



Image: <u>CERN-EX-66954B</u> © 1998-2018 CERN

### Packaging for Python and Beyond

Chris Burr PyHEP 2019  $\circ$  16th October 2019





MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.







#### > PyPl is the main repository of Python packages Almost 200,000 packages ► Almost 1.5 million releases

Anyone can claim a name and upload without any review

### Python Package Index









- pip is the standard way to install packages
- Should be as simple as:

#### pip install PACKAGE\_NAME





## pip is the standard way to install packages Should be as simple as:

► You should not:

sudo pip install PACKAGE\_NAME



5

#### pip install --user PACKAGE\_NAME



#### pip install PACKAGE\_NAME







# Standard library module to create environments in isolated folders python -m venv /path/to/new/virtual/environment source /path/to/new/virtual/environment/bin/activate No special privileges required Easy to manage many independent environments



#### pip install xrootd

Collecting vrootd

Downloading https://files.pythonhosted.org/packages/50/77/c6ab02f4580c8ac246cced118053faf0262e230263583787beft
Building wheels for collected packages: xrootd
Building wheel for xrootd (setup.py) error
ERROR: Command errored out with exit status 1:
<pre>command: /Users/christopherburr/miniconda3/envs/test-split-root6/bin/python -u -c 'import sys, setuptools, to file='"'''/private/var/folders/m4/ny3rx96n7j97182lb50mh8mc0000gn/T/pip-install-q63znd5u/xrootd/setup.py'"'' le(code,file, '"''''exec'"'''))' bdist_wheel -d /private/var/folders/m4/ny3rx96n7j97182lb50mh8mc0000gn/T/pip cwd: /private/var/folders/m4/ny3rx96n7j97182lb50mh8mc0000gn/T/pip-install-q63znd5u/xrootd/ complete output (11 lines):</pre>
TTT No git repository info found Trying to interpret VERSTON INFO
[1] VERSION INFO file invalid Unable to determine the version Using "unknown"
/Users/christonherburr/miniconda3/envs/test-snlit-root6/lib/nython3 7/site-nackages/setuntools/dist nv·474· Users/christonherburr/miniconda3/envs/test-snlit-root6/lib/nython3 7/site-nackages/setuntools/dist nv·474· Users/christonherburr/miniconda3/envs/test-snlit-root6/lib/nython3 7/site-nackages/setuntools/dist nv·474· Users/christonherburr/miniconda3/envs/test-snlit-root6/lib/nython3 7/site-nackages/setuntools/dist nv·474· Users/christonherburr/miniconda3/envs/test-snlit-root6/lib/nython3 7/site-nackages/setuntools/dist nv·474·
' to '4 11 0'
normalized version.
usage: setup.pv [alobal opts] cmd1 [cmd1 opts] [cmd2 [cmd2 opts]]
or: setup.pyhelp [cmd1 cmd2]
or: setup.pyhelp-commands
or: setup.py cmdhelp
error: optionpython-tag not recognized
ERROR: Failed building wheel for xrootd
Running setup.py clean for xrootd
Failed to build xrootd
Installing collected packages: xrootd
Running setup.py install for xrootd error
ERROR: Command errored out with exit status 1:
<pre>command: /Users/christopherburr/miniconda3/envs/test-split-root6/bin/python -u -c 'import sys, setuptools, ';file='"'''/private/var/folders/m4/ny3rx96n7j97182lb50mh8mc0000gn/T/pip-install-q63znd5u/xrootd/setup.py'' pile(code,file, '"''''exec'"'''))' installrecord /private/var/folders/m4/ny3rx96n7j97182lb50mh8mc0000gn/T/pip-install-q63znd5u/xrootd/</pre>
[I] No git repository info found. Trying to interpret VERSION_INFO
[!] VERSION_INFO file invalid. Unable to determine the version. Using "unknown"
/Users/christopherburr/miniconda3/envs/test-split-root6/lib/python3.7/site-packages/setuptools/dist.py:474:
' to '4.11.0'
normalized_version,
running install
error: pkg-config probably not installed: FileNotFoundError(2, "No such file or directory: 'pkg-config'")
ERROR: Command errored out with exit status 1: /Users/christopherburr/miniconda3/envs/test-split-root6/bin/pytho tall=a63zpd5u/xrootd/setup py/"/"/: file =/"/"/ (private/var/folders/m4/py3px96p7j971821b50mb8mc0000cp/T/pip-

## Why virtual environments aren't enough

f47c213e6/xrootd-4.11.0.tar.gz (2.5MB)

tokenize; sys.argv[0] = '"'"'/private/var/folders/m4/ny3rx96n7j97182lb50mh8mc0000gn/T/pip-install-q63znd5u/xrootd/setup.py'"'"'; '"';f=getattr(tokenize, '"'"'open'"'", open)(\_\_file\_\_);code=f.read().replace('"'"'\r\n'"'"', '"'"'\n'"'"');f.close();exec(compi ip-wheel-klnoemiw --python-tag cp37

serWarning: Normalizing '4.11.0

, tokenize; sys.argv[0] = '"'"'/private/var/folders/m4/ny3rx96n7j97182lb50mh8mc0000gn/T/pip-install-q63znd5u/xrootd/setup.py'"'" '"'"';f=getattr(tokenize, '"'"'open'"'"', open)(\_\_file\_\_);code=f.read().replace('"'''\r\n'"'"', '"'"'\n'"'"');f.close();exec(com /T/pip-record-3iyw\_ie8/install-record.txt --single-version-externally-managed --compile

UserWarning: Normalizing '4.11.0

on -u -c 'import sys, setuptools, tokenize; sys.argv[0] = '"'"'/private/var/folders/m4/ny3rx96n7j97182lb50mh8mc0000gn/T/pip-ins install-q63znd5u/xrootd/setup.py'"'"';f=getattr(tokenize, '"'"'open'"'", open)(\_\_file\_\_);code=f.read().replace('"'"\r\n'"'"', 4/ny3rx96n7j97182lb50mh8mc0000gn/T/pip-record-3iyw\_ie8/install-record.txt --single-version-externally-managed --compile Check t







#### pip install xgboost

Collecting xgboost

Downloading https://files.pythonhosted.org/packages/96/84/4e2cae6247f397f83d8adc5c2a2a0c5d7d790a14a4c7400ff6574586f589/xgboost-0.90.tar.gz (676kB) 686kB 2.2MB/s

iniconda3/envs/my-root-6.18.02/bin/cla /Developer/CommandLineTools/usr/bin/make] - checked 0 DDMLC\_LOG\_CUSTOMIZE=1 -std=c++11 -Wall -Wno-unknown-pragmas -Iinclude -Idmlc-core/include -O3 -funroll-loops -msse2 -fPIC -fopenmp -MM -MT build/c\_api/c\_api\_error.o src/c\_api/c\_api\_error.cc >bu

#### 

## Why virtual environments aren't enough





Dependent on having a suitable existing Python installation Dependent on the upstream maintainers having good packaging Incomplete: uses parts of the host system, especially its compilers

# Why virtual environments aren't enough





Dependent on having a suitable existing Python installation Dependent on the upstream maintainers having good packaging Incomplete: uses parts of the host system, especially its compilers Limited to PyPI packages (though this is evolving: QT and CMake are now available)

# Why virtual environments aren't enough





- Language agnostic package manager (Python, C++, R, Julia, Rust, Go, Java, Ruby, Fortran, ...)
- Multi platform (Linux, macOS, Windows)
- Multi architecture (i386, x86\_84, aarch64, ppc64le)
- Usage is similar to venv
- conda activate my-environment

Anaconda, Inc. provides around 2,000 packages

#### conda create -- name my-environment python=3.7 ipython numpy jupyterlab





Language Multi plat ► Multi arch Usage is s conda create

### jupyterlab conda activa PLEASES R. M.A.N.

Anaconda, Inc. provides around 2,000 packages



..)







# Community maintained collection of conda packages Over 1,600 members

### Over 8,000 packages available and rapidly growing



Anaconda Repository Curated by Anaconda

Anaconda Cloud Uploaded by users & organizations

Anaconda Enterprise Curated by your organization

conda-forge Curated by the community











Community maintained collection of conda packages ► Over 1,600 members

Over 8,000 packages available and rapidly growing





### CONDA® packages

Anaconda Repository Curated by Anaconda

Anaconda Cloud Uploaded by users & organizations

Anaconda Enterprise Curated by your organization

conda-forge Curated by the community











Language agnostic! Provides it's own compiler toolchain with full C++17 support (Clang 9/GCC 7.3) "Hard" packages are already present and maintained (build systems, Boost, X11)

It's not just analysis tools Install bash, htop, vim, singularity

Effectively no dependencies on the host system MacOS 10.9+/CentOS 6+/Windows 64bit

Open community

Missing package: You can add it

Broken/hard to install package: You can fix it (and people are generally happy to help)



- Reliably install ROOT in under 5 minutes on any machine
   Linux, macOS, and Windows Subsystem for Linux (cling doesn't support native 64-bit Windows)
- ► Complete installation with C++17, graphics, OpenGL
- Seamlessly integrates with the rest of conda-forge
   No builtin dependencies (excluding cling for now)
   No PYTHONPATH/LD\_LIBRARY\_PATH mess
- Downloaded over 50,000 times since it was released 9 months ago
- Currently working with the ROOT team to integrate with their nighties
   Plan to release the binaries to a dedicated conda channel
   If you've be interested in using these, please let me know!

Data Analysis Framework





# How does conda-forge work?





```
Raw Blame History 🖵 🖍 🕅
61 lines (53 sloc) 1.21 KB
     {% set name = "zfit" %}
      {% set version = "0.3.6" %}
      package:
        name: "{{ name|lower }}"
        version: "{{ version }}"
      source:
        url: https://pypi.io/packages/source/{{ name[0] }}/{{ name }}-{{ version }}.tar.gz
        sha256: 26e76eb100c95ed52241f3b552d7dd16f59091a83f5e01b263f6fa9f12b30cfe
 10
 11
      build:
 12
 13
        number: 0
        script: "{{ PYTHON }} -m pip install . -vv "
 14
        noarch: python
 15
 16
 17
      requirements:
        host:
 19
          - pip
 20
          - python >=3.6
          - setuptools_scm
 21
 22
          - setuptools_scm_git_archive
 23
        run:
 24
          - python >=3.6
 25
          - tensorflow-base >=1.14.0
 26
          - tensorflow-probability >=0.6.0
 27
          - scipy >=1.2
 28

    uproot

 29

    pandas

 30
          - numpy
 31
          – iminuit
 32
          - typing
 33
          - colorlog
 34

    texttable

 35
          # Workaround for https://github.com/conda-forge/tensorflow-probability-feedstock/pull/11

    decorator

          - cloudpickle >=0.6.1
 37
 38
```

## Adding a new package to conda-forge

#### Create a pull request against https://github.com/conda-forge/staged-recipes

### Can be mostly automated using conda skeleton pypi zfit

```
39
    test:
40
       imports:
41
         - zfit
         - zfit.core
42
         – zfit.minimizers
43
44
         - zfit.models
         - zfit.util
45
46
         - zfit.ztf
47
48
     about:
49
       home: https://github.com/zfit/zfit
50
       license: BSD-3-Clause
51
       license_family: BSD
52
       license_file: LICENSE
53
       summary: Physics extension to zfit
       doc url: https://zfit.readthedocs.io/
54
55
      dev_url: https://github.com/zfit/zfit
56
57
     extra:
58
       recipe-maintainers:
59
         - chrisburr
60
         - mayou36
```







3 4 ■■■■ recipe/meta.yaml A						
		@@ -1,13 +1,13 @@				
1	1	{% set name = "phasespace" %}				
2		<pre>- {% set version = "1.0.3" %}</pre>				
	2	+ {% set version = "1.0.4" %}				
3	3					
4	4	package:				
5	5	name: "{{ name lower }}"				
ô	6	version: "{{ version }}"				
7	7					
8	8	source:				
9	9	url: https://github.com/zfit/phasespace/archive/{{ version }}.tar.gz				
0		- sha256: ad1b322c1c47378ec6687c1ec30d5b92101c883e96bbebea345526b21596516e				
	10	+ sha256: 18e709a27111f96276aaa1f0df073e4cefc5e764ed9551de24b345aa3cc88790				
1	11					
2	12	build:				
3	13	number: 0				
ΣĮ3	C.					





- Conda only supports installing binaries\*
- Relies on the solver knowing about API/ABI compatibility
- > Packages with shared libraries should specify what their ABI stability is
- Doesn't necessarily restrict what you can do
  - Variants can be used to provide different versions of a packages
  - > BLAS can be provided by netlib, mkl, blis and openblas
  - Several MPI variants
  - TensorFlow has CPU and (several) GPU variants

\*Some organisations mirror the conda-forge build infrastructure for their own internal use







> What about when ABIs change? More

- > A line is added to a git repository
- Pull requests appear that rebuild packages in the correct order

Current Migrations:









#### Migration is currently ongoing for ppc64le and aarch64 support ► ROOT is included as a target



#### Support for compiling CUDA with nvcc is rapidly maturing Adds three additional additional targets (different driver versions) Close to being fully supported by the conda-forge tooling ► GPU variants of packages are already being added

# **Alternative architectures**

aarch64 and ppc64le addition Migration Status

ıg-pr (1)	awaiting-parents (21	) bot-error (0)
B-PR	AWAITING-PARENTS	BOT-ERROR







- One repository per package ("feedstock")
- > All packages are built using well known CI providers
- A year ago this was Travis CI + Circle CI + Appveyor
- Now mostly Azure Pipelines
- > Native builds for alternative architectures: ppc64le with Travis CI
  - ► aarch64 with Drone CI
- All managed by an external package: conda-smithy Used to regenerate CI configuration for each update





### Long term reproducibly is important for science

### Easy to export exact builds of everything that was installed conda list --explicit --md5 > environment.txt

# This file may be used to create an environment using:

# \$ conda create --name <env> --file <this file>

# platform: osx-64

**@EXPLICIT** 

https://repo.anaconda.com/pkgs/main/osx-64/blas-1.0-openblas.conda#4ff605b9a6c88bbfd4428e6f9703d9ff https://conda.anaconda.org/conda-forge/osx-64/bzip2-1.0.8-h01d97ff\_1.tar.bz2#8397e58ec046d7961aca25e741fa175a https://conda.anaconda.org/conda-forge/osx-64/ca-certificates-2019.9.11-hecc5488\_0.tar.bz2#c2dd037b660ef2c2b7c5c5535ec1a09b https://conda.anaconda.org/conda-forge/noarch/conda-forge-pinning-2019.10.01-0.tar.bz2#49710c701b5be98e25bce22d150a0471 https://conda.anaconda.org/conda-forge/osx-64/fribidi-1.0.5-h01d97ff\_1002.tar.bz2#d5db22ff1a37c085f1f50999facb755c https://conda.anaconda.org/conda-forge/osx-64/giflib-5.1.7-h01d97ff\_1.tar.bz2#796a4523450c9ab1b495c5c47b10c859 https://conda.anaconda.org/conda-forge/osx-64/jpeg-9c-h1de35cc\_1001.tar.bz2#bcc9abfebf1cc26568e1ec4502834512 https://conda.anaconda.org/conda-forge/osx-64/libcxxabi-9.0.0-0.tar.bz2#39d20d41d2ecf4d0e8eb626c855b0bef https://conda.anaconda.org/conda-forge/osx-64/libgfortran-3.0.1-0.tar.bz2#d69b2c1cc8250395702768acf3a912e3 https://conda.anaconda.org/conda-forge/osx-64/libiconv-1.15-h01d97ff\_1005.tar.bz2#ec331659b7d7ec6565a54bf1ca65ac82 https://conda.anaconda.org/conda-forge/osx-64/libsodium-1.0.17-h01d97ff\_0.tar.bz2#3dd1aeaa242c631c14e4b3b144669fa7 https://conda.anaconda.org/conda-forge/osx-64/libuv-1.32.0-h01d97ff\_0.tar.bz2#fcbcb890e331687f5b118371d3073464 https://conda.anaconda.org/conda-forge/osx-64/llvm-openmp-9.0.0-h40edb58\_0.tar.bz2#7dc368e4f1c6bf92a6dd5e50e25942d9  $https://conda.anaconda.org/conda-forge/osx-64/lzo-2.10-h1de35cc_1000.tar.bz2\#9cc86637a9a6ecc9005376afc5b296e0$ https://conda.anaconda.org/conda-forge/osx-64/pandoc-2.7.3-0.tar.bz2#9d7c8563365d39806e683227bc6cfa25 $https://conda.anaconda.org/conda-forge/osx-64/perl-5.26.2-haec8ef5_1006.tar.bz2\#7c0ced1da3e20d98b1045f3e945d3d1c$ https://conda.anaconda.org/conda-forge/osx-64/pixman-0.38.0-h01d97ff\_1003.tar.bz2#db285954d969a73e3adc622d0db30605

#### conda create -- name my-new-env -- file environment.txt



- Bad metadata in old packages can "poison" the solver Mechanism exists for patching this but it's fiddly
- Current solution is to move packages to a "broken" channel Channel list must have "conda-forge/label/broken" appended ► URL also changes (for now)
- Lots of benefits to having a docker container "just in case" Currently setting up a mirroring proxy for conda within LHCb Also deploying to CVMFS > Ask me if you're interested



### Creates a relocatable self extracting archive of a conda environment

<pre>jcrist computer_one \$ source activate example (example) icrist computer one \$ # Package the current environment</pre>							
(example) jerist computer_one \$ # rackage the current environment							
(example) jcrist computer_one \$ conda-pack							
Collecting packages							
Packing environment at '/Users/jcrist/anaconda/envs/example' to 'example							
[#####################################							
<pre>(example) jcrist computer_one \$ ls</pre>							
example.tar.gz							
(example) jcrist computer_one \$ # The environment is packaged as a tar.g:							
<pre>(example) jcrist computer_one \$ # Get the file size</pre>							
(example) jcrist computer_one \$ du -h example.tar.gz icrist computer two \$ # Activate the	environment						
55M example.tar.gz	n/activate						

```
(myenv) jcrist computer_two $ # Use applications in the environment
(myenv) jcrist computer_two $ which ipython
/Users/jcrist/computer_two/myenv/bin/ipython
(myenv) jcrist computer_two $ ipython
Python 3.6.5 |Anaconda, Inc.| (default, Apr 26 2018, 08:42:37)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.4.0 -- An enhanced Interactive Python. Type '?' for help.
```

In [1]: import numpy







# For library maintainers...



of the supervised in



### Automate deployment ► Keep it simple if possible

### Make it easy to run tests against a pre-existing installation

If it's compiled: provide as many pre-build wheels as you can pip install should work for Python packages ► But also include source distributions!

### **General comments**





#### Use setuptools\_scm Removes all duplication of version numbers

setup( name='cirun', use\_scm\_version=True, setup\_requires=['setuptools\_scm'],

Uses version control to compute version numbers

Release: 1.0.0

### Even accounts for installs from dirty clones



- 9 commits later: 1.0.1.dev9+g59ff2f1



### > PyPI finally supports API tokens!

Your account

Sour projects

🄹 Account settings

christopher.burr@cern.ch  $\circ$  PyHEP 2019  $\circ$  Packaging for Python and Beyond

#### Add API token BETA

#### Token name

CI-testing

What is this token for?

#### Scope

Entire account (all projects)

Entire account (all projects)

Project: econ-ark

Project: Forms990-analysis

Project: mauritstestpackage-sumana-version

Project: twine

http://pyfound.blogspot.com/2019/07/pypi-now-supports-uploading-via-api.html







### First: Don't vendor or have builtin dependencies

christopher.burr@cern.ch  $\circ$  PyHEP 2019  $\circ$  Packaging for Python and Beyond







## First: Don't vendor or have builtin dependencies Second: Seriously...don't...







## First: Don't vendor or have builtin dependencies Second: Seriously...don't... Third: Make it easy to unvendor them







- First: Don't vendor or have builtin dependencies
- Second: Seriously...don't...
- Third: Make it easy to unvendor them
- Finally: At least make sure they're contained
  - ► Statically link
  - Ensure symbols are hidden
  - Don't put vendored files in standard locations (e.g. shared libraries in \$PREFIX/lib/)







### Create Python wheels from conda packages

\$ conda press --subdir osx-64 --skip-python --fatten iminuit=1.3.7=py37h86efe34\_0 created fat wheel: iminuit-1.3.7-0\_py37h86efe34-cp37-cp37m-macosx\_10\_9\_x86\_64.whl

► Why? Centralises the building of packages Easier to pull in dependencies Use newer ABIs than manylinux<n>

https://pydata.org/nyc2019/schedule/presentation/41/conda-press-or-reinventing-the-wheel/







# Creating packages





















christopher.burr@cern.ch  $\circ$  PyHEP 2019  $\circ$  Packaging for Python and Beyond

#### Tests



setup(



See <u>https://github.com/pypa/sampleproject</u> for a more complete example

### Package structure: setup.py

```
from setuptools import setup, find_packages
import os
from os.path import abspath, dirname, join
from io import open
# Get the long description from the README file
with open(join(abspath(dirname(__file__)), 'README.md'), encoding='utf-8') as f:
    long_description = f.read()
package_data = []
for root, dirs, files in os.walk(join(here, 'src/cirun/data')):
    package_data += [join(root, fn) for fn in files]
    name='cirun',
    use_scm_version=True,
    description='Create customised GitLab CI runners',
    long_description=long_description,
    long_description_content_type='text/markdown',
    url='https://gitlab.cern.ch/lhcb-gitlab-runners/cirun',
    author='LHCb',
    classifiers=[...],
    keywords='LHCb CERN GitLab CI testing openshift',
    packages=find_packages('src'),
    package_dir={'': 'src'},
    python_requires='>=2.7, !=3.0.*, !=3.1.*, !=3.2.*, !=3.3.*, !=3.4.*, !=3.5.*',
    setup_requires=['setuptools_scm'],
    install_requires=['celery', 'flask', 'gitlab_runner_api'],
    tests_require=['pytest'],
    package_data={
        'cirun': package_data,
    },
    zip_safe=False,
    project_urls={
        'Bug Reports': 'https://gitlab.cern.ch/lhcb-gitlab-runners/cirun/issues',
        'Source': 'https://gitlab.cern.ch/lhcb-gitlab-runners/cirun',
    },
```







```
from setuptools import setup, find_packages
import os
from os.path import abspath, dirname, join
from io import open
# Get the long description from the README file
with open(join(abspath(dirname(__file__)), 'README.md'), encoding='utf-8') as f:
    long_description = f.read()
```

```
description='Create customised GitLab CI runners',
long_description=long_description,
long_description_content_type='text/markdown',
```















setup( name='cirun',

See <u>https://github.com/pypa/sampleproject</u> for a more complete example

use\_scm\_version=True,

### No need to have a file with the version number

### Also versions unreleased code

Release: 1.0.0

9 commits later: 1.0.1.dev9+g59ff2f1











Needed to use the package









- "docs": docs\_require,
- "dev": docs\_require + tests\_require,

## Set up a development environment with pip install -e ".[dev]"







- It's controversial but I like source directories
- Helps isolate editable installs
- Less likely to result in "works for me"
- For a fuller justification see here: <u>https://blog.ionelmc.ro/2014/05/25/python-packaging/</u>