Python 2 to Python 3 transition

G. Iadarola and L. Mether

Many thanks to P. Dijkstal, K. Li, and R. De Maria

Python 2 to Python 3

PyECLOUD, **PyHEADTAIL**, **PyPIC**, **PyPARIS**, etc… have been developed in **Python 2.7**

→ Python 3 and related libraries were not mature enough at that time

Python 2.7 **served us very well** in the last 8 years

→ It is being **retired on 31 Dec 2019** (end of official support)

The future is **Python 3**:

→ Solves several "youth limitations" of Python 2

→ It has **already several years of history**, latest releases are very solid

→ It is **already a standard** for all new application, also at CERN (see sixtracklib ecosystem)

Python 2 to Python 3

Python 2 and Python 3 are **not compatible with each other**

→ **Code changes are required** to migrate from py2 to py3

→ Python 2 and Python 3 **libraries cannot work together** (e.g. we could not have PyPARIS in py3 and PyEC in py2)

We need to migrate all the tools at the same time!

How do you get python 3

The simplest way is to install **miniconda3**:

- You can find a simple tutorial in the PyECLOUD wiki

  https://github.com/PyCOMPLETE/PyECLOUD/wiki/Install-miniconda

For the **clusters** (where we need to use MPI) it is more convenient to **compile python from the source code**

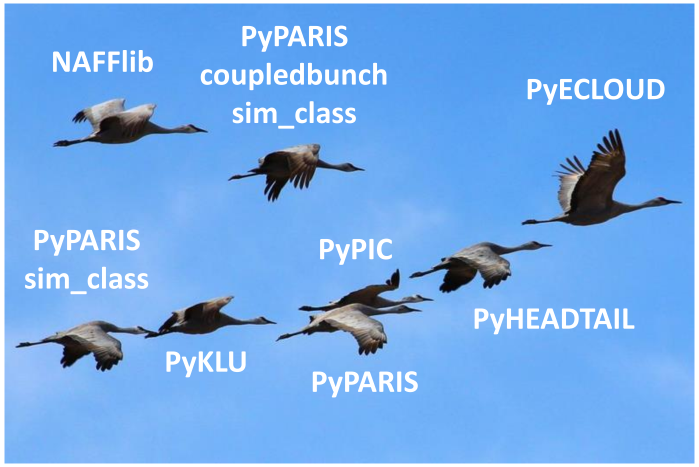
- The recipe is available in the PyECLOUD wiki:

  https://github.com/PyCOMPLETE/PyECLOUD/wiki/Setup-python-3-%28including-mpi4py%29-without-admin-rights

A public installation for the team is available in **lxplus**. Can be activated by:

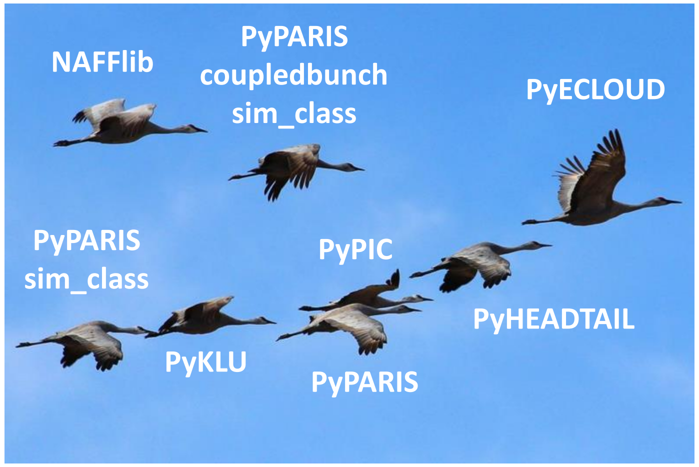
```
source /afs/cern.ch/work/e/ecloud/public/python38_for_lxplus/venvpy38/bin/activate
```

Migration of simulation codes



- **Python 3 versions** for the entire PyECLOUD-PyHEADTAIL ecosystem have been **prepared and tested** over the last months
  - Strategy: **modified in several places Python 2 code**, so that **Python 3** code could be **automatically generated** (using 2to3 tool)
  - Avoid developing two codes
- Same strategy will be applied next year to **machine follow-up tools**

Migration of simulation codes



- Python 3 versions have been **released this week** for all packages:
  - https://github.com/PyCOMPLETE/PyECLOUD/releases/tag/v8.2.0_python3
  - https://github.com/PyCOMPLETE/PyHEADTAIL/releases/tag/v1.13.5_python3
  - https://github.com/PyCOMPLETE/PyPIC/releases/tag/v2.4.4_python3
  - https://github.com/PyCOMPLETE/PyPARIS/releases/tag/v2.4.1_python3
  - https://github.com/PyCOMPLETE/PyPARIS_sim_class/releases/tag/v1.2.4_python3
  - https://github.com/PyCOMPLETE/PyPARIS_CoupledBunch_sim_class/releases/tag/v1.0.0_python3

- PyKLU and NAFFLIB are natively compatible with Python 3

Transition period (Dec 2019 – Jan 2020)

We will have a **transition period** (Dec 2019 – Jan 2020) :

- **Python 2.7** versions will **remains the default** (used for production studies)
- **Pilot studies with Python 3** will be performed on **different systems** to identify problems that escaped tests done so far

Special procedure to install python 3 versions during the **transition period** is available at:

- https://github.com/PyCOMPLETE/PyECLOUD/wiki/Python-3-test-period

At the **end of the transition period** (mid Jan 2020, if no surprises):

- **Python 3 versions** will **become the default**
- After, it will be possible to install python 3 versions **using the standard steps**:
  - https://github.com/PyCOMPLETE/PyECLOUD/wiki/How-to-install-PyECLOUD

HTCondor test

Tested **PyECLOUD buildup study** in Python 3 on **HTCondor**

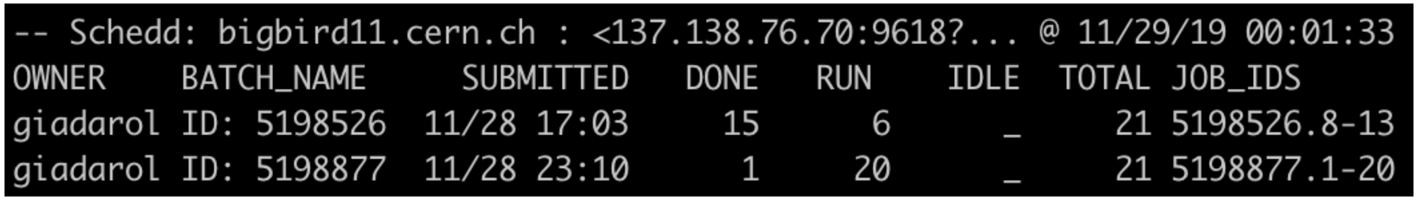
→ No surprises

Done in a **public space** so that you can have a look:

- Workspace folder:

  /afs/cern.ch/work/e/ecloud/public/example_sim_workspace_py3
- Study folder:

  /afs/cern.ch/work/e/ecloud/public/example_sim_workspace_py3/test_study



```
-- Schedd: bigbird11.cern.ch : <137.138.76.70:9618?... @ 11/29/19 00:01:33
OWNER     BATCH_NAME      SUBMITTED    DONE    RUN     IDLE  TOTAL JOB_IDS
giadarol ID: 5198526  11/28 17:03     15       6       _     21 5198526.8-13
giadarol ID: 5198877  11/28 23:10      1      20       _     21 5198877.1-20
```

Migrating your own code

We all have lots of code in the form of scripts, notebooks etc...

→ This will also **need to be migrated**

How to do it?

- Typically it is just **small things**
    - Python 3 forbids mixed (tab/space) indentations
        - You can check already in python 2 using “python -tt hello.py”
    - print ‘Hello’ → print(‘Hello’)
    - xrange → range
    - Division between integers (3/2=1 in python 2, 3/2=1.5 in python3)
    - Careful with strings:
        - Bad: `if a is ‘mystring’`
        - `Good: if a == ‘mystring’`
    - Relative imports work a bit differently

- You can use **a tool that attempts to do it automatically**
    - `pip install` **`2to3`**
    - `Usage: 2to3 -w example.py`
    - **Test!!!!**

And now…

- Make your **first simulation studies in python 3**
- **Migrate your setup and analysis scripts**
    - Communicate any issues to Lotta or Gianni