BoF - Java session summary

12/12/19

Summary:
About fifty people met for the Java Birds of a Feather session. After quickly creating a list of topics, the discussion turned to the first topic on the list, the state of Starlark APIs for Java. The general issue described was that the Starlark APIs are not identical to the native rules, and the Bazel team clarified that the goal is to re-write the existing Java rules in Starlark using the same API, which should also open up functionality to other rules (Scala, Kotlin, etc) that want to interoperate. Specific issues were raised and later filed (#6910).

The next topic was the ability to have cross-platform caching: Java libraries should in general not be platform dependent and should be able to be shared from, say, Linux and Mac machines. This is already a desired feature (#4714) and is part of ongoing work by the Bazel Configurability team, but is unfortunately not something that can be quickly fixed.

An issue was raised about the support for Java rules with Bazel query: Some users reported that running “bazel query” would require a local JDK even if it wasn’t needed and a remote one was available. Issues related to this have been raised and investigated (#10468).

A general discussion of IDE support showed that most users are using IntelliJ, which is decent but lacks some features. A few users wanted to create their customized builds but were frustrated by binary information in the plugin (intellij/#674). Users also noted other issues, including the problem of cleaning up unused dependencies and managing IDE upgrade cycles.

Several other topics were discussed towards the end of the time: everyone agreed that there are many benefits to using Bazel for Java code (including improved CI and remote caching, the ability to work with other languages in the same build system, and support for large repositories). Users also discussed ways to make it easier to create custom java toolchains, including being able to enable strict Java dependency checking per target instead of globally, and ways to upgrade JDK versions without waiting for the Bazel team to release support.

All in all the session went very well, and all attendees seemed to be happy with using Bazel to build and test their Java code.

Birds of a Feather - Java

Topics Of Interest

1. Starlark APIs for Java
2. Cross-platform caching of jars
3. Bazel query not requiring java installed
4. IDE support
5. Custom toolchains for Java

Starlark APIs for Java

Java sandwich is not identical to the native rules. What more needs to be exposed/happen?

- Long term goals is to expose the rules themselves as Starlark rules calling the same APIs;
- Specific things missing: creating java binary; rules_scala is also interested; forking java stub template because it is not exposed
- java_common API compile binary
- Use case: codegen of synthesizing a main class from scratch;
- java_common API with empty jars, e.g. used for exports

Cross-platform caching of jars

- CI builds are done on Linux; developers on Mac; jars are not identical for platforms
- Discussed multiple times, not only for Java; internally we can use platforms & toolchains or configuration trimming
- Plans for re-configuration how outputs are handled (long-term, not next 6months or 1year)
- Once the inputs are the same, the outputs are considered the same (the toolchain has to be identical)
- Look for existing feature requests: #4714

Bazel query not requiring java installed

- One needs java installed locally for bazel query (one doesn't specify java runtime)
- Herb, open source project
- Don't want people to install anything on their machines
- Investigate query without installed jvm - why an error? - find issue

IDE support

- Java support for Intellij; build is supposed to sync before seeing what's part of the build
- Writing new code is hard

- Unused dep problem: people pile up dependencies
- Dependency tree visualization in Java
- Managing upgrade cycles is also difficult; breaks stuff
- Bump up support for Bazel in Intellij request on Intellij side
- Developers judge bazel by IDE integration
- Precompiled protos in intellij plugin: https://github.com/bazelbuild/intellij/issues/674

Benefits for using Java with Bazel

- Multi language ecosystem
- Big monorepo
- Remote caching
- CI benefits - reasonable amount of time with Bazel
- For Java small projects may not much benefit

Custom toolchains for Java

- What the state of the art is; exposes out of the box JDK 10 and 11 toolchains
- Struggle for the Bazel team to keep up with the new cadence of Java
- Custom toolchains with the latest JDK, but possible with custom error prone checks
- Add better documentation for using new JDKs - user and developer
- How to extend JavaBuilder with new JDK - incompatible changes are possible
- JDK12 was straightforward
- Participants not using java >11

Strict java dependencies

- Is it possible to migrate per target? Easier for migrating from maven;
- rules_scala advice: Enable warn level, built automatic tools to align it, then turn error level on
- Unused deps doesn't work for third party deps

Duplicated records on the classpath

- #1071 (one version enforcement)