

PyROOT, cppyy, and the Python ecosphere

Community view for the HSF

Wim Lavrijsen (LBNL)
HEP Software Foundation Workshop
January 20-22, 2015, SLAC

- **PyROOT: alternative interpreter for ROOT**
 - UIs determine productivity: having a choice is good
 - Folks are *very* sensitive about their user interfaces ...
- **cppyy: C++ ↔ Python language bindings**
 - Matches the scale and heterogeneity of HEP codes
 - Provides the underpinnings of PyROOT
 - Used to drive experiments' frameworks
- **Enable HEP software within Python ecosphere**
 - Python as a *high-level productivity language*
 - Bindings give easy access to high performance libraries
 - SciPy, NumPy, iPython/notebooks, etc.

- **PyROOT/cppyy has been in use for > 10 years**
 - First PyROOT prototypes in 2003
 - Became part of ROOT in 2004
 - Main tool for Python bindings in HEP since 2005
 - Collaboration with PyPy project since 2010
 - PyPy = Python+JIT → gets C++-speed for numerical code
- **Wide-spread, and growing:**
 - With ROOT => used in virtually all HEP experiments
 - Dictionary-based bindings guarantee access to data
 - With PyPy => included in most Linux/Mac distros

- **Development to follow changes in hard-/software**
 - Cling backend for PyPy
 - Leverage PyPy STM for multi-core hardware
 - Leverage PyPy JIT for HEP specific use cases
 - Deep integration between Python & Cling run-times
 - Integration with NumPy/Numba
 - On-going tracking of C++ language evolution
 - Python 3.x support
- **All require focused development**
 - These are sizable tasks
 - Not just coding, also updates to docs, tests, etc.

- **PyROOT support/funding has been clumpy**
 - US ATLAS for ATLAS needs
 - Office of Science for ROOT6 development
 - My “copious spare time”
- **But, some great community support**
 - Core developers of the ROOT and PyPy projects
 - *rootpy.org* project
 - Distribution packagers for Mac, Linux
 - Power-users on message forums
 - Multiple sets of bug reporting systems, nightly test systems, mailing lists, forums, and IRC

- **From my experience with other sciences:**
 - Many short-lived, frequently renewed, contacts
 - Bug fixes and optimizations
 - Conference papers (in Computer Science)
 - Larger projects tend to be applications, not SDKs
- **HEP has been unique in delivering *products***
 - Code .. + documentation, + support, + maintenance
- **HSF could provide for:**
 - Uptake of technological changes as they arrive
 - Timely response to needs of experiments
 - Guarantee of continuity and preservation of expertise