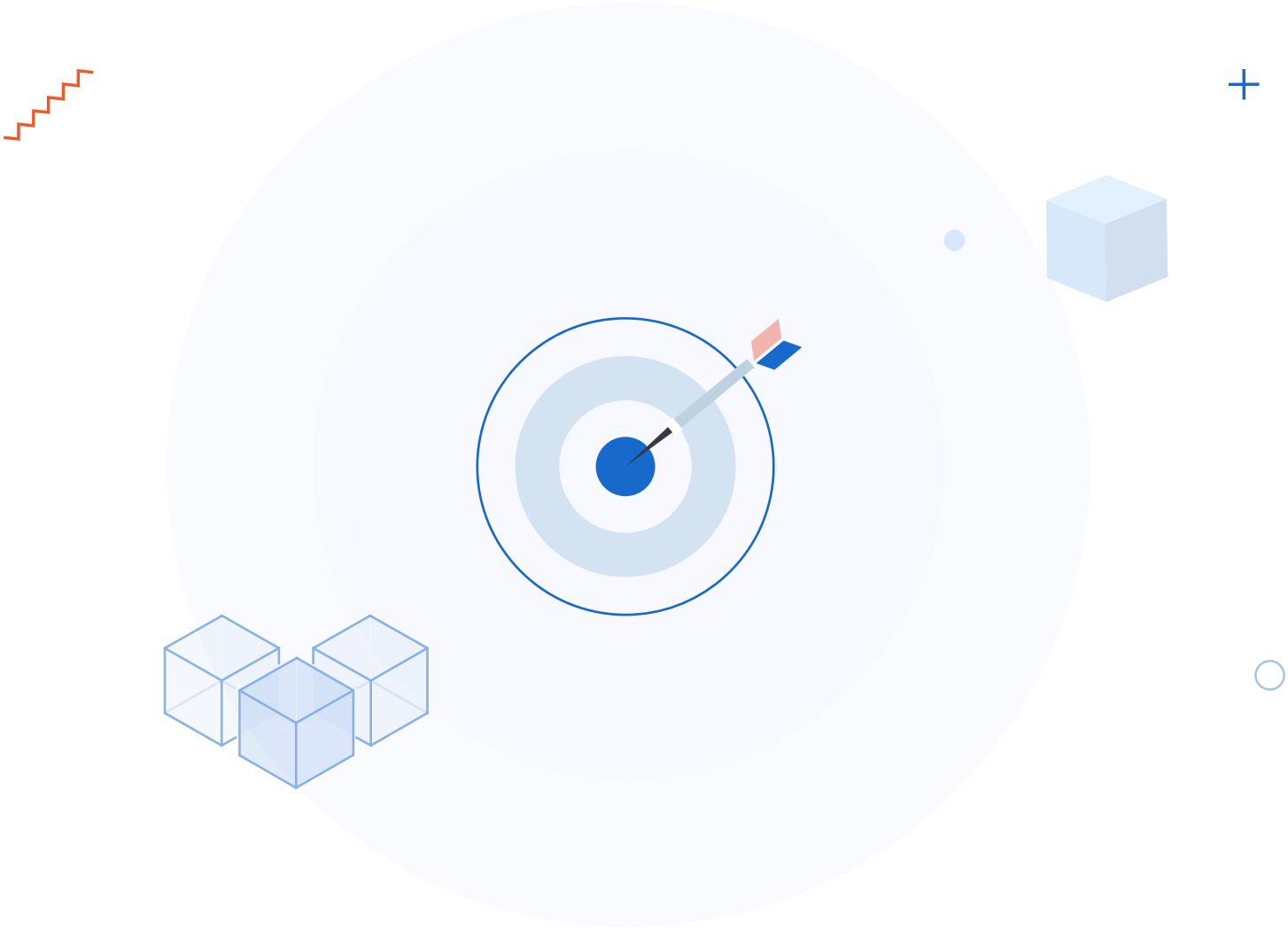
Anecdotally, engineers reported getting their fastest build on main with a **1 minute Bazel test** of the whole repo, thanks to a high cache hit rate. This is the first time engineers on the team have experienced such fast builds!

Conclusion

Aspect Workflows delivered on the promise to make builds significantly faster, while also reducing compute costs.

Workflows also enabled a couple of key features. The Continuous Delivery pipeline pushes dozens of artifacts for each green main build, but only those which were modified. Also, we augmented the Buildkite user interface with annotations showing real-time test results from Bazel's Build Events stream, so that engineers don't need to wait for the build to complete before learning of a problem.

As of March 2024, Sourcegraph is fully relying on Aspect Workflows to run Bazel in their CI/CD pipeline.




Aspect
BUILD
Sourcegraph