Nix Overview

Nix Features

- Add more Nix features from `http://nixos.org/nix/about.html`
- Diagrams.

Architecture of Nix

Under the hood a little bit

- /nix/store
- Perhaps the crazy linker thing.
- How hashing works. Why binary subtiution is the thing.
- A -> B. What are the inputs?
  - What inputs?
  - What outputs?
  - hashing for the store
- Caching/Memoisation.

Nix Commands

- Installing haskell tools like ghc (different versions) and nix2cabal.
- Usage of nix-shell
- Usage of nix2cabal

Accelerating Haskell Development with Nix

Where are we now?

- Using cabal sandboxes.
- Perhaps some shared sandboxes.
- Waiting for builds is no fun.
- Wasting time building ‘lens‘ for each of your projects that uses it is not good.
- Let’s not accept the status quo.
- One option is to use Halcyon – a build cache for Cabal.

What's not good?

- Long build times.
- Building the same dependencies over and over again in different sandboxes.
- These sandboxes could be on your machine or your team members machine.
- Or on the build box.
- There is wastage of time but also of disk space.
- With SSDs, disk space isn't as cheap as it used to be.

Demo

- Instant ‘lens‘ environment.
- Instant ‘reflex‘ environment with tryreflex.
  `https://github.com/ryantrinkle/try-reflex`
- Work through hutton-razor.

Downsides of Nix.

- Still early (but you'd be getting in at a great time)

Nix Workshop at Hack Night

- Get set up with NixOS (and perhaps Nix on Mac/Linux/*BSD if you're more adventurous).
- Learn how to set up a modern Haskell development environment.
- Hopefully, learn how to work on sources to multiple dependencies in your tree.

References

- Ollie Charles
  - How I develop with Nix
  - `http://wiki.ocharles.org.uk/Nix`
- Peter Simons
  - Nix loves Haskell — slides — slides.md
  - Haskell User Guide for Nixpkgs
- `https://nixos.org/nix/manual/`