Functional Package Management with GNU Guix

Ricardo Wurmus
rekado@elephly.net

OpenTechSummit
14 May 2015
Good idea

- easy to install, upgrade, remove software
- dependency resolution
- centrally maintained repositories
Common problems

- outdated packages
- version conflicts
- changes affect all users
- potentially dangerous
- trust
Partial solutions

- **third-party repositories**
  EPEL, PPAs, ...

- **manual compilation**
  install to custom prefix, static linking

- **language-specific package systems**
  gem, cabal, pip, cpan, npm ...

- **build your own system package**
  RPM, deb, PKGBUILD, ...

- **meta package managers**
  e.g. fpm generating RPM, deb, gem

- **giving up**
  virtual machines, “app images”, snapshots
Partial solutions

- **third-party repositories**
  EPEL, PPAs, ...

- **manual compilation**
  install to custom prefix, static linking

- **language-specific package systems**
  gem, cabal, pip, cpan, npm ...

- **build your own system package**
  RPM, deb, PKGBUILD, ...

- **meta package managers**
  e.g. fpm generating RPM, deb, gem

- **giving up**
  virtual machines, “app images”, snapshots
Partial solutions

- third-party repositories
  EPEL, PPAs, ...

- manual compilation
  install to custom prefix, static linking

- language-specific package systems
  gem, cabal, pip, cpan, npm ...

- build your own system package
  RPM, deb, PKGBUILD, ...

- meta package managers
  e.g. fpm generating RPM, deb, gem

- giving up
  virtual machines, “app images”, snapshots
Partial solutions

- third-party repositories
  EPEL, PPAs, ...
- manual compilation
  install to custom prefix, static linking
- language-specific package systems
  gem, cabal, pip, cpan, npm ...
- build your own system package
  RPM, deb, PKGBUILD, ...
- meta package managers
  e.g. fpm generating RPM, deb, gem
- giving up
  virtual machines, “app images”, snapshots
Partial solutions

- **third-party repositories**
  EPEL, PPAs, ...

- **manual compilation**
  install to custom prefix, static linking

- **language-specific package systems**
  gem, cabal, pip, cpan, npm ...

- **build your own system package**
  RPM, deb, PKGBUILD, ...

- **meta package managers**
  e.g. fpm generating RPM, deb, gem

- **giving up**
  virtual machines, “app images”, snapshots
Partial solutions

- **third-party repositories**
  EPEL, PPAs, ...

- **manual compilation**
  install to custom prefix, static linking

- **language-specific package systems**
  gem, cabal, pip, cpan, npm ...

- **build your own system package**
  RPM, deb, PKGBUILD, ...

- **meta package managers**
  e.g. fpm generating RPM, deb, gem

- **giving up**
  virtual machines, “app images”, snapshots
gnu

→

u
Functional package management

- no **global** values: /bin, /usr/include, /usr/lib, ...
- **purity**: only declared inputs are visible at build time
- **reproducible results**: build outputs can be cached and substituted; automatic deduplication!
- **immutable results** without **side effects**: nothing outside of the cache and internal state is modified
build processes
chroot, separate UIDs

build daemon

RPCs

Guile, make, etc.

Guile, make, etc.

Guile, make, etc.

Guix client

(guix packages)

(guix store)
(define hello
 (package
  (name "hello")
  (version "2.8")
  (source (origin
           (method url-fetch)
           (uri (string-append
                  "mirror://gnu/.../hello-
                  ".tar.gz"))
           (sha256 (base32 "0wqd...dz6"))))
  (build-system gnu-build-system)
  (synopsis "Hello, GNU world: An example GNU package")
  (description "Produce a friendly greeting.")
  (home-page "http://www.gnu.org/software/hello/")
  (license gpl3+)))
version 0.8.2 is out now
quickly growing collection of package recipes (1800+)
install the distribution
use it, report bugs, add packages
share your ideas!