Achieving the promised 3x-10x Bazel Speedup

DPE Summit 2023
Speaker

Alex Eagle
Founder & Co-CEO at Aspect

- Ex-Google 2008-2020
  Tech Lead on Bazel-adjacent systems: CI and Build Results viewer.
- Bazel Community Leader.
- Founded Aspect to bring Bazel’s promised benefits to all developers!
Users have reported **3x test time reductions** and **10x faster build speeds** after switching to Bazel.
Who are these users?

Bazelcon 2022

Step change in CI performance

Bazel unlocked a 52% reduction in our build and test time in CI, while simultaneously improving our main branch stability by 5.5%.

Reduction in Avg CI Build & Test Time
Every year at BazelCon...

BazelCon 2021
... we’re reminded it’s possible!

Bazelcon 2020

As fast as
10 min
from code change to production deployment
But how fast are your Bazel builds?

Fastest no-op case?
But how fast are your Bazel builds?

99%ile slow case?
Estimated effort to build a Bazel CI/CD

We built this for our early consulting clients and know how much work it is to build it yourself.
Avoid Stale Results

Rely on Bazel Correctness
So, I just need a persistent worker?
So many de-optimizations

**Analysis phase**

Bazel needs Dep Graph

- Restarting Bazel JVM
- Accidental analysis cache discards
So many de-optimizations

Analysis phase
- Bazel needs Dep Graph
- Restarting Bazel JVM
- Accidental analysis cache discards

Low cache-hit rate
- Doing too much execution
- Non-determinism, due to
  - 3p installs
  - Tools
  - Stamping
- Cache has split-brain
So many de-optimizations

Analysis phase
Bazel needs Dep Graph
- Restarting Bazel JVM
- Accidental analysis cache discards

Low cache-hit rate
Doing too much execution
- Non-determinism, due to
  - 3p installs
  - Tools
  - Stamping
- Cache has split-brain

Hosting mistakes
It's the machine
- Spinning disks, network volumes
- No RAID
- Resource leaks
So many de-optimizations

**Analysis phase**
- Restarting Bazel JVM
- Accidental analysis cache discards

**Low cache-hit rate**
- Doing too much execution
  - Non-determinism, due to
    - 3p installs
    - Tools
    - Stamping
  - Cache has split-brain

**Hosting mistakes**
- It’s the machine
  - Spinning disks, network volumes
  - No RAID
  - Resource leaks

**Cluster mistakes**
- Distributed systems 101
  - Checkout causes invalidations
  - Not elastic / slow scale-out
  - New workers are cold
Remote Execution to the rescue?

The “performance optimization of last resort”

- Used to “paper over” too much execution: increases costs by “throwing more machines”.
- Network ingress/egress costs, especially “bad” rules like `rules_docker`

Alternatives Exist

- Test Selection ➔ less-frequent triggers

**Broken**

- Not all Bazel rules work when `host platform != exec platform`
- Much stricter hermeticity requirements

**Expensive**

- Used to “paper over” too much execution: increases costs by “throwing more machines”.
- Network ingress/egress costs, especially “bad” rules like `rules_docker`
You can have a second computer once you’ve shown you know how to use the first one.

- Paul Barham
Features
Buildcop

Monorepo -> Monobuild

Any developer can break everyone’s releases
Metrics

- Remote Cache

Cache Hits (Duration)

Cache Misses (Duration)

- Bazel Invocations

Main Branch Test Invocation Times

Test Invocation Times

Build Invocation Errors

Test Analysis Time

Cache Hit Rate

99.2

Main Branch Test Invocation Averages

4.25 min 4.25 min 2.62 min

Test Invocation Averages

1.50 min 1.49 min 1.43 min

Main Branch Build Errors

56
Selective Continuous Delivery

Green `main` build

bazel run --stamp //infrastructure/modules/workflows:release

(1:43:01 PM) [ASPECT] [delivery-manifest] 1 deliverable targets have hashes never seen on prior builds: //infrastructure/modules/workflows:release
Developer Experience
Skip it: Aspect Workflows

**Runs on your cloud, on your existing CI**

No migrations required.

**Co-managed, we carry the pager**

Avoid the perverse incentive of paying more for slower builds. No migrations required.

**Free one-month trial**

We'll prove it’s blazing fast AND pays for itself!
Fast no-op

Reduced 2 minutes of overhead
Fast 99%ile

AssemblyAI

About 2x speedup
Customer Case Study

Challenge

Bazel build&test on CircleCI taking over 10min avg, often over 30min.
CI Cost too high: 2 SWE equiv.

Solution

- Aspect Workflows trial for one month.
- Improve stability and uptime, evaluate spot instances.
- “Pretty smooth transition” for developers.
- Regular improvement through monthly releases.

Results

10x faster no-op build (from 11 min to 1 min)
2-3x Speedup of typical build & test
67% Reduced compute $ (despite higher usage)

We went from having significant limits in CI and tools to where the limits are now just due to our code.
Thank you for listening!
Next Steps

- See it in action on our OSS repositories
- Book a demo