Solving the Deployment Crisis with GNU Guix

Christopher Allan Webber & David Thompson
Saturday, March 19th, 2016
Setting the stage
Who are we?

- Who is David Thompson?
- Who is Christopher Webber?
The web we want
The sad reality (centralization)
The sad reality (censorship)
The sad reality (surveillance)
The sad reality (fragility)
The sad reality (fragility)
The sad reality (fragility)
The sad reality (fragility)
4) Bring MediaGoblin to the people!
Stuff is complex to run

""""
" (_o (_o
G , ) )
__/   (uuuu)
/\   '-----'
'__',
\ \ '___',
' _3| [] | )3
"""

I AM GONNA BE HONEST WITH YA

I AM AFEARED TO UPGRADE MY OWN
SOFTWARES
Dependent on phase of the moon
One Language Package Manager Per Child

---

**INSTALL.SH**

```bash
#!/bin/bash

pip install "$1" &
easy_install "$1" &
brew install "$1" &
npm install "$1" &
yum install "$1" & dnf install "$1" &
docker run "$1" &
pkg install "$1" &
apt-get install "$1" &
sudo apt-get install "$1" &
steamcmd +app_update "$1" validate &
git clone https://github.com/"$1"/"$1" &
cd "$1"; ./configure; make; make install &
curl "$1" | bash &
```
Have fun managing configuration
So... docker?? (Or something like it?)

Easy for users! “I already built this for you, just pull it down and use it!”
Maybe not :(
Distro-sized static compiling considered hazardous

• Extremely heavy: throws away dynamic linking
• Hard to introspect, rebuild
• Analysis of Docker Hub: over 70% have medium vulnerabilities, 30-40% high (shellshock, heartbleed) vulnerabilities
  
• Reproducible? Not in the sense of
  https://reproducible-builds.org/
• Docker’s DSL is not expressive
• Still dependent on “phase of the moon” of distributions!

1http://www.banyanops.com/blog/analyzing-docker-hub/
Unfortunately, it’s not just Docker

- CoreOS
- xdg-app
- Qubes
- Snappy
- egads!

We are losing the ability to reason about free software!

A policy issue disguised as a technical issue?

If it’s too hard to build, run and modify software, what does this mean for user freedom?
And so here we are
Enter Guix & GuixSD!
Enter Guix & GuixSD!
Functional package management

- Precisely specified dependencies
- Reproducible builds (same input, same output)
- Built packages are immutable
- Unprivileged package management
- Atomic upgrades and roll backs
Packages capture the **full** dependency graph
Your profile, my profile

- Unprivileged package management (no root privileges needed!)
- Each user may install software *without conflicting* with other users or the system
- Each user may have *many* profiles

Guix gives users *practical freedom* to use different sets of packages for different tasks.
Transactional upgrades and rollbacks

Bad upgrade? No problem!

guix package --roll-back

Congratulations, you are now a time wizard!
Like Git, for your operating system!

cwebber@oolong:$ ls -l ~/.guix-profile/bin
total 1612
lrwxrwxrwx 3 root guixbuild 65 Dec 31 1969 2to3 -> /gnu/store/xw8ikmsj7b62aimwyd9kxwvygxs78hl-python-3.4.3/bin/2to3
lrwxrwxrwx 3 root guixbuild 69 Dec 31 1969 2to3-3.4 -> /gnu/store/xw8ikmsj7b62aimwyd9kxwvygxs78hl-python-3.4.3/bin/2to3-3.4
lrwxrwxrwx 16 root guixbuild 66 Dec 31 1969 abbaye -> /gnu/store/9qr1p42rsjp02cpk7q7qa4zg5pi-abbaye-1.13/bin/abbyae
lrwxrwxrwx 3 root guixbuild 69 Dec 31 1969 aclocal -> /gnu/store/rr3wsxa5q3hkwv9q8kmml14clrv7rdk-automake-1.15/bin/aclocal
lrwxrwxrwx 3 root guixbuild 74 Dec 31 1969 aclocal-1.15 -> /gnu/store/rr3wsxa5q3hkwv9q8kmml14clrv7rdk-automake-1.15/bin/aclocal1.15
lrwxrwxrwx 22 root 1001 73 Dec 31 1969 aconnect -> /gnu/store/ljqsldmm16j0hbw12mjhkyl7zb0ip75-alsa-utils-1.1.0/bin/aconnect
lrwxrwxrwx 3 root guixbuild 71 Dec 31 1969 ayclic -> /gnu/store/4g4gq0dnv1cap0fa0idf248ys7cx2mv-graphviz-2.38.0/bin/acyclik
lrwxrwxrwx 22 root 1001 77 Dec 31 1969 addr2line -> /gnu/store/7m8s5qm2xyz30lwzxhaccqh7i4kpy8i-gcc-toolchain-5.3.0/bin/addr2line
lrwxrwxrwx 22 root 1001 73 Dec 31 1969 alsaloop -> /gnu/store/ljqsldmm16j0hbw12mjhkyl7zb0ip75-alsa-utils-1.1.0/bin/alsaloop
lrwxrwxrwx 22 root 1001 74 Dec 31 1969 alsamixer -> /gnu/store/ljqsldmm16j0hbw12mjhkyl7zb0ip75-alsa-utils-1.1.0/bin/alsamixer

cwebber@oolong:$ ls /gnu/store | head
0001mfr72xdjw284dmldw067zylyf2p0-gradp-2.21.drv
004fhpfsnncl119v7zaa4p7rj31b7f-redland-1.0.17-guile-builder
0267biv0d5f8g6h6scnj8bzx4yn567d-other.drv
0211ka4a8v87d0ikrn543b10wd6a7z-guile-static-stripped-2.0.11.drv
030hbbab317r4kqslb459h3js157fki-libspectre-0.2.7.drv
044b0h788s3bcjm26q7szvk1k8qsqzd-git-manpages-2.6.3.tar.xz.drv
04pk19mwi54mvfrid8363pmh2g1vbz-unzip-6.0.drv
05sz52jrc8yrr8z205gpc5m313qppq-gif-2.7.10.drv
099003g12bt1pny50a8cg9iyl9bbniz-libxshmfence-1.1.tar.bz2.drv
0cc6nc4n4a66j19k04ngvf9xwy6vmrmn-shared-mime-info-1.2.tar.xz.drv

cwebber@oolong:$ 
Keep the history until you don’t need it

<table>
<thead>
<tr>
<th>N.</th>
<th>Current</th>
<th>Time</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>212</td>
<td>2016-03-03 17:12:31</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-212-link</td>
</tr>
<tr>
<td></td>
<td>211</td>
<td>2016-03-03 13:49:49</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-211-link</td>
</tr>
<tr>
<td>*</td>
<td>210</td>
<td>2016-03-03 10:16:30</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-210-link</td>
</tr>
<tr>
<td></td>
<td>209</td>
<td>2016-03-02 08:23:37</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-209-link</td>
</tr>
<tr>
<td></td>
<td>208</td>
<td>2016-03-02 07:40:41</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-208-link</td>
</tr>
<tr>
<td></td>
<td>207</td>
<td>2016-03-02 07:35:21</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-207-link</td>
</tr>
<tr>
<td></td>
<td>206</td>
<td>2016-03-02 07:35:02</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-206-link</td>
</tr>
<tr>
<td></td>
<td>205</td>
<td>2016-03-02 07:34:24</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-205-link</td>
</tr>
<tr>
<td></td>
<td>204</td>
<td>2016-03-02 07:33:51</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-204-link</td>
</tr>
<tr>
<td></td>
<td>203</td>
<td>2016-03-01 19:18:01</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-203-link</td>
</tr>
<tr>
<td></td>
<td>202</td>
<td>2016-03-01 15:03:39</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-202-link</td>
</tr>
<tr>
<td></td>
<td>201</td>
<td>2016-03-01 14:27:47</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-201-link</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>2016-03-01 12:25:28</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-200-link</td>
</tr>
<tr>
<td></td>
<td>199</td>
<td>2016-03-01 12:23:29</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-199-link</td>
</tr>
<tr>
<td></td>
<td>198</td>
<td>2016-03-01 12:03:22</td>
<td>/var/guix/profiles/per-user/cwebber/guix-profile-198-link</td>
</tr>
</tbody>
</table>

```diff
diff -u --label \#/buffer \*Guix\ guix-profile:: \generation\ 210\&\> --label \#/\buffer \*Guix\ guix-profile\\
\generation\ 212\&\> /tmp/buffer-content-664-e1 /tmp/buffer-content-664woE
--- #\buffer \*Guix\ guix-profile: \generation\ 210\&
+++ #\buffer \*Guix\ guix-profile: \generation\ 212\&
@@ -1,40 +1,40 @@
+abbaye-1.13:out       /gnu/store/9qr1p42rsjpz0cphk7q7za4915g5pibi-abbaye-1.13
+alsa-utils-1.1.0:out   /gnu/store/1lgqsldmm16j0hbwl12mhky1z7zb0ip75-alsa-utils-1.1.0
+aspell-0.60.6.1:out    /gnu/store/d6y54503s9sagx0cssyf5j92gw9x-0:aspell-0.60.6.1
+aspell-0.60.6.1:out    /gnu/store/1dq0k9sk83479x1259nq18p5iikk48cg6-aspell-0.60.6.1
+aspell-dict-en-2016.01.19-0:out /gnu/store/c1i352syqqxwqlpqw5l6i1q3jha\zsm-y-aspell-dict-en-2016.01.19-0
+asword-0.8:out         /gnu/store/4h14sh9myqggx2kv5lijqh4p1lw6s-asword-0.8
+autoconf-2.69:out       /gnu/store/bj5924kpe2jwmw8nsc53d2ip9dsw7kp-autoconf-2.69
+autokrwake-1.15:out     /gnu/store/3cq8rzbynpfbvz8wpw98vmd7yi\hc-automake-1.15
+asword-0.8:out         /gnu/store/dkvv6zdj3zd24vzq4g8d0rlayvrgpzd5y-asword-0.8
+autoconf-2.69:out       /gnu/store/mbr57wq24d1r1qkzsa5szyszss2a7b5q-4-autoconf-2.69
+autokrwake-1.15:out     /gnu/store/rrw9zxda5q53hkw9q8kmm614c1rv4rd-automake-1.15
```
The guix environment tool can be used to quickly create development environments.

Like Python’s virtualenv, but for anything.
$ which irb
which: no irb in (/run/current-system/profile/bin)
$ guix environment --ad-hoc ruby ruby-nokogiri
$ which irb
/gnu/store/q2ldaivsnfdmvlxnc7hlw5skc9f9xw5g-profile/bin/irb
$ irb
irb(main):001:0> require 'nokogiri'
=> true
irb(main):002:0> Nokogiri
=> Nokogiri
Guix is written in Scheme

```scheme
(define-public grep
  (package
    (name "grep")
    (version "2.21")
    (source
      (origin
        (method url-fetch)
        (uri (string-append "mirror://gnu/grep/grep-")
          version ".tar.xz")
        (sha256
          (base32
            "1pp5n15qwrxrlpibwjjhgsibyyv5cafhamf8lzjygs6y00fa2i2j")
          (patches (list (search-patch "grep-CVE-2015-1345.patch"))))
      (build-system gnu-build-system)
      (synopsis "Print lines matching a pattern")
      (description
        "grep is a tool for finding text inside files. Text is found by
         matching a pattern provided by the user in one or many files. The pattern
         may be provided as a basic or extended regular expression, or as fixed
         strings. By default, the matching text is simply printed to the screen,
         however the output can be greatly customized to include, for example, line
         numbers. GNU grep offers many extensions over the standard utility,
         including, for example, recursive directory searching.")
      (license gpl3+)
      (home-page "http://www.gnu.org/software/grep/")))
```

[Ins] grep.scm (26, 0) [Scheme] Git-master
All data structures, procedures, etc. are exposed as Guile APIs.

Using these APIs, we’ve implemented:

- Declarative config management (Puppet, Chef, Salt, Ansible, etc.)
- Universal language packaging (PyPI, RubyGems, ELPA, CRAN, etc.)
- Local dev environments (virtualenv, rvm, rbenv, nvm, etc.)
- Local VM creation (Vagrant)
- Linux containers (Docker, rkt, lxc, etc.)

The world is yours to hack!
Full system configuration management

```
(operating-system
 (host-name "izanagi")
 (timezone "America/New_York")
 (locale "en_US.UTF-8")
 (bootloader (grub-configuration (device "/dev/sda")))
 (file-systems (cons (file-system
                         (device "root")
                         (title 'label)
                         (mount-point "/")
                         (type "ext4"))
                           %base-file-systems))
 (users (list (user-account
                (name "dave")
                (comment "David Thompson")
                (group "users")
                (supplementary-groups '("wheel" "netdev" "audio"
                                      "video" "cdrom")
                (home-directory "/home/dave")))
 (packages (cons* adwaita-icon-theme avahi gnome-terminal
                  htop less man-db ncurses nss-certs
                  openssh wicd unzip rsync xfce
                  %base-packages))
 (services %desktop-services)
 (name-service-switch %mdns-host-lookup-nss))
```

U:@--- izanagi.scm  Bot (52,0)  Git:master (Scheme Guix Guile/A Paredit Proj
$ guix import pypi pyglet
(package
  (name "python-pyglet")
  (version "1.2.4")
  (source
    (origin
      (method url-fetch)
      (uri (pypi-uri "pyglet" version))
      (sha256
        (base32
          "0i9la03pm51swv2z8f17bx7qz2yjfxg6hn7i9c42s81bryxzyqlz")))
    (build-system python-build-system)
    (inputs
      `(('"python-setuptools" ,python-setuptools)))
    (home-page
      "http://pyglet.readthedocs.org/en/pyglet-1.2-maintenance/"
    (synopsis
      "Cross-platform windowing and multimedia library"
    (description
      "Cross-platform windowing and multimedia library"
    (license bsd-3))

Package importers
What’s in (/gnu/)store?
guix deploy

- Remote cluster management tool
- Provision bare metal, VMs, containers
- Complete replacement for Chef, Puppet, Ansible, Salt, etc.
Recipe based deployments? With a GUI?

- Select pre-built recipes for MediaGoblin, Wordpress, etc.
- “Don’t repeat yourself“ configuration
- Could we make a simple web interface for this?
Sandboxed applications

- Why should Solitaire have access to your GPG/SSH keys?
- Limit web browser’s access to your home directory
Easy installs?

- Graphical distro installer
- Graphical interface for:
  - system configuration
  - user-specific packaging
  - easy roll-backs
NodeJS, we’re looking at you! ²

Also, Java. ³

Send help.

²http://dustycloud.org/blog/javascript-packaging-dystopia/
Wrapping up
State of Guix

- “It’s still beta!”
- But probably more stable than most devops stuff
- A delight to run (we use it!)
- Easy to develop and get involved in
- \( \approx 3350 \) packages
- \( \approx 20-30 \) contributors per month and increasing
There’s more to do (and you can help!)

- Test drive Guix or GuixSD and send feedback
- Write GuixSD system services
- Improve documentation
- Improve containers, add packages, add importers, ... 

Talk with the developers via IRC at #guix on Freenode or via email at guix-devel@gnu.org or help-guix@gnu.org
A short story

• Tar pit image by Ray Bouknight, CC BY 2.0: https://www.flickr.com/photos/raybouk/8341369957

• Rena runs aground (container fail image), CC BY 2.0 https://www.flickr.com/photos/nzdefenceforce/6386334175/
• Caminandes video screenshot by Blender Institute, CC BY 4.0
  http://www.caminandes.com/

• Chemical warehouse image from Pixabay, CC0 https://pixabay.com/en/warehouse-chemistry-industry-629641/

• GuixSD logos by Luis Felipe López Acevedo, CC BY-SA 4.0
  http://www.gnu.org/software/guix/graphics/

• Docker + Twitter image by Karen Rustad

• Slight snippet from Guix (grep package), GPLv3 or later
Thanks! Questions?

© 2016 Christopher Allan Webber <cwebber@dustycloud.org>
© 2016 David Thompson <davet@gnu.org>

This presentation is licensed under the Creative Commons Attribution Share-Alike 4.0 International license.

More on Guix: https://gnu.org/software/guix