

Functional Programming: Real World Performance, Nix and Warp Server

John Doe

August 20, 2020

Outline

How Familiar is everyone with FP

Disclaimer, not an expert

- ▶ Logical fallacies will be used, not too fond of this but $\neg \setminus () _ / \neg$.
- ▶ Linux user for 7 years now
 - ▶ Ubuntu
 - ▶ Proxmox
 - ▶ ArchLinux
 - ▶ Centos (server management)

Choose some project

https:

[//github.com/search?q=filename%3Ashell.nix&type=Code](https://github.com/search?q=filename%3Ashell.nix&type=Code)

Goals

- ▶ Functional Programming **Principles** (not only languages)
- ▶ Academic mental exercise (hope not too boring):D
 - ▶ not necessarily useful
 - ▶ exposure to a **what if?** world

What is the problem?

- ▶ Modern smart phones vs old phones
- ▶ What OS has everyone used?
 - ▶ windows
 - ▶ ubuntu/mac apt-get brew

What is the problem?

- ▶ multiple versions
- ▶ mutability
 - ▶ mysql-python
- ▶ not accurate dependency graph
- ▶ dependency hell

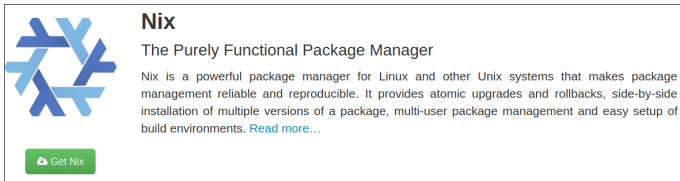
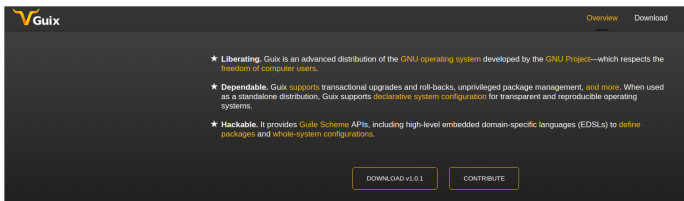
Some modern day package management systems

Package manager	Distributions
apt, apt-get	Debian, Ubuntu
rpm, yum	Redhat, Centos
pacman	ArchLinux
brew	MacOS

What it should/could/would have been?

- ▶ Imagine now that we implemented all the things of a functional programming language to create a functional package management system?
- ▶ What can we do with this?

GUIX vs Nix



Introducing Nix Package Management

How does nix actually work?

Nix expressions

- ▶ functional expressions, not general purpose please do not program things with it
- ▶ comes with its own BNF grammar

expressions

<code>e ::= x</code>	identifier
<code>nat str</code>	literal
<code>[e*]</code>	list
<code>rec[?] {b*}</code>	(optionally recursive) attribute set
<code>let b* in e</code>	local declarations
<code>e.x</code>	attribute selection
<code>x : e</code>	plain λ -abstraction
<code>{fs[?]} : e</code>	λ -abstraction pattern-matching an attribute set
<code>e e</code>	function application
<code>if e then e else e</code>	conditional
<code>with e₁; e₂</code>	add attributes from set e_1 to lexical scope of e_2
<code>(e)</code>	grouping

bindings

<code>b ::= ap = e;</code>	allows concise nested attribute sets, e.g. <code>x.y.z = true</code>
<code>inherit x*;</code>	copy value of attribute x from lexical scope

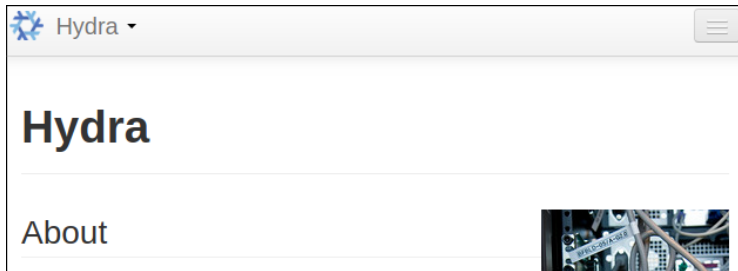
Nix as infrastructure (imagination)

- ▶ how might one use nix in JPMC's infrastructure?

Main componenets

- ▶ Hydra caching
- ▶ Dependency management
- ▶ Ease of use
 - ▶ nix-shell
- ▶ Security

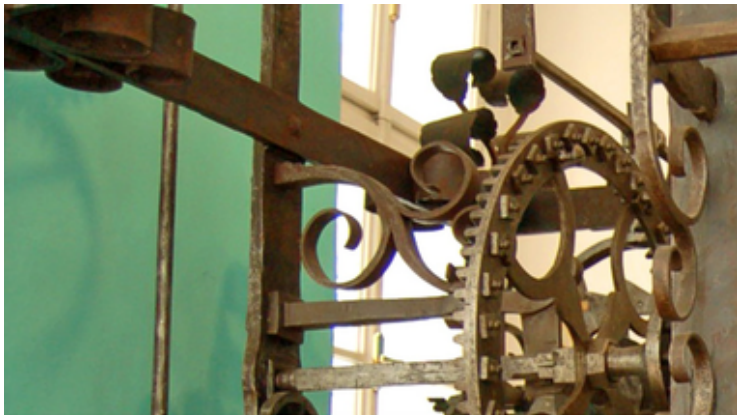
Caching build farm or cachix



references

- ▶ [HTML] Nix: A Safe and Policy-Free System for Software Deployment.
 - ▶ E Dolstra, M De Jonge, E Visser - [usenix.org](https://www.usenix.org/conference/lisa2004/presentation/dolstra)
 - ▶ <https://nixos.org/~eelco/pubs/nspfssd-lisa2004-final.pdf>
- ▶ [PDF] A Purely Functional Linux Distribution - NixOS
 - ▶ E Dolstra
 - ▶ <https://nixos.org/~eelco/pubs/nixos-jfp-final.pdf>
- ▶ Hydra - NixOS
 - ▶ <https://nixos.org/~eelco/pubs/hydra-scp-submitted.pdf>

Part 2 Warp optimization



The Performance of Open Source Applications

Speed, Precision, and a Bit of Serendipity