



BazelCon

Migrating Twitter's Monorepo from Pants to Bazel - 2021



Yi Cheng (he/him)
Engineer, Twitter

Table of contents

- Recap 2020
- Functional Support
 - Translation layer
 - 3rdparty resolve
 - IDE support
- Infrastructure (focus)
 - Build cache / Build stats / analytics
 - CI (Submit Queue)

Recap 2020

- Starting point - 2020
 - 20 Millions lines of code
 - 10x in generated code
 - 2/3 of JVM, 1/3 Python, and else
 - ~2000 Engineers
- Goal - minimize user friction
 - Incremental process
 - Continuous Pants Support until Bazel can fully take over.
- Where we are today (Q4 2021)
 - 30% migrated
 - Expect to finish by early 2022

Functional Support

- Translate Pants' concepts in BUILD files into Bazel concepts.
 - Migrating Twitter's Monorepo from Pants to Bazel
 - <https://opensourcelive.withgoogle.com/events/bazelcon2020/watch?talk=day1-talk2>
- 3rdparty solution - Eugene Yokota's talk on
 - Resolving Twitter's 3rdparty/jvm with bazel-multiversion
 - <https://opensourcelive.withgoogle.com/events/bazelcon2021?talk=resolving-twitters-3rdparty-jvm-with-bazel-multiversion>
- Bazel IntelliJ Plugin based IDE Support
- Globbing issue
 - a/BUILD: **srcs = [glob(**/*)]**
 - a/x.txt
 - a/c/BUILD
 - a/c/y.txt

Infrastructure

Perspectives from

- The build tool
 - Build Cache
 - Buildstats
- Continuous Integration System (Submit Queue)

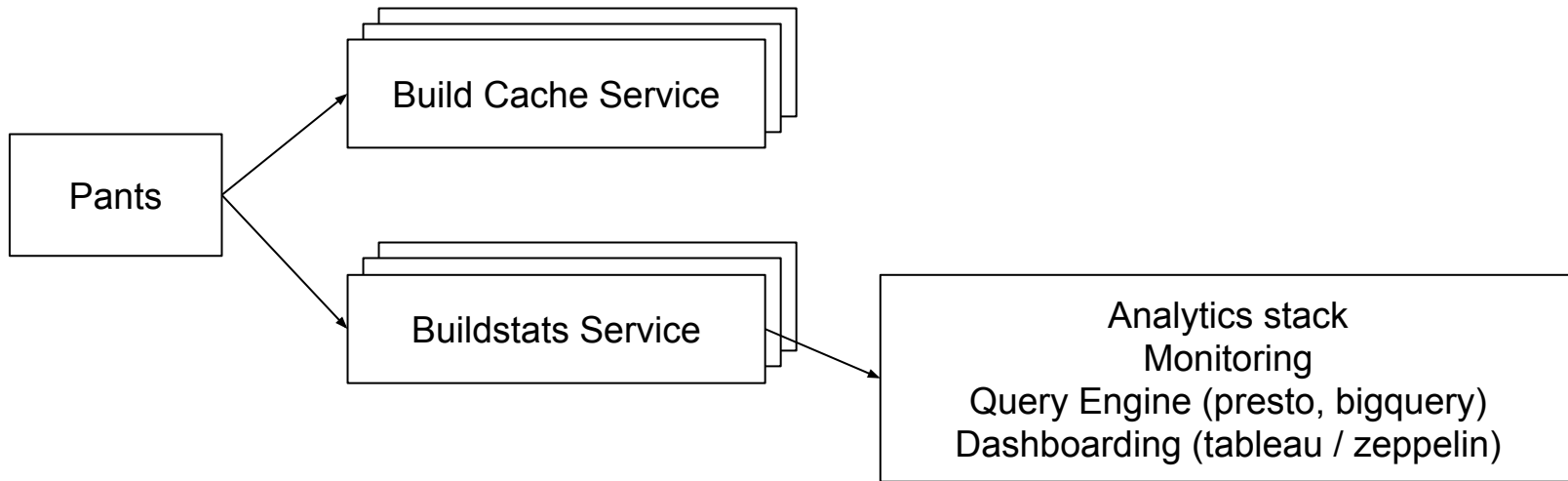
Twitter's Infra Around Pants

Build Cache

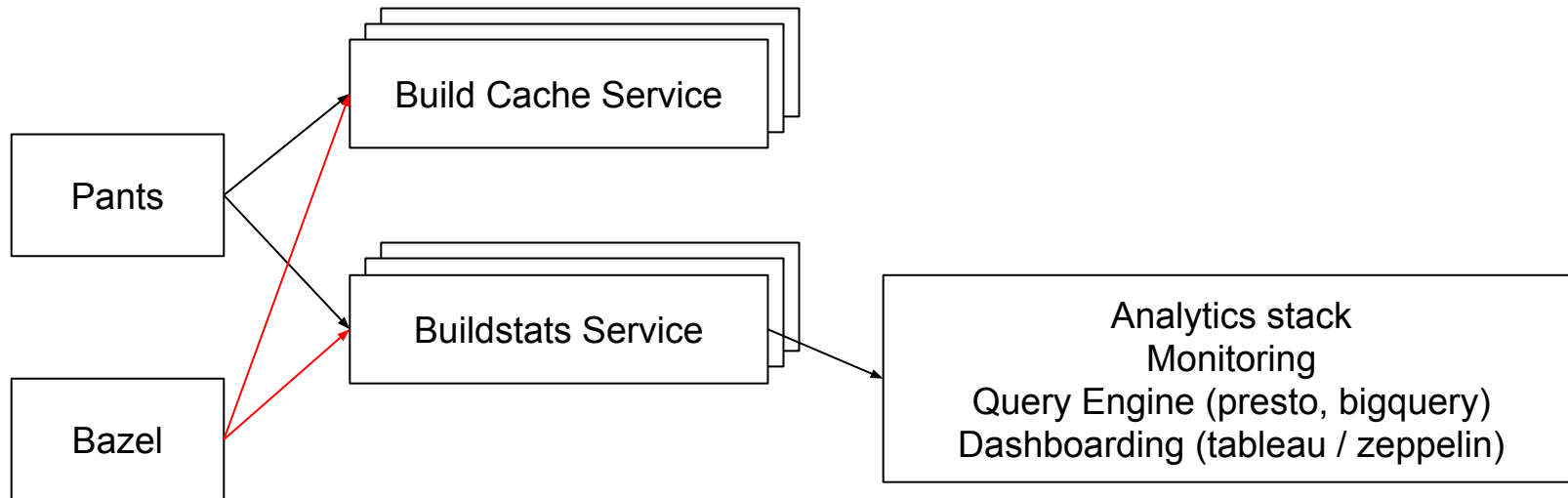
- Caching service for the build tool, so CI workers and developers can share the artifacts previously built.

Buildstats

- Analytics pipeline
- Real time monitoring



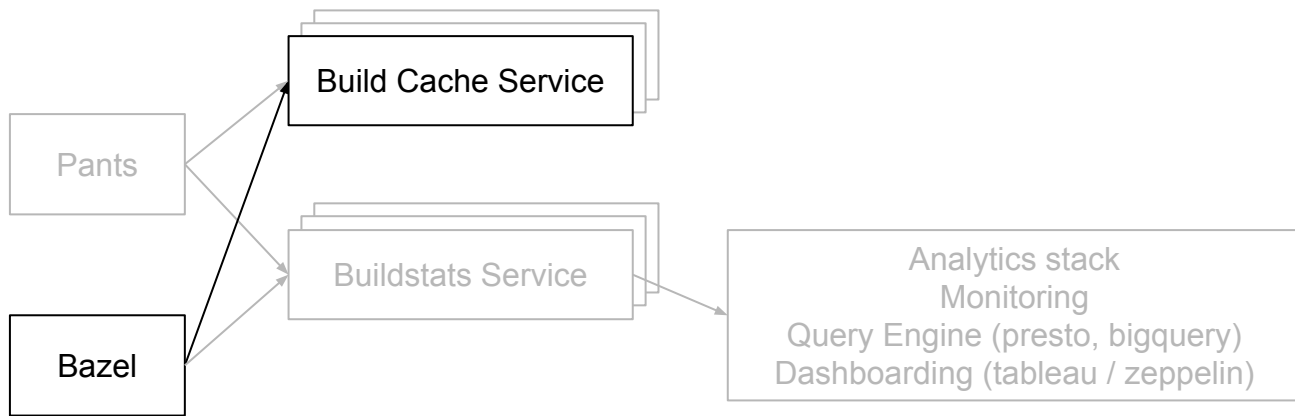
Twitter's Infra Around Bazel



Build Cache Service - Easy

Retrofit

- Pants: `--cache-read-from=https://<buildcache endpoint>/pants`
- Bazel: `--remote_cache=https://<buildcache endpoint>/bazel`



Reference: <https://docs.bazel.build/versions/main/remote-caching.html>

Credit to Chi Wang (@coeuve) on netrc support: <https://github.com/bazelbuild/bazel/pull/12130>

Build Cache Service - Gotcha

Cache Access Pattern

- Pants
 - Cache entry - per task and target
- Bazel
 - Action cache
 - CAS

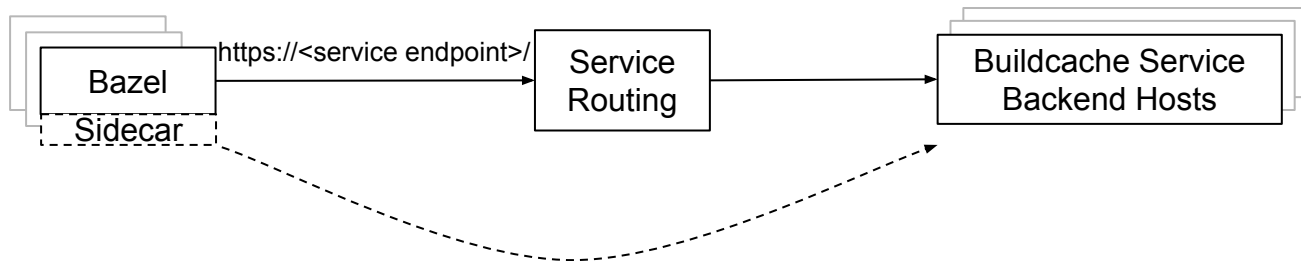
Hermeticity issues

- PATH leaks from Aurora mesos containing host specific path
 - Alternatives: docker
- rules_scala
 - Volatile statsfile writes:
https://github.com/bazelbuild/rules_scala/pull/1298

Build Cache Service - Gotcha - cont'd

Bazel's traffic causes high load on Twitter's internal service routing

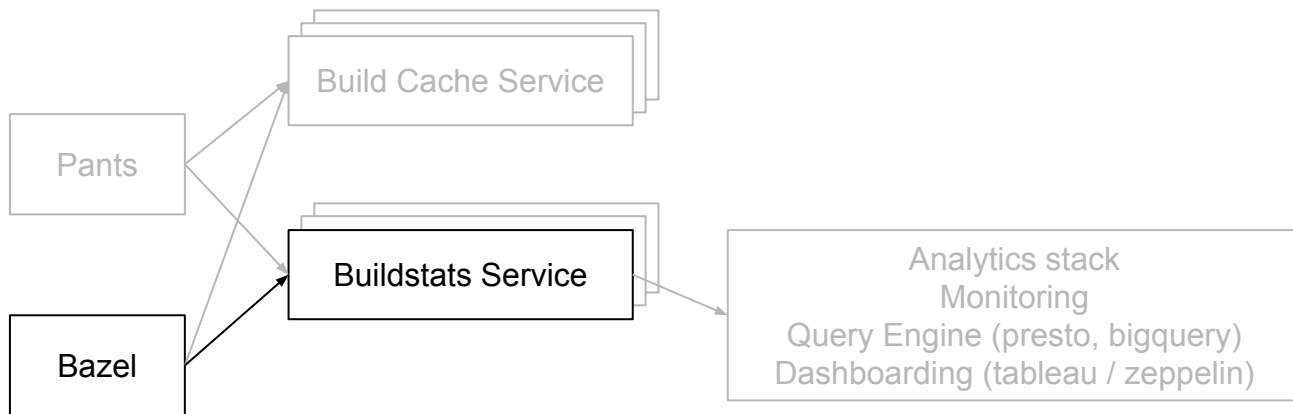
- Every http request to `https://<service endpoint>/` is routed to the backend via internal service routing
- Alternatives
 - gRPC - keep alive - fewer new requests to be made
 - Sidecar - perform service discovery locally on the client side - talk to service hosts directly.



Buildstats - Server

Retrofit

- Pants: <https://<buildstats endpoint>/pants>
- Bazel: <https://<buildstats endpoint>/bazel>



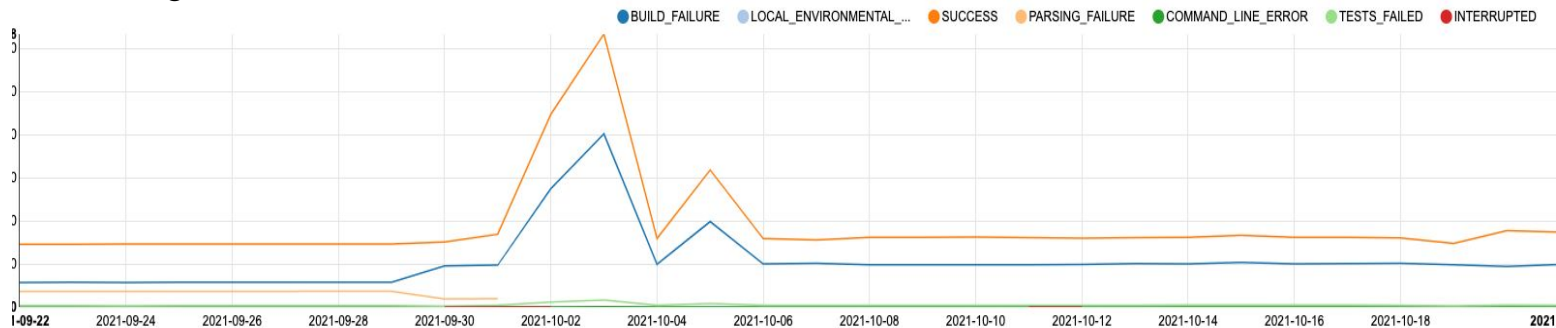
Reference: <https://docs.bazel.build/versions/main/build-event-protocol.html>

Buildstats - Client

- **--bes_backend=grpcs://<buildstats endpoint>**
- **--build_event_binary_file=output.proto**, then **curl --data-binary output.proto https://<buildstats endpoint>**

Buildstats - Dashboard

Overall Progress



Bazel build time p50/p90



Submit Queue (CI) System

Submit Queue with Pants

1. Given the 'git diff', query Pants to find the relevant targets to test (X)
2. Fan out X to a cluster of workers to run `./pants test <target>`

Submit Queue (CI) System - cont'd

Submit Queue with Bazel

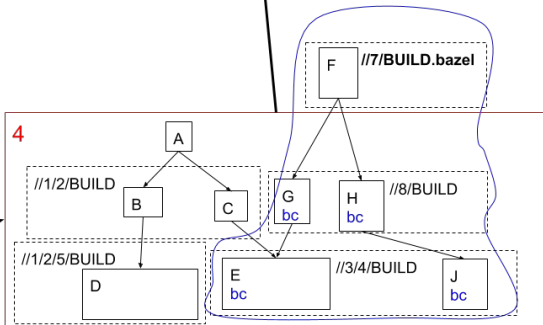
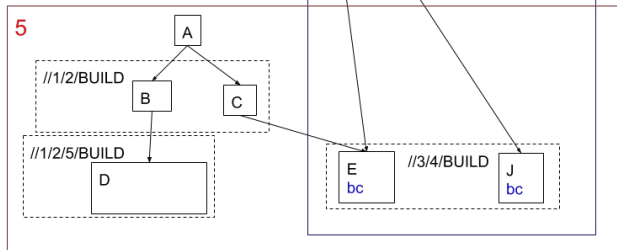
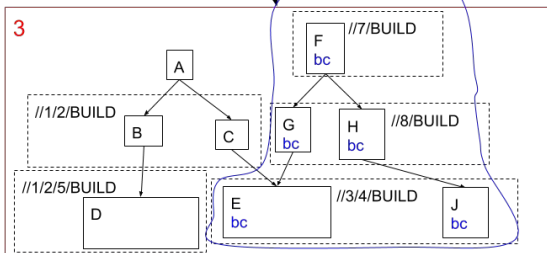
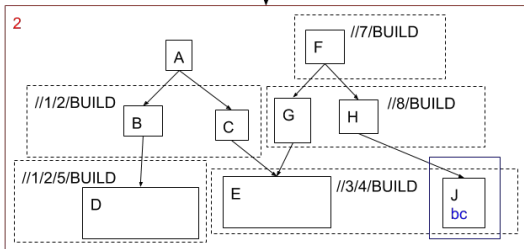
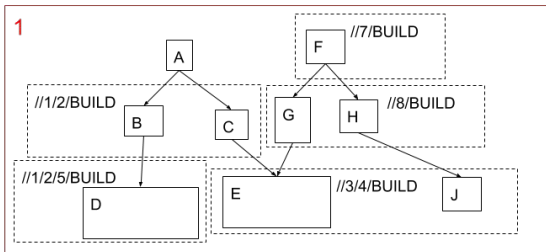
1. Given the 'git diff', query ~~Pants~~ Bazel to find the relevant targets to test (X)
 - a. Inspired by **Designing a Language Agnostic CI using Bazel Queries**
<https://opensource.withgoogle.com/events/bazelcon2020/watch?talk=day1-talk4> (Queries based on file changes)
2. Fan out X to a cluster of workers to run `./bazel test <target>`

Tip: use `--query_file` for long queries

Pants universe

Bazel buildable universe

bc = "bazel_compatible" tagged target



Submit Queue (CI) System - cont'd Bazel Buildable Universe

Takeaways

- Incremental migration is doable (at least from Pants)
- 3rdparty solution requires assessment and possible custom solution
 - Eugene's talk
<https://opensourcelive.withgoogle.com/events/bazelcon2021?talk=resolving-tweeters-3rdparty-jvm-with-bazel-multiversion>
- Bazel could introduce additional pressure on existing infrastructure

Acknowledgements



**Angela
Guardia**

metrics
analysis



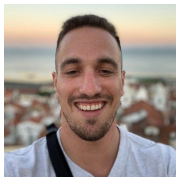
**Eugene
Yokota**

Buildcache
Server Support
for Bazel



**Drew
Gassaway**

Submit Queue
Integration
with Bazel



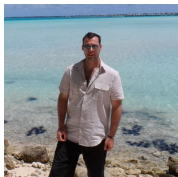
Ryan Couto

Submit Queue
Integration
with Bazel



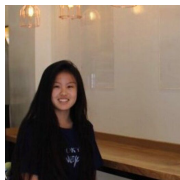
**Jeanette
Bruno**

Submit Queue
Integration
with Bazel



**Adam
Singer**

Buildcache /
Hermiticity
Investigation



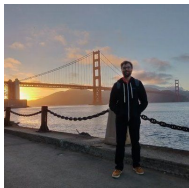
Hannah Ku

metrics
analysis



**Katya
Tyurina**

Buildcache
access pattern
investigation



**Pierre Viseu
Chevalier**

Buildcache
access pattern
investigation



BazelCon

Thank you!



Yi Cheng (he/him)
Engineer, Twitter