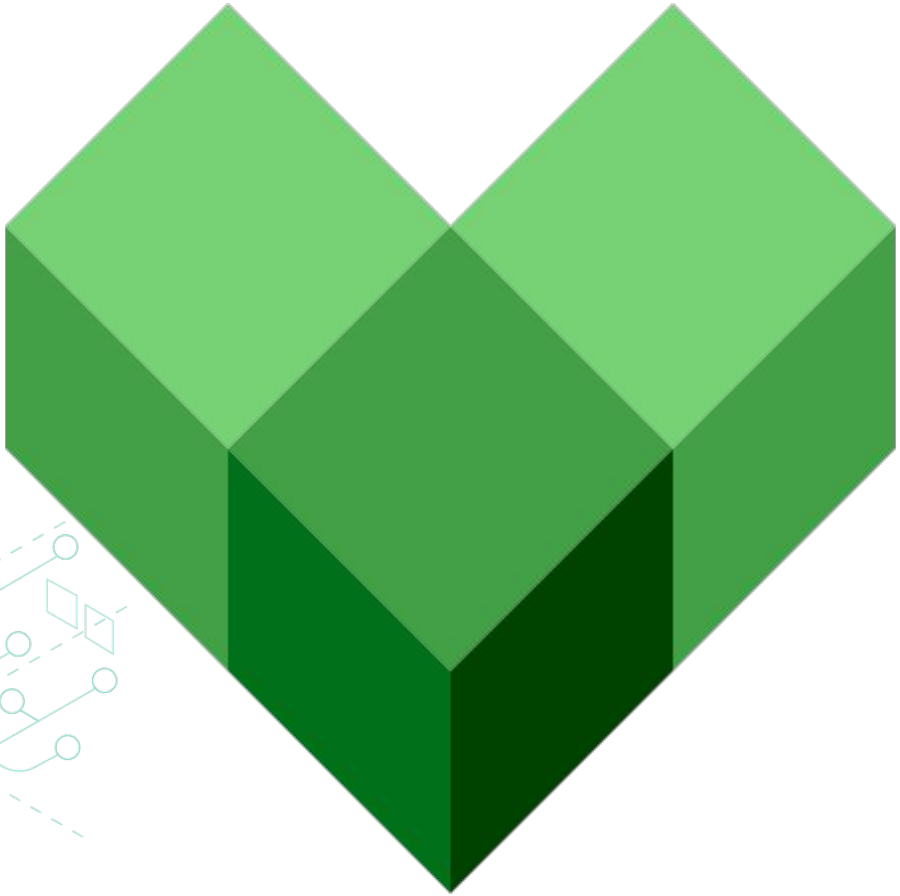
Fast builds with Bazel

2019-03-19

Maarten Hoogendoorn
@moretea.nl

Container Solutions

info@container-solutions.com
container-solutions.com



Introduction

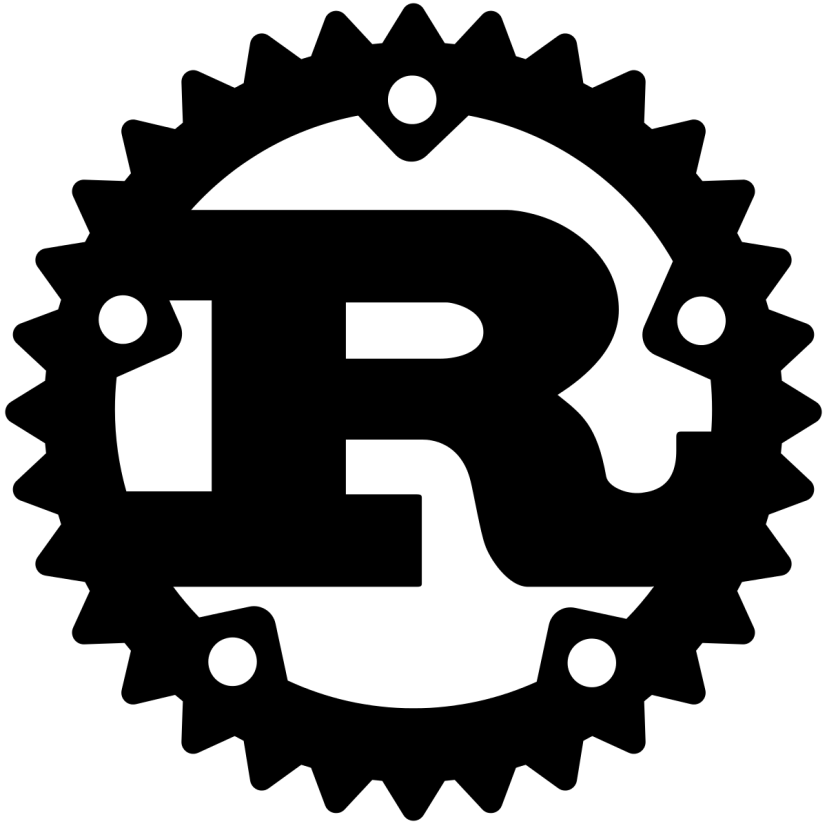
Maarten Hoogendoorn

@moretea_nl

@moretea

Rust & Nix(OS)

- Dev → DevOps → Cloud Native



?

http://rust-lang.nl

Agenda

- Build Troubles
- How Bazel Can Help
- Demo

Build Troubles

- `$ make clean; make all`
- ```
  $ ./build.sh
  cmake
  cargo build
  yarn build
  ...
  ```
  ```

Build Troubles?

Build Troubles!

Slow dev process

Dependency Management

Building dependencies is handled by a home-grown “deps” system that keeps the source code for each dependency as a tarball on S3, and stores any local modifications as patch files in our main Git repository. It uses a set of shell scripts to apply the patches and build the dependencies.

The dependencies are built locally and are cached as long as they don’t change. Once a single dependency changes, all of them are rebuilt from scratch.

This is a major pain point, and we are currently in the process of redesigning the dependency management system to improve this: Detect what has changed, and rebuild only that component and those that depend on it.

Custom Templating

Apart from using Autotools to create Makefiles, we also use a home-baked templating system to substitute configured values, such as directory paths, directly into source files. This should be eliminated and be based on runtime configuration instead.

Build Times

Build times are much longer than we would like. A full build of the system by Jenkins takes around 50 minutes. This includes:

Client Solution

1. **Trust** **No trust** → **trust** in dependency tracking
2. **Speed** **Long** → **short** build times
3. **Simple** **Complex** → **easy** to use for devs

Nice To Have (in general)

- One tool to rule them all
- Fast
- Trustworthy

Google Search I'm Feeling Lucky

Google offered in: Nederlands Frysk

kubernetes

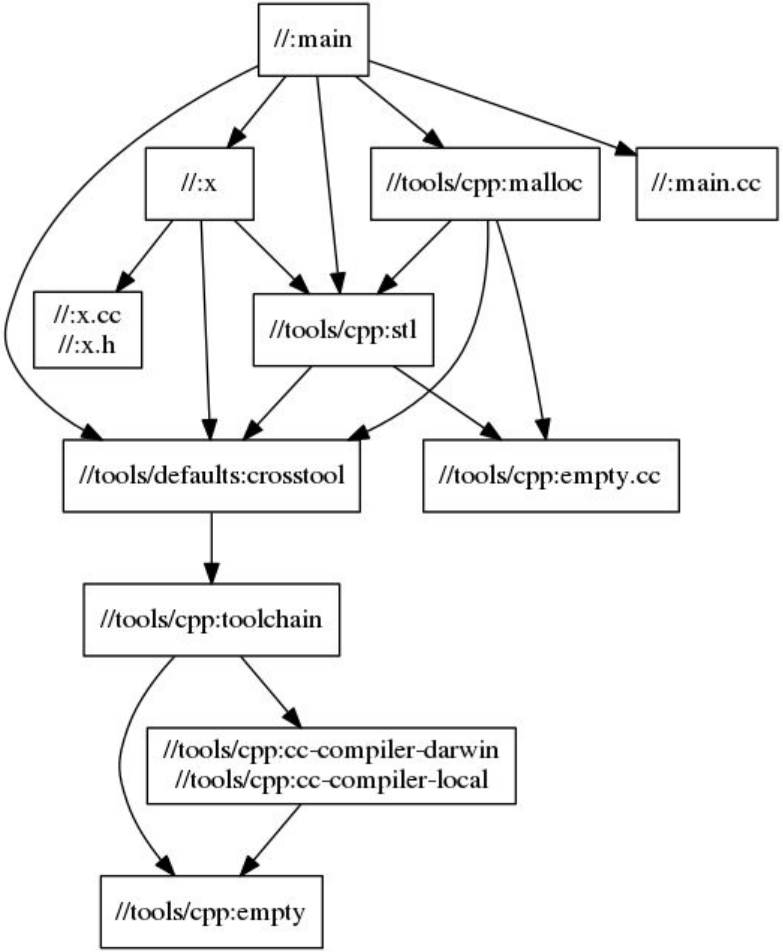
Blaze

Blaze

{Fast, Correct} - Choose two

How does Bazel help?

- DAG
- Offload to remote systems
  - Artifact cache
  - Execution


//:main
//:x
//tools/cpp:malloc
//:main.cc
//:x.cc
//:x.h
//tools/cpp:stl
//tools/defaults:crosstool
//tools/cpp:empty.cc
//tools/cpp:toolchain
//tools/cpp:cc-compiler-darwin
//tools/cpp:cc-compiler-local
//tools/cpp:empty

BazelCon 2018 Day 1: Faster Builds With Remote Execution and Caching

https://www.youtube.com/watch?v=MyuJRUwT5LI

Demo!

Thanks!

http://rust-lang.nl

Extra

modal-presenter

github.com/moretea/modal-presenter