Harnessing the Power of Python in ArcGIS Using the Conda Distribution

Shaun Walbridge

https://github.com/scw/conda-uc-2016-demo

High Quality PDF (2MB)

Python

Why Python?

- Accessible for new-comers, and the most taught first language in US universites
- Extensive package collection (56 thousand on PyPI), broad user-base
- Strong glue language used to bind together many environments, both open source and commercial
- Open source with liberal license — do what you want

Why Python?

In the box:

- The SciPy Stack (NumPy, SciPy, Pandas, matplotlib, sympy)
  - Scientific Programming with the SciPy Stack
- xlrd, netCDF4, requests, PyPDF, pytz

Why Python?

Beyond the box:

- Integrating Open-Source Statistical Packages with ArcGIS using Python and R — Tomorrow at 10:15am, Ballroom 6D
- Python: Extending with Other Libraries — Today at 4:00pm, Tech Theater 16
- Deeper Dive into Conda in DevSummit Tech Session Video

Package Management for Python

Why not `pip`, wheels, virtualenvs?

- Don’t handle the harder problem of system dependencies, considered out of scope by Python packagers — does it end up in `site-packages`?
- Package devs: On OSX and Linux, ‘easy’ to get the deps! Use a system package manager (e.g. `apt`, `brew`, `yum`) and the included compiler (e.g. `clang`, `gcc`).
- It’s still not easy to make reproducible builds, and what about Windows?

What about Windows?

- We are particularly stuck on Windows which lacks broadly used package management
- Only developers have a C compiler on their machine
- A hard problem

What about Windows?

- We are particularly stuck on Windows which lacks broadly used package management
- Only developers have a C compiler on their machine
- A hard problem
    - Enter Conda

Why Conda?

- Scientific Python community identified that there was a gap not being addressed by the core Python infrastructure, limiting their ability to get packages into the hands of users
- Industry standard built by people who care about this space — Continuum Analytics

Why Conda?

- It solves a hard problem:
- Handles dependencies for many languages (C, C++, R and of course Python)
- Built for Python first, but it really solves a much broader infrastructural issue.

Conda

Conda

- Cross-platform: simply develop recipes for building and installing software on Linux, OS X and Windows.
- Open source: Esri is using it, you can use it in your own projects for other contexts

What can it install? Not just scientific packages. It can help with:

- GUI toolkits (PyQt, TKinter)
- C++ Libraries (Boost)
- IDEs (Spyder, Juptyer)

Conda

- Environments: Can isolate a Python environment, flexibly make changes withot affecting installed software.
- Requirements — include explicit state information, not just the package name. Names aren’t enough!
- Also handles platforms and Jupyter notebooks

How Does it Work?

Conda packages can come from a variety of locations:

- On disk (`file://`)
- Public repositories (Anaconda Cloud, self-hosted)
- Private repositories
- anaconda.org

Conda Basics

Command line interface, for now

- Conda Cheatsheet

Conda Basics Demo

Conda Basics

Activating environments, a couple ways:

- Use the shortcuts
- Manually activate the environment:

```
cd C:\ArcGIS\bin\Python\Scripts
activate arcgispro-py3
```

Conda Basics

To start:

```
conda --help
```

- A collection of packages and Python install is called an *environment* or *env*, the building block for managing Python with Conda
- Can have multiple environments and seamlessly switch between them

Conda Basics

Once you're in an environment get details with `info`:

```
conda info
```

Conda info is the starting point — it tells you the state of the environment.

Conda Basics

conda info

```
Current conda install:

             platform : win-64
        conda version : 4.0.6
  conda-build version : not installed
       python version : 3.5.1.final.0
     requests version : 2.9.1
     root environment : C:\ArcGIS\bin\Python  (writable)
  default environment : C:\ArcGIS\bin\Python\envs\arcgispro-py3
     envs directories : C:\ArcGIS\bin\Python\envs
        package cache : C:\ArcGIS\bin\Python\pkgs
         channel URLs : https://conda.anaconda.org/esri/win-64/
                        https://conda.anaconda.org/esri/noarch/
                        https://repo.continuum.io/pkgs/free/win-64/
                        https://repo.continuum.io/pkgs/free/noarch/
          config file : C:\ArcGIS\bin\Python\.condarc
```

Conda Basics

conda list

```
# packages in environment at C:\ArcGIS\bin\Python\envs\arcgispro-py3:
#
arcgispro                 1.3                      0    esri
colorama                  0.3.6                 py34_0    defaults
future                    0.15.2                py34_0    defaults
matplotlib                1.4.3              np19py34_0    defaults
msvc_runtime              1.0.1                 vc10_0  [vc10]  defaults
nose                      1.3.7                 py34_0    defaults
numpy                     1.9.3                 py34_0e  [arcgispro]  esri
openssl                   1.0.2h                vc10_0  [vc10]  defaults
pandas                    0.17.1             np19py34_0    esri
pip                       8.1.1                 py34_1    defaults
py                        1.4.31                py34_0    defaults
pyparsing                 2.1.1                 py34_0    defaults
pypdf2                    1.25.1                  py_0    esri
pytest                    2.9.1                 py34_0    defaults
python                    3.4.4                      4    defaults
python-dateutil           2.5.3                 py34_0    defaults
pytz                      2016.4                py34_0    defaults
```

Conda Basics

Creating new environments:

- A few different ways. Can manually specify the dependencies:

```
conda create --name my_env python=3.4 numpy flask dask
```

- Can also use a file which includes all the dependencies:

```
conda create --name my_env --file my_sweet_depends.txt
```

These can contain explcit information about channels, to ensure that the new environment precisely matches the requirements.

Conda vs…

Name	Means	Will Ship?
Conda	The command itself	✓
Miniconda	A minimum set of Python packages to build and run Conda.	✓
Anaconda	A distribution 200+ packages built with Conda	
Anaconda Server	Host the full infrastructure internally	

Skikit Learn Demo

- Have tweets about an iOS app released at #SXSW
- What to people think of it? Use a naive bayes classifier to determine sentiment

Skikit Learn Demo

```
# Scikit learn model based Lukas Biewald's Scikit Learn class
# https://github.com/lukas/scikit-class

import arcpy
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB

input_csv = arcpy.GetParameterAsText(0)
test_string = arcpy.GetParameterAsText(1)

df = pd.read_csv(input_csv)
target = df['is_there_an_emotion_directed_at_a_brand_or_product']
text = df['tweet_text']
```

Skikit Learn Demo

```
fixed_text = text[pd.notnull(text)]
fixed_target = target[pd.notnull(text)]

count_vect = CountVectorizer()
count_vect.fit(fixed_text)
counts = count_vect.transform(fixed_text)

nb = MultinomialNB()
nb.fit(counts, fixed_target)

# print out our prediction
arcpy.AddMessage(nb.predict(count_vect.transform([test_string][0])))
```

Deeper Dive

Multiple Pythons

Currently:

Platform	Python version
Desktop	Python 2.7.x (2.7.10)
Pro	Python 3.4.x (3.4.3)

Multiple Pythons

Upgrade code? Python migration for ArcGIS Pro

- Do it! You can support 2 + 3 without that much work
- Still need to change `arcpy.mapping` to `arcpy.mp` when moving from Desktop to Pro, but no Python language level changes needed. . . .

But… this can be costly. For many organizations, a significant burden, even if the language changes are relatively small. Multiple Pythons is a solution to this.

Challenges

Have to make sure you're running the right Python (*what happens when you type `python` at the command line?*)

- Working to make this easy as possible
- It'll be easy to tell in app
- Isolated installation fixes a variety of issues

Requires some user education over the "only one Python on the box" model

What Do I Get Out of the Box?

- Conda command and a Conda root Python install
- New modules (e.g. `requests`)
- Conda environment with all of the ArcGIS Pro dependencies as Conda packages

How can I use this?

- We already ship you the SciPy stack — powerful and out of the box, can use today (Pro and 10.4)
- Can start using conda today. Miniconda is fully stand-alone, won't affect your global Python (unless you tell it to)
- Package your work: this is an opportunity to distribute it, possibly including commercial side as well.

Where Can I Run This?

- ArcGIS Pro 1.3
  - Will be *the* Python install.
- Future:
  - UI for interaction
  - Take advantage of more features
  - Integration with platform

from future import *

Effectively manage complex software dependencies with Conda. Thousands of packages exist today, can integrate it into your organization's needs.

Closing

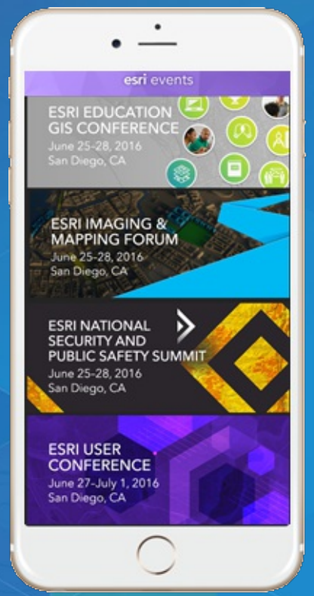
Thanks

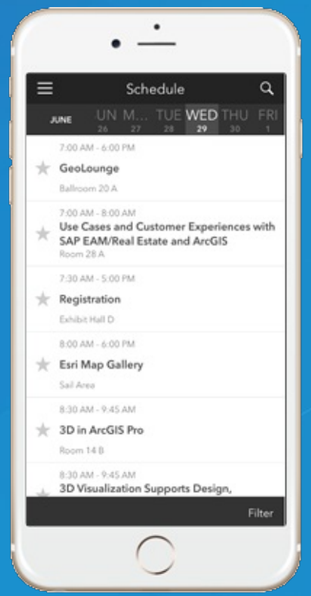
Esri Conda Team:

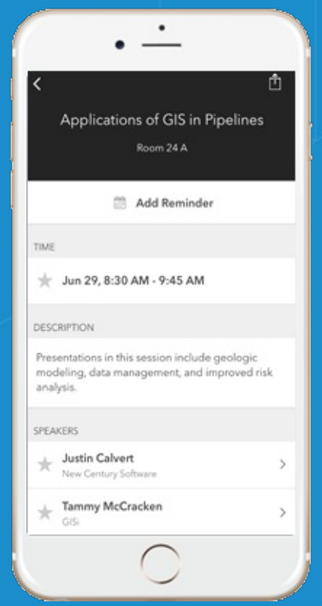
Continuum Analytics for creating and open sourcing Conda

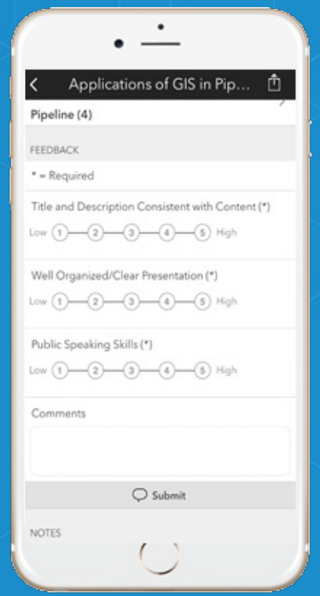
Rate This Session

Please take our survey, find session in app and provide review










esri®