Reproducible Software Deployment in Scientific Computing
Ricardo Wurmus

sysadmin at MDC since 2014

Guix contributor since 2014

writes Lisp for fun

Emacs or vim? Yes.
Reproducible Software Deployment in Scientific Computing
Goals of Reproducibility

1. Dispel doubts
2. Facilitate further experimentation
[T]here is no point in repeating a computation identically. The results will be the same. So the only reason to re-run a computation is *when there are doubts about the exact software* or data that were used […].

The point of computational reproducibility is to *dispel those doubts*. The holy grail of computational reproducibility is not a world in which every computation is run five times, but a world in which a straightforward and cheap analysis of the published material verifies that it is reproducible, so that there is *no need to run it again*.

https://blog.khinsen.net/posts/2017/01/24/reproducibility-does-not-imply-reproduction
Goals of Reproducibility

1. Dispel doubts
2. Facilitate further experimentation
To repeat an experiment we first need to reproduce its environment.
How hard could this possibly be?
Very.
Conda?
Conda?

#4508 Dependency conflicts — glibc, libgcc, etc

#483 libstdc++.so.6: version `GLIBCXX_3.4.20' not...

#8415 R : libstdc++.so.6: version `CXXABI_1.3.9' not found

#6821 Non-deterministic behavior in solving for env...

Me, 6 months ago: I am going to save this conda environment with all the versions of all the packages so it can be recreated later; this is Reproducible Science!

conda, today: these versions don’t work together, lol.
Containers to the rescue?
Namespaces

pid  process table
net  networking
ipc  inter process comms
mnt  file system mount points
uts  unix time sharing
user user ids

Containers are not magic
Containers lack transparency

strawberry?

whale oil?
FROM amd64/debian:stretch
RUN apt-get update & & \\   apt-get install git make \   curl gcc g++ ...
RUN curl -L -O https:// ... & & \   ... & & \   make -j 4 & & ...
RUN git clone https:// ... & & \   ... & & \   make ... /usr/local/lib/ ...
Did we take a wrong turn?

C.H. Pander & E. d'Alton
“Die vergleichende Osteologie” (1821)
https://doi.org/10.5962/bhl.title.61021
The Purely Functional Software Deployment Model

$f(x, y...)$
headers
sources
build tools
libraries
...

\( f(x, y...) \)
headers
sources
build tools
libraries...

\[ f(x, y...) \]

cabba9e-samtools-1.7/
  └── bin
      └── samtools
  └── lib
      └── ...

...
Reproducible deployment of 28,000+ packages
Captures all inputs, including configuration space
Builds software in isolation
package
environments
containers
systems
package

environments

containers

systems
terribly bored?
check out the refcard:
guix shell

$ which octave
which: no octave ...
$ guix shell octave
[env] $ which octave
/gnu/store/ ... -profile/bin/octave
[env] $ exit
$ which octave
which: no octave ...
guix shell -m manifest.scm

(specifications→manifest
 (list "r-minimal"
  "r-genomation"
  "python"
  "python-pytorch"
  "samtools@0.1"))
guix time-machine

$ guix time-machine -C channels.scm -- \ 
  shell -m manifest.scm

Snapshot of all software known to Guix

Your selection
guix describe

Generation 38 Sep 18 2023 13:01:07 (current)

nonguix bb184bd
  repository URL: https://gitlab.com/nonguix/nonguix
  branch: master
  commit: bb184bd0a8f91beec3a00718759e96c7828853de

guix 2737099
  repository URL: https://git.savannah.gnu.org/git/guix.git
  branch: master
  commit: 273709932fa47f6ae56c96564a9415fdbff4a169
guix pack --format docker
python python-numpy

/gnu/store/
...-docker-pack.tar.gz
1. Reproducibility constrains variables
2. Version numbers don't mean much
3. Containers are outputs, not inputs
4. Guix captures the full dependency graph
5. Sysadmins hold the keys to practical reproducibility
Learn more

https://hpc.guix.info
https://guix.gnu.org

Let’s talk!

#guix on libera.chat
ricardo.wurmus@mdc-berlin.de