IPython: modern tools for interactive & web-enabled scientific computing

Fernando Pérez
http://fperez.org, @fperez_org
Fernando.Perez@berkeley.edu

Helen Wills Neuroscience Institute, UC Berkeley

NERSC User Day, LBL, Berkeley,
Feb 13, 2013
Outline

1 Scientific Python

2 IPython: Interactive Python
Beyond (Floating Point) Number Crunching

Hardware floating point

Extended precision floating point

Symbolic manipulation
Interval arithmetic
Rationals
Arbitrary precision integers

Hardware control
Multi-language integration

Text processing
Databases
Graphical user interfaces
Web interfaces
Hardware control
Multi-language integration

Data formats: HDF5, XML, ...

FORTRAN

Python

Multi-language integration
Scientific Python: a Rich Ecosystem

IPython

NumPy

SciPy

matplotlib

SymPy

Cython

Mayavi

pandas

StatsModels

scikit-learn

scikits-image

PyTables

NetworkX
Outline

1 Scientific Python

2 IPython: Interactive Python
The Lifecycle of a Scientific Idea (schematically)

1. Individual exploratory work
2. Collaborative development
3. Production work (HPC, cloud, parallel)
4. Publication (with reproducible results!)
5. Education

The Problem with most tools
Barriers and discontinuities in workflow in between all the steps
The Lifecycle of a Scientific Idea (schematically)

1. Individual exploratory work
2. Collaborative development
3. Production work (HPC, cloud, parallel)
4. Publication (with reproducible results!)
5. Education

The Problem with most tools
Barriers and discontinuities in workflow in between all the steps
IPython’s goal:
Fluid transitions in all these steps
Demo
Pillar #1: An architecture for interactive computing

![Diagram of IPython architecture]

- **IPython Kernel**
- **Terminal console**
- **Qt Console**
- **Web Notebook**
- **ZeroMQ**

Client: monitor email, publish, ..
Pillar #2: the Notebook Format

- **JSON** but version control-friendly
- Easy for machine processing, **fixable by hand if need be.**
- Lots of hooks for **metadata**
- **Not** Python-specific (R and Ruby notebooks exist, Julia planned)
- Produce Markdown, reST, \LaTeX, HTML, etc...

An open format for **sharing**, **publishing** and **archiving** executable computational work
Documented protocols and formats: a growing ecosystem around IPython
This example computes PI to certain precision using 4 processors and a monte carlo simulation.

```python
import random
from mpi4py import MPI
comm = MPI.COMM_WORLD
import numpy as np

def computePi(nsamples):
    rank, size = comm.Get_rank(), comm.Get_size()
    oldpi, pi = 0.0, 0.0, 0.0
    done = False
    while not done:
        inside = 0
        for i in xrange(nsamples):
            x = random.random()
            y = random.random()
            if (x**2) + (y**2) < 1:
                inside += 1

        oldpi = pi
        pi = (inside * 1.0) / nsamples
        comm.allreduce(pi, op=MPI.SUM)
        if rank == 0:
            print "computation completed.

In [1]: from IPython.parallel import Client
In [2]: rc = Client()
In [3]: rc.ids
[0, 1, 2, 3]
In [4]: dtype = rc[:] # use all engines
In [5]: serial_result = map(lambda x:x**10, range(32))
In [6]: parallel_result = dview.map_sync(lambda x: x**10, range(32))
In [7]: serial_result==parallel_result
True
In [8]: parallel_result
[0, 1, 1024, 59049, 1048576, 9765625, 60466176, 282475249, 1073741824, 3486784401L, 10000000000L, 25937424601L, 61917364224L, 137858491840L, 289254654976L, 57665298269L]"
A vim client to control an IPython kernel/console

Paul Ivanov (Berkeley), https://github.com/ivanov/vim-ipython
Notebooks on Windows Azure Cloud

IPython Notebook on Windows Azure

For a quick overview of installation and IPython, please watch:

The IPython project provides a collection of tools for scientific computing that include powerful interactive shells, high-performance and easy to use parallel libraries and a web-based environment called the IPython Notebook. The Notebook provides a working environment for interactive computing that combines code execution with the creation of a live computational document. These notebook files can contain arbitrary text, mathematical formulas, input code, results, graphics, videos and any other kind of media that a modern web browser is capable of displaying.

Whether you’re absolutely new to Python and want to learn it in a fun, interactive environment or do some serious parallel/technical computing, the IPython Notebook is a great choice. As an illustration of its capabilities, the following screenshot shows the IPython Notebook being used, in combination with the SciPy and matplotlib packages, to analyze the structure of a sound recording:

Simple spectral analysis
Star Cluster: IPython parallel + Notebook on Amazon EC2
Justin Riley (MIT): http://web.mit.edu/star/cluster
### Instance Launcher

Select any instance type to see more information about it. If you are unsure about your options, please consult the documentation.

If you are using NotebookCloud for learning programming, you should select Hello World.

---

#### Your NotebookCloud Servers

No instances (launched from your NotebookCloud account) exist on your AWS account.
## Scientific

- **Software Carpentry**
- **EPD**: Enthought Python Distribution.
- **Continuum**: Anaconda, Wakari.
- **Sage**: open source mathematics.
- **PyRAF**: Space Telescope Science Institute
- **CASA**: Nat. Radio Astronomy Observatory
- **Ganga**: CERN
- **PyMAD**: neutron spectrom., Laue Langevin
- **Sardana**: European Synchrotron Radiation
- **ASCEND**: eng. modeling (Carnegie Mellon).
- **JModelica**: dynamical systems.
- **DASH**: Denver Aerosol Sources and Health.
- **Trilinos**: Sandia National Lab.
- **DoD**: baseline configuration.
- **Mayavi**: 3d visualization, Enthought.
- **NiPyype**: computational pipelines, MIT.

## Web/Other

- **Visual Studio 2010**: MS.
- **Django**.
- **Turbo Gears**.
- **Pylons** web framework
- **Zope** and **Plone** CMS.
- Axon Shell, BBC
- **Kamaelia**.
- **Schevo** database.
- **Pitz**: distributed task/bug tracking.
- **iVR** (interactive Virtual Reality).
- **Movable Python** (portable Python environment).
- ...
Brian Granger - Physics, Cal State San Luis Obispo
Min Ragan-Kelley - Nuclear Engineering, UC Berkeley
Matthias Bussonnier - Physics, Institut Curie, Paris
Brad Froehle - Mathematics, UC Berkeley
Paul Ivanov - Neuroscience, UC Berkeley.
Robert Kern - Enthought
Thomas Kluyver - Biology, U. Sheffield
Jonathan March - Enthought
Evan Patterson - Physics, Caltech/Enthought
John Hunter - TradeLink Securities, Chicago.
Stefan van der Walt - Applied Math, U. Stellenbosch
Satra Ghosh - MIT Neuroscience
Gaël Varoquaux - Neurospin (Orsay, France)
Ville Vainio - CS, Tampere University of Technology, Finland
Ondrej Certik - Physics, U Nevada Reno
Darren Dale - Cornell
Justin Riley - MIT
Mark Voorhies - UC San Francisco
Nicholas Rougier - INRIA Nancy Grand Est
Thomas Spura - Fedora project
Many more! (~150 commit authors)
Support
Thank you!

- **Enthought**, Austin, TX: **Lots!**
- **Microsoft**: WinHPC support, Visual Studio integration, Azure (thanks to Shahrokh Mortazavi).
- **DoD/DRC Inc**: funding through Sept. 2012 (thanks to Jose Unpingco and Chris Keeses).
- **NIH**: via NiPy grant
- **NSF**: via Sage compmath grant
- **Tech-X Corp., Boulder, CO**: Parallel/notebook (previous versions)
Recent Stable Funding

2-year funding, core team (7 people)

Stanford: reproducibility & data sharing (neuroimaging, other data-intensive cases)
NumFOCUS: http://numfocus.org
Foundation to promote Open Source Scientific Computing

- Support the development of multiple projects.
- Community-created and driven.
- A neutral ground for industry, academia and government to support scientific open source.
- 501(c)3 - donations are tax-exempt in the USA