Giving *pandas* ROOT to chew on experiences with the XENON1T Dark Matter experiment

Daniela Remenska
eScience engineer
Beyond HEP-specific tools

• The XENON(1T) experiment
  • Relatively small (150 researchers), no dedicated software manpower
  • Signal processing & physics analysis software written in Python3
    - C++ is an overkill for the kind of data analytics they do daily
    - Low-level C/C++ libraries only for performance-critical tasks
  • Tension between HEP-specific software and generic “big data” solutions. Some XENON researchers prefer C++/ROOT.
  • Can't we just have both, please?
Interfacing HEP and the Big Data Ecosystems
Goals

- Understand the current capabilities and limitations of the Python(3)/ROOT ecosystem
- Install everything in user space, without compiling and all the dependency hell.

```bash
ln -s `python3.4-config --exec-prefix`/lib/libpython3.4m.dylib `python3.4-config --exec-prefix`/lib/libpython3.4.dylib
./configure --enable-python --with-python-incdir=`python3.4-config --exec-prefix`/include/python3.4m --with-python-libdir=`python3.4-config --exec-prefix`/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:`python3.4-config --exec-prefix`/lib
export PYTHONPATH=$LD_LIBRARY_PATH

conda install python=3.4 root=6 numpy matplotlib pandas rootpy jupyter
```
Interfacing ROOT to Python *pandas*

- **root_pandas:** ROOT I/O for Pandas
  - *pandas* should read ROOT files

- **rootpy:** Pythonic ROOT
  - Truly “Pythonic” ROOT interface

- **PyROOT:** A Python -- ROOT Bridge
  - Python bindings for ROOT

- **ROOT:**
  - C++
from rootpy.tree import Tree, TreeModel
from rootpy.tree import FloatCol, IntCol
from rootpy.tree import FloatArrayCol, CharCol
from rootpy.io import root_open
from random import gauss, choice

define the model
class Event(TreeModel):
s = CharCol()
x = FloatCol(default='nan')
y = IntCol(default=0)
f = FloatArrayCol(5)
tree = Tree("test", model=Event)

fill the tree
for i in range(5):
    tree.s = ord(choice(ascii_letters))
    tree.x = gauss(.5, 1.)
    tree.y = i
    for j in range(5):
        tree.f[j] = gauss(-2, 5)
    tree.fill()
tree.write()
f.close()

Listing 1: Creating tree models with rootpy

class ReconstructedPosition(object):
x = float('nan')
y = float('nan')
...

class Peak(object):
    area = 0.0
    detector = 'ptc'
    ...
    rec_positions = list(ReconstructedPosition)

class Hit(object):
    channel = 0
    center = 0.0
    ...

class Event(object):
    event_number = 0
    dataset_name = 'Unknown'
    ...
    peaks = list(Peak)
hits = np.array([], dtype=Hit.get_dtype())

Listing 2: The pax Event model
Computing model technology chain

XENON1T computing model technology chain
ROOT with Anaconda

- **Anaconda**: Python scientific libraries bundled together in one installation
- **Conda**: binary packaging and dependency management system
  - Isolated environments ++
  - Python-agnostic
package:
  name: gensim
  version: 0.8.8

source:
  fn: gensim-0.8.8.tar.gz [py2k]
  url: https://pypi.python.org/packages/source/g/gensim/gensim-0.8.8.tar.gz
  md5: 39b47895185f05a81b83ebf1a6748953 [py2k]

# patches:
# List any patch files here
# - fix.patch

build:
  number: 1

requirements:
  build:
    - python
    - setuptools
    - scipy
  run:
    - python
    - scipy

test:
  # Python imports
  imports:
    - gensim.similarities
    - gensim.test
    - gensim.corpora
    - gensim.models

about:
  home: http://radimrehurek.com/gensim
  license: GNU Library or Lesser General Public License (LGPL)
Portable ROOT Anaconda binaries

- dynamic dependencies on GCC/glibc
- ROOT 6 needs GCC 4.8 or newer
- should work on older Linux distributions
  - preferably the same binary (one size fits all)
  - SLC 6 + CERN Developer Toolset (v2) makes it possible to have fresh compiler with old glibc (2.12)
- Please give it a try and report problems
  conda install -c nlesc root
Continuous Integration

-Jenkins

ROOT

<table>
<thead>
<tr>
<th>CONDA_PY</th>
<th>$ROOT, $CONDA_PY</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/2.7</td>
<td>5/3.4</td>
</tr>
<tr>
<td>6/2.7</td>
<td>6/3.4</td>
</tr>
</tbody>
</table>

trigger a build

$docker$

run

build.sh

upload

root-{$ROOT}-py{$CONDA_PY}.tar.bz2

slc6-devtoolset-anaconda

NLeSC / Packages / root

root-conda-recipes
Sustainability & Issues

- Install a conda environment and “don't touch it”
  - New versions of ROOT (and dependencies) will need building/testing/publishing
  - “old glibc” is a moving target
- Conda is stable, but still evolving
OH: TIL installing ROOT is already as easy as `conda install -c nlesc root` via @ibabusch

1:30 PM - 18 Sep 2015

kreczko commented on Nov 25, 2015

@remenska: nice work. Following the conversation in the HSF packaging forum (https://groups.google.com/forum/#!topic/hep-sf-packaging-wg/h4HWHnVkBAA) Will advertise this to our groups.

cdeii commented on Sep 3, 2015

@remenska – That's awesome, thank you!

@ndawe – Should these binaries be used for rootpy testing on travis-ci (either exclusively or in addition to what's there now)?