Slacker paper notes

- **Abstract**: Lazy-loading of docker container chunks

Introduction

- HelloBench, a `service_healthy` benchmark
- MultiMake, a script that compiles 16 binaries with different settings
- Slacker, a docker storage driver

Docker Background

- Docker file systems consist of layers, with each layer being applied to the previous to actually get the final state
- This is not so far from git

Driver methods

Method	Description
Get(id)=dir	mount "id" layer file system, return mount point
Put(id)	unmount "id" layer file system
Create(parent, id)	logically copy "parent" layer to "id" layer
Diff(parent, id)=tar	return compressed tar of changes in "id" layer
ApplyDiff(id, tar)	apply changes in tar to "id" layer

- AUFS, *Another Union File System* is the default for docker

> Union filesystem: Filesystem applied on top of another. My project TeleDrive creates a layman's implementation of a union filesystem

Thoughts/Opinions

Systems like Nix promise to solve the very issues detailed in this paper: Namely that of present-day containers being bloated, both because they present duplication and redudant usage of resources.

The basic premise of using a container is to avoid the headache that comes with managing applications installed on bare metals, regardless of the efficiency that

comes with it (in terms of saved space).

Nix promises to solve this by having a centralised read-only "store", with everyone being able to read it. Replit is an organisation that implements this in production, allowing a shared nix store for all their Repls, which were earlier backed by docker containers.

Yet, apart from replit, use of Nix is not so widespread, and docker remains the container system of choice. The problems associated with Nix include that the standard Linux FHS is not followed, so every pre-existing ELF binary needs to be patched to be made compatible with NixOS. The nixpkgs project tries to do this for all software, but they still lag behind upcoming releases: Nix will not catch up until the upstream developers themselves package for it.

Slacker faces a similar problem if it were to be implemented: If anything in its implementation created edge cases that made it break in comparision to the "canonical implementation" of the Docker FS driver, it would continue to always lag behind the official version.

Another example is that of `yarn` or `pnpm` not being as widespread as `npm`, even though they offer space benefits compared to `npm` in terms of file duplication for `node_modules` — some specific modules don't play nice with `yarn` or `pnpm`.