

The nix package manager

Declarative package management on Linux/OSX

Tobias Mayer

March 8, 2018

TomTom Development Germany

language (pip, cargo)

system (apt, dnf, brew)

language (pip, cargo)

- + declarative
- build time / run time discrepancy
- incomplete

system (apt, dnf, brew)

Motivation

language (pip, cargo)

- + declarative
- build time / run time discrepancy
- incomplete

system (apt, dnf, brew)

- + self-sufficient
- tied to platform
- stateful

Motivation

language (pip, cargo)

- + declarative
- build time / run time discrepancy
- incomplete

system (apt, dnf, brew)

- + self-sufficient
 - tied to platform
 - stateful
- not reproducible
 - not deterministic

The nix ecosystem

- package manager
- expression language
- nixpkgs
- nixos
- nixops
- disnix
- hydra



The nix ecosystem

- package manager
- expression language
- nixpkgs
- nixos
- nixops
- disnix
- hydra



- Source based distribution
- Unique per-package install location
/nix/store/hash-name-version
example: /nix/store/86blj9iqyxwmdgkn3dyrpib1gkbmz91v-bash-4.4-p5

Approach

- Source based distribution
- Unique per-package install location

/nix/store/hash-name-version

example: /nix/store/86blj9iqyxwmdgkn3dyrpib1gkbmz91v-bash-4.4-p5

-> multiple variants of a package available simultaneously

Approach

- Source based distribution
- Unique per-package install location

/nix/store/hash-name-version

example: /nix/store/86blj9iqyxwmdgkn3dyrpib1gkbmz91v-bash-4.4-p5

- > multiple variants of a package available simultaneously
- > no duplication

Approach

- Source based distribution
- Unique per-package install location
/nix/store/hash-name-version
example: /nix/store/86blj9iqyxwmdgkn3dyrpib1gkbmz91v-bash-4.4-p5
- > multiple variants of a package available simultaneously
- > no duplication
- > transparent binary caching

The nix package manager

Commands

- `nix-instantiate`
- `nix-store`
- `nix-shell`
- `nix-build`
- `nix-env`
- `nix-collect-garbage`
- `nix-hash`
- ...
- `nix` (added with version 2, currently beta)

- github.com/NixOS/nixpkgs
- Public collection of nix expressions
- sorted into channels:
 - nixpkgs-unstable
 - nixos-unstable
 - nixos-17.09
 - nixos-18.03
 - ...

example expression

```
1 { stdenv, fetchurl, perl }:  
2  
3 stdenv.mkDerivation {  
4   name = "hello-2.10";  
5   builder = ./builder.sh;  
6   src = fetchurl {  
7     url = ftp://ftp...nl/  
8           hello-2.10.tar.gz;  
9     sha256 = "1...md7jsfd8pa";  
10  };  
11  buildInputs = [ perl ];  
11 }
```

.drv file

```
Derive (
  [("out", "/nix/store/kmwd1~hello-2.10", "", "")],
  [("/nix/store/2zy7f~hello-2.10.tar.gz.drv", ["out"]),
    ("/nix/store/dxlr6~stdenv.drv", ["out"]),
    ("/nix/store/qkf7y~bash-4.4-p12.drv", ["out"]),
    ("/nix/store/wy1na~perl-5.24.3.drv", ["out"])],
  ["/nix/store/9krlz~default-builder.sh"],
  "x86_64-linux",
  "/nix/store/zqh3l~bash-4.4-p12/bin/bash",
  ["-e", "/nix/store/9krlz~default-builder.sh"],
  [("buildInputs", ""),
    ("builder", "/nix/store/zqh3l~bash-4.4-p12/bin/bash"),
    ("stdenv", "/nix/store/l3nz0~stdenv"),
    ...
    ("system", "x86_64-linux")]
)
```

.drv file (outputs)

```
Derive (  
  [("out", "/nix/store/kmwd1~hello-2.10", "", "")],  
  [("/nix/store/2zy7f~hello-2.10.tar.gz.drv", ["out"]),  
    ("/nix/store/dxlr6~stdenv.drv", ["out"]),  
    ("/nix/store/qkf7y~bash-4.4-p12.drv", ["out"]),  
    ("/nix/store/wy1na~perl-5.24.3.drv", ["out"])],  
  ["/nix/store/9krlz~default-builder.sh"],  
  "x86_64-linux",  
  "/nix/store/zqh3l~bash-4.4-p12/bin/bash",  
  ["-e", "/nix/store/9krlz~default-builder.sh"],  
  ["buildInputs", ""],  
  ("builder", "/nix/store/zqh3l~bash-4.4-p12/bin/bash"),  
  ("stdenv", "/nix/store/l3nz0~stdenv"),  
  ...  
  ("system", "x86_64-linux")]  
)
```


.drv file (inputs)

```
Derive (  
  [("out", "/nix/store/kmwd1~hello-2.10", "", "")],  
  [("/nix/store/2zy7f~hello-2.10.tar.gz.drv", ["out"]),  
    ("/nix/store/dxlr6~stdenv.drv", ["out"]),  
    ("/nix/store/qkf7y~bash-4.4-p12.drv", ["out"]),  
    ("/nix/store/wy1na~perl-5.24.3.drv", ["out"])],  
  ["/nix/store/9krlz~default-builder.sh"],  
  "x86_64-linux",  
  "/nix/store/zqh3l~bash-4.4-p12/bin/bash",  
  ["-e", "/nix/store/9krlz~default-builder.sh"],  
  [("buildInputs", ""),  
    ("builder", "/nix/store/zqh3l~bash-4.4-p12/bin/bash"),  
    ("stdenv", "/nix/store/l3nz0~stdenv"),  
    ...  
    ("system", "x86_64-linux")]  
)
```

.drv file (builder+arguments)

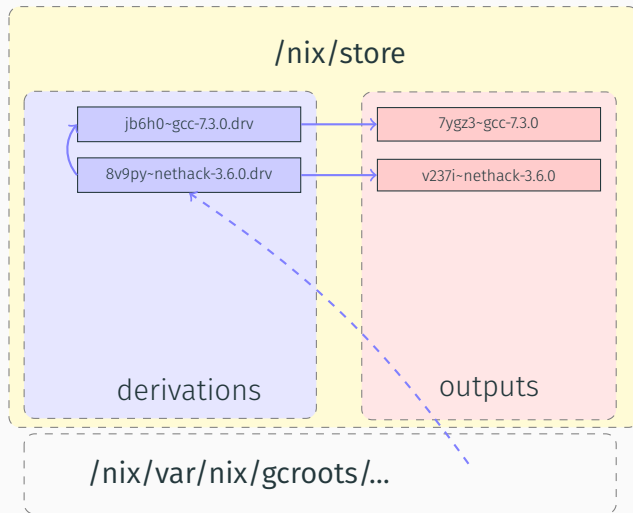
```
Derive (  
  [("out", "/nix/store/kmwd1~hello-2.10", "", "")],  
  [("/nix/store/2zy7f~hello-2.10.tar.gz.drv", ["out"]),  
    ("/nix/store/dxlr6~stdenv.drv", ["out"]),  
    ("/nix/store/qkf7y~bash-4.4-p12.drv", ["out"]),  
    ("/nix/store/wy1na~perl-5.24.3.drv", ["out"])],  
  ["/nix/store/9krlz~default-builder.sh"],  
  "x86_64-linux",  
  "/nix/store/zqh3l~bash-4.4-p12/bin/bash",  
  ["-e", "/nix/store/9krlz~default-builder.sh"],  
  ["buildInputs", ""],  
  ("builder", "/nix/store/zqh3l~bash-4.4-p12/bin/bash"),  
  ("stdenv", "/nix/store/l3nz0~stdenv"),  
  ...  
  ("system", "x86_64-linux")]  
)
```

.drv file (environment variables)

```
Derive (  
  [("out", "/nix/store/kmwd1~hello-2.10", "", "")],  
  [("/nix/store/2zy7f~hello-2.10.tar.gz.drv", ["out"]),  
    ("/nix/store/dxlr6~stdenv.drv", ["out"]),  
    ("/nix/store/qkf7y~bash-4.4-p12.drv", ["out"]),  
    ("/nix/store/wy1na~perl-5.24.3.drv", ["out"])],  
  ["/nix/store/9krlz~default-builder.sh"],  
  "x86_64-linux",  
  "/nix/store/zqh3l~bash-4.4-p12/bin/bash",  
  ["-e", "/nix/store/9krlz~default-builder.sh"],  
  [("buildInputs", ""),  
    ("builder", "/nix/store/zqh3l~bash-4.4-p12/bin/bash"),  
    ("stdenv", "/nix/store/l3nz0~stdenv"),  
    ...  
    ("system", "x86_64-linux")]  
)
```

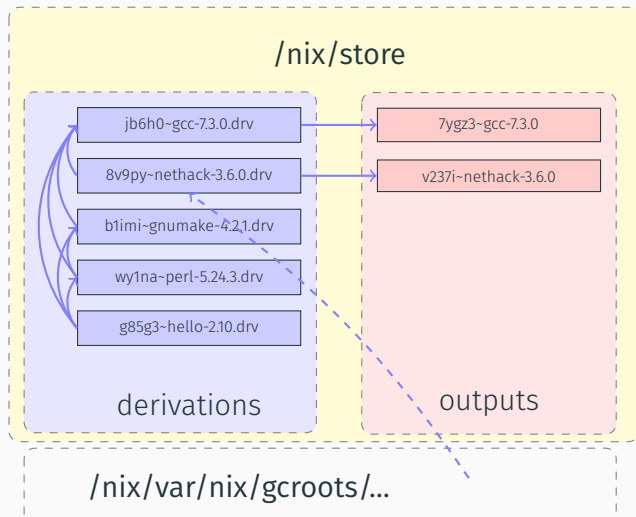
The store (conceptual)

initial state



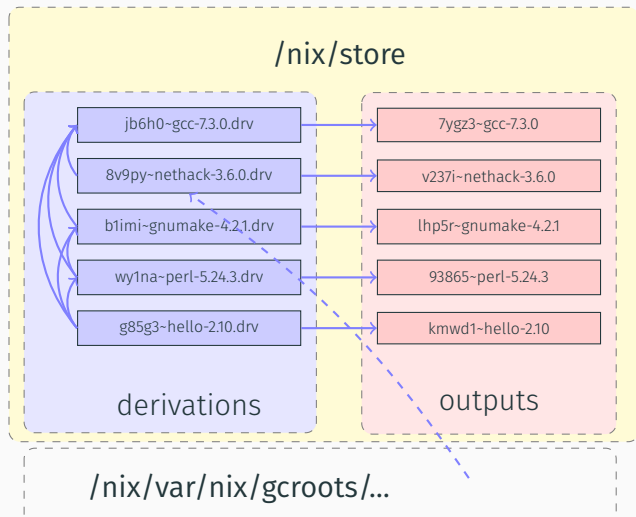
The store (conceptual)

```
nix-instantiate -E 'with import <nixpkgs> { }; hello'
```



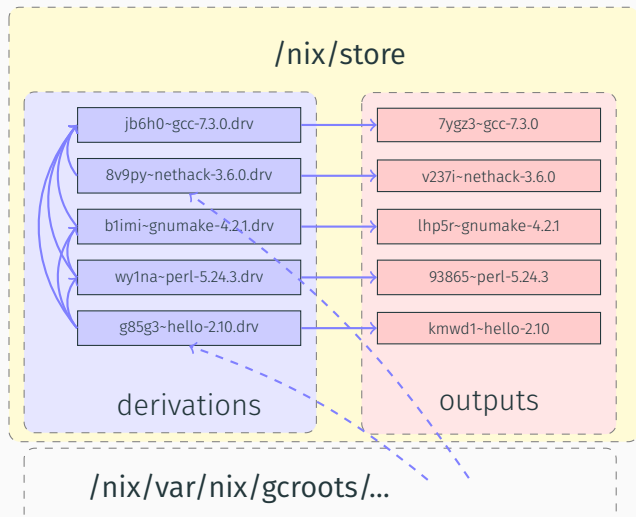
The store (conceptual)

```
nix-store -r /nix/store/g85g3~hello-2.10.drv
```



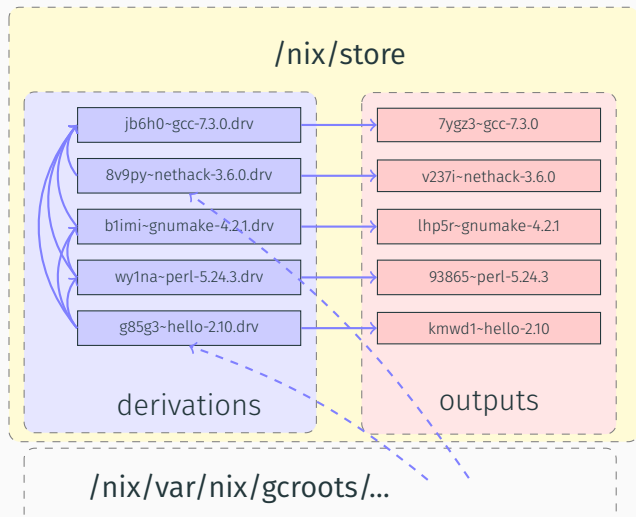
The store (conceptual)

```
ln -s /nix/store/g85g3~hello-2.10.drv /nix/var/nix/gcroots/
```



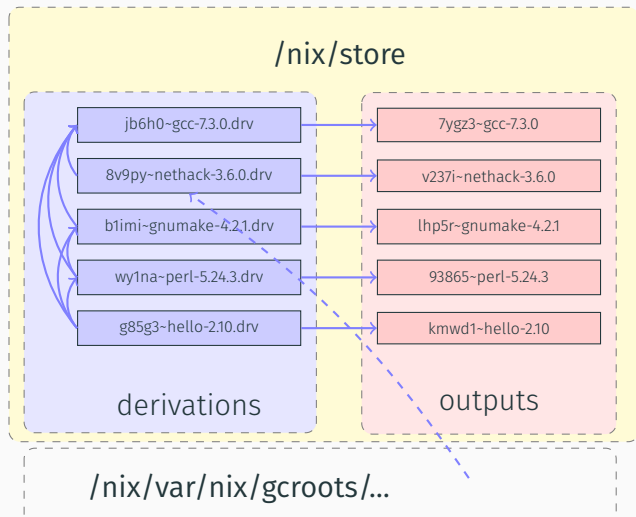
The store (conceptual)

`nix-store -gc`



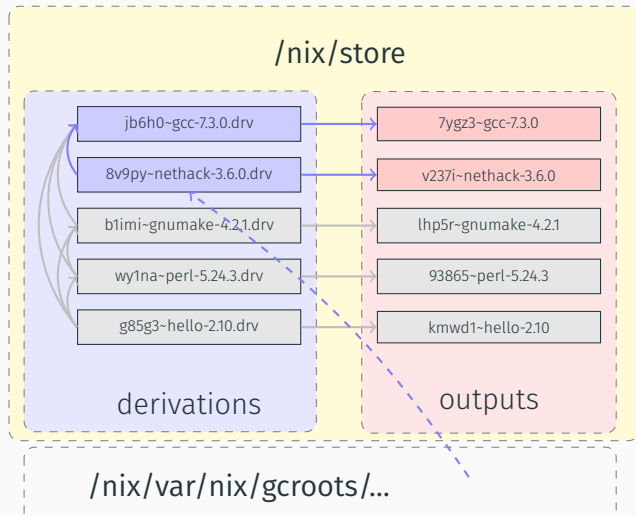
The store (conceptual)

```
rm /nix/var/nix/gcroots/g85g3~hello-2.10.drv
```



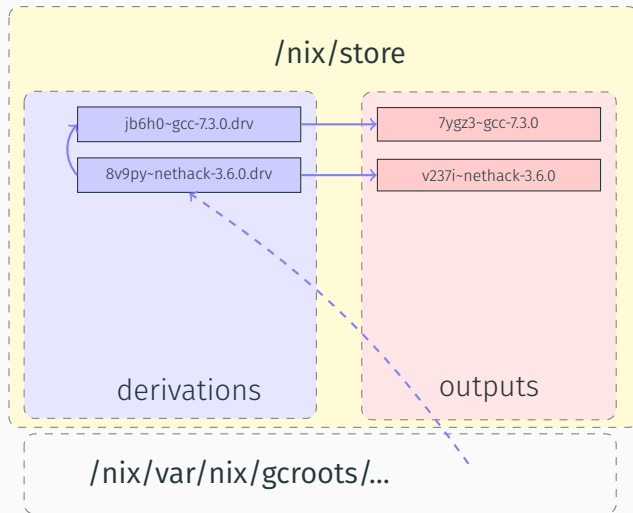
The store (conceptual)

nix-store -gc



The store (conceptual)

final state



expression language

- functional
- pure
- lazy
- numbers, booleans, strings, paths
- lists: [3 "hello" true]
- sets: { x = 123; text = "Hi!"; y = f { foo = "bar"; }; }
- function def: funcname = pattern: body
- function call: funcname argument
- function def (2): funcname = { pattern1, pattern2 ? defaultval }: body
- function call (2): funcname { argument set }

nix-shell expression

shell.nix:

```
{}:  
let  
  pkgs = import (fetchTarball  
    https://github.com/NixOS/nixpkgs/archive/17.03.tar.gz) {};  
in with pkgs; {  
  environment = stdenv.mkDerivation {  
    name = "myProject";  
    nativeBuildInputs = [ cmake ];  
    buildInputs = [ boost ];  
  };  
}
```

use with 'nix-shell -pure -indirect -add-root gcroots/shell.gc'

- parameterized shells
- link time deps with the same toolset
- write expressions for dependencies
- create docker images from expressions

Thank you!

github.com/tobimpub/presentations/20180308-devday-nix