IPython 2015 Final Report - Sloan Foundation

December 28, 2015

Submitted by: Fernando Perez and Brian Granger.

In December 2012, IPython was awarded a $1.15 million grant from the Alfred P. Sloan Foundation, funding the core team to improve the Notebook as well as general IPython development for the 2013-2014 period.

This is a final report for the project's progress, covering the period from our previous interim report until the end of our no-cost extension, in September of 2015 (extended from the project's original end date of April 2015).

1 Activities on our roadmap

For this cycle, we focused our attention on our two main remaining items:

- A multiuser notebook server, which we called **JupyterHub**.
- A system of interactive widgets in the IPython Notebook, now officially released as the **ipywidgets** project.

We now briefly report on our progress on these two areas, and then we continue with a summary of other results, broader impacts and finally a collection of the proposed indicators of success from our original proposal. But first, we start with a section outlining an expansion in scope of the project that led to the creation of a new named entity, called "Project Jupyter", a term we will use throughout this report.

2 Project Jupyter

As we described in our previous project report, our architecture could be used to support kernels for the Notebook in languages other than Python, and at the time we listed 8 separate non-Python kernels that had been developed. We rapidly realized that it was important to present to the broader community a project name that would not be so tightly identified with the Python programming language, since for many of these other projects it felt awkward to say that they were a "Haskell kernel for IPython", for example.

At the Scipy 2014 conference, we announced our new name, "Project Jupyter". This was not strictly a renaming of IPython, but rather a renaming of *parts* of IPython, all the components that were language-agnostic and that could be reused across all programming languages. IPython will continue to exist, now providing the Python-specific kernel for the Jupyter architecture.

We proceeded with this renaming in a gradual way: we created a new Jupyter website and broke the original IPython project repository into new repositories hosted in a new GitHub organization at github.com/jupyter, and allowed the community to get used to the change without deleting the existing IPython infrastructure. The IPython tools, like the Notebook viewer, continue to exist at nbviewer.ipython.org for backwards compatibility, but we now promote officially as our new URL nbviewer.jupyter.org instead.

Overall, this transition has been a success. The community has embraced our language-agnostic vision wholeheartedly, and while we continue to build IPython as our official kernel, many other communities have adopted our platform. Our list of compatible kernels now lists over 50 of them, and it continues to grow.

3 JupyterHub: a Multi-User Notebook Server

As of Nov 4, 2015, we have made three releases (up to version 0.3.0) of JupyterHub, a tool that offers a flexible design for deploying Jupyter Notebook services in a multiuser context.

The basic architecture of JupyterHub assumes only that there is a Unix-class multiuser server where users can log in with their own user accounts, and it will spawn Jupyter Notebook processes on their behalf when they log in. By making only this simple assumption, the system can be deployed in a typical classroom, university or research group setting with a server or cluster with moderate effort, and without requiring extensive infrastructure expertise. However, the JupyterHub design has two key extension points:

1. *Authentication:* the way users are authenticated into the system is easily extensible, and today multiple custom autneticators have already been developed and deployed, both by our team and by third-parties.

2. *Execution:* the notebook server that is initiated on behalf of each user can be replaced by one that instead of starting a local process, executes in a customized way, for example running in a separate cluster, or in a Docker container.

Taking advantage of both of these extension points, multiple custom configurations become possible. This flexibility has led to the development of new authenticators and notebook spawning tools, that allow users to compose sophisticated deployments that go beyond the basic, out-of-the-box design.

There are many deployments of JupyterHub world-wide, using various combinations of customized authentication and execution decisions, depending on the specific use case being targeted. Here we list a few relevant ones in scientific research and education, but we regularly see questions on our mailing list and posts online indicating a much broader use base.

1. Bryn Mawr: led by professor Doug Blank, this was actually the earliest deployment of JupyterHub, since he started using it when the code first hit GitHub in its pre-alpha state. Doug provided us with invaluable feedback as he used it in the Fall of 2014 to teach his undergraduate programming courses while the tool was still in its embryonic phase.

2. CalPoly. Brian Granger has deployed JupyterHub for teaching Data Science at CalPoly at a cluster he manages, and as part of this effort he has developed tools to streamline the process in similar contexts (educators, researchers and others using small/medium size deployments with limited engineering staff and support). He worked closely with Jess Hamrick in the development of nbgrader, the homework grading tool described below.

3. Berkeley: at UC Berkeley, educational deployments have been made initially by Jess Hamrick for the Spring 2015 Computational Models of Cognition course, and based on her success, by the campus team teaching the undergraduate Data Science curriculum. See the broader impact section below for details.

4. CERN: at the upcoming CS3 workshop next January, Danilo Piparo from CERN will be presenting a JupyterHub-based service for scientific computing which integrates Notebooks with cloud storage, syncronization, sharing, OpenStack compute and data analysis frameworks. Two members of our team, Min Ragan-Kelley and Thomas Kluyver, recently visted CERN and interacted with the developers of this project.

5. Wikipedia: the Wikipedia "lab" division now maintains a JupyterHub, described here, and which can be accessed by anyone wwith a MediaWiki account thanks to the development by their team of a custom MediaWiki authenticator that they have also released as open source here.

6. NERSC, the National Energy Research Scientific Computing Center, is the Department of Energy's primary scientific computing facility for its Office of Science. It now offers its users a JupyterHub instance, and we are collaborating with the team doing this deployment to understand how JupyterHub can scale to the nation's largest supercomputing facilities.

7. SDSC is the San Diego SuperComputing Center, one of the country's largest university-run supercomputing facilities, and similarly to NERSC, is offering also a JupyterHub deployment on an experimental basis. We are also in contact with this team to understand how their needs can help us improve JupyterHub's design for HPC environments.

8. Astronomy department at ETH Zurich

9. U. Minnessota Population Center

10. U. Sheffield.

11. Danish e-Infrastructure cooperation.

3.1 nbgrader: grading of notebooks in JupyterHub

While not originally part of our development roadmap, during the educational deployment of JupyterHub in real-world class scenarios by Brian Granger (at CalPoly) and Jess Hamrick (at UC Berkeley), it became clear that while notebooks were an excellent format for distributing complex and rich homework sets to students in computationally intensive courses, their grading presented a number of challenges to instructors.

Led by the two of them, our team developed a new tool called nbgrader that integrates the management, delivery, automated (where possible) and manual (where relevant or necessary) grading of homework sets as notebooks for classes into JupyterHub. Both Brian and Jess used it for their own courses, and Jess presented the results of her experience at the 2015 SciPy conference (reference below).

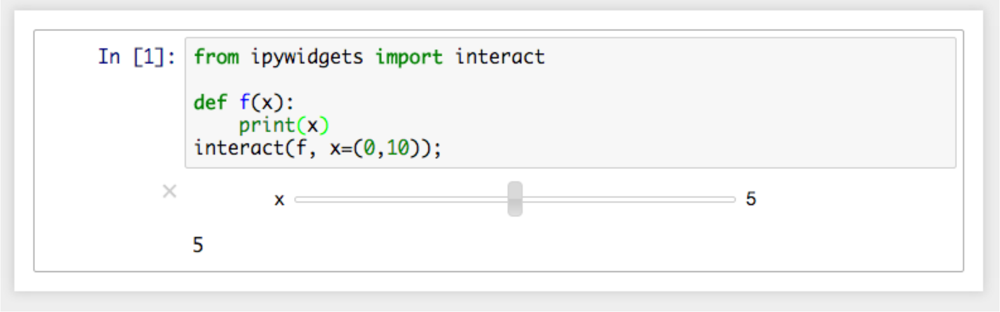
The project has now seen contributions from others in the open source community and questions about it in our mailing lists, indicating usage in other course settings beyond our team.

4 Interactive Widgets for HTML: `ipywidgets`

Our second main focus of development for this period was the creation of a library of HTML widgets, that would allow users to put interactive controls such as buttons, sliders and menu boxes in the IPython Notebook, connected to the code being executed by the Python kernel in the back end. This would permit the graphical control of parameters, which in some cases may be a more fluid and convenient workflow than editing a variable and re-executing an input cell multiple times.

We have successfully implemented our widget library; the initial implementation lived in the `IPython.html.widgets` namespace, but as part of our recent modularization of the project's structure, we have released this functionality as a standalone project named ipywidgets.

These tools allow users to write a minimal amount of code to obtain a set of graphical controls that they can then use to interact with their functions. The following figure illustrates the minimal example that produces a slider that prints its value to the screen:

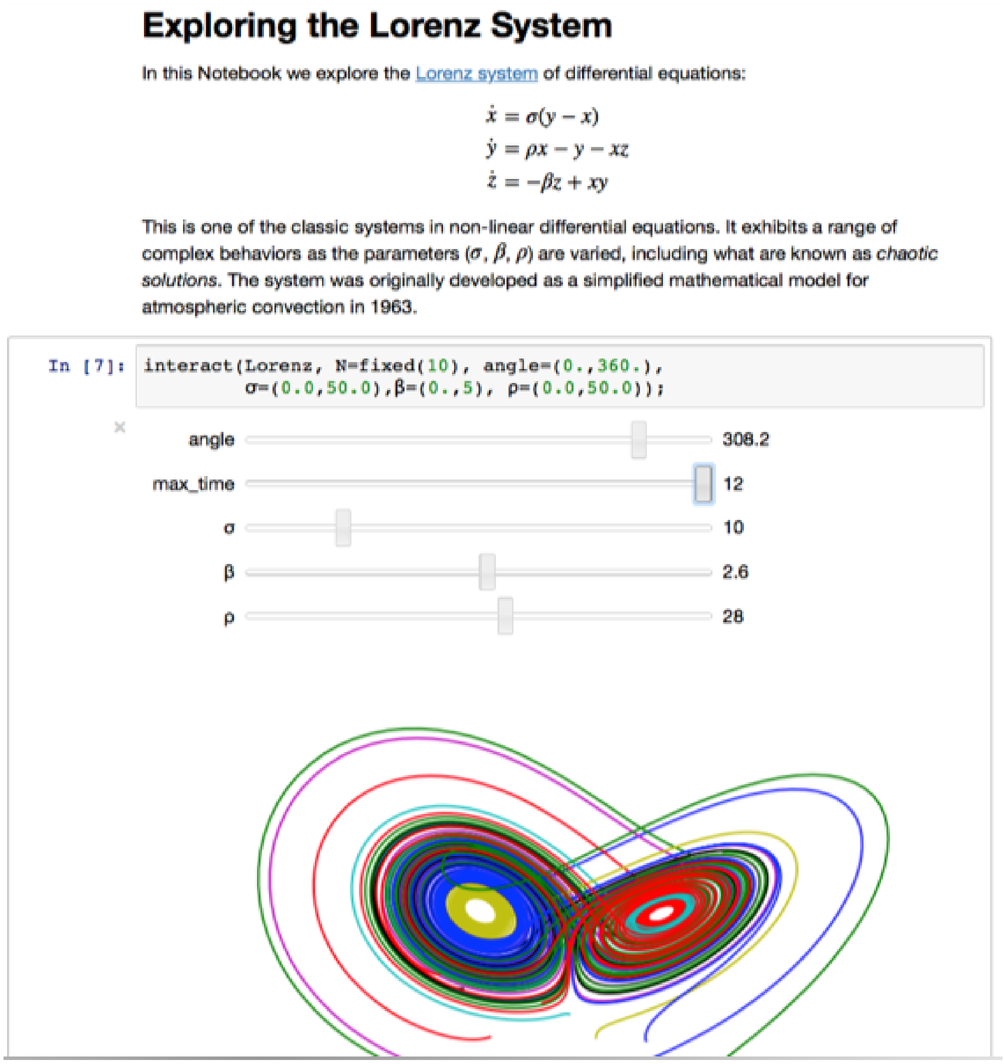


```
In [1]: from ipywidgets import interact

        def f(x):
            print(x)
        interact(f, x=(0,10));
```

```
5
```

We have released several versions of the library (version 4.1.1 came out on October 26), and it has seen wide adoption in research, education and industry. While we continue actively developing the project, we consider our core targets for this cycle to have been successfully met, as the library allows users to provide, with a minimal amount of effort, interactive controls for the meaningful exploration of computational ideas, as illustrated in the figure below (manipulating the Lorentz equations).



Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (σ, β, ρ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                 σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0));
```

angle 308.2

max_time 12

σ 10

β 2.6

ρ 28

4.1 Live demo in partnership with Nature and Rackspace

Using this idea, we partnered with the editors at Nature, who were preparing an article for Nature News about IPython as part of their series on tools for data sharing and reproducible research, and prepared a live demo of this system hosted on the nature.com domain. This demo is hosted by servers donated by Rackspace.com, a company that supports our project with hosting services and engineering staff who participate as core developers.

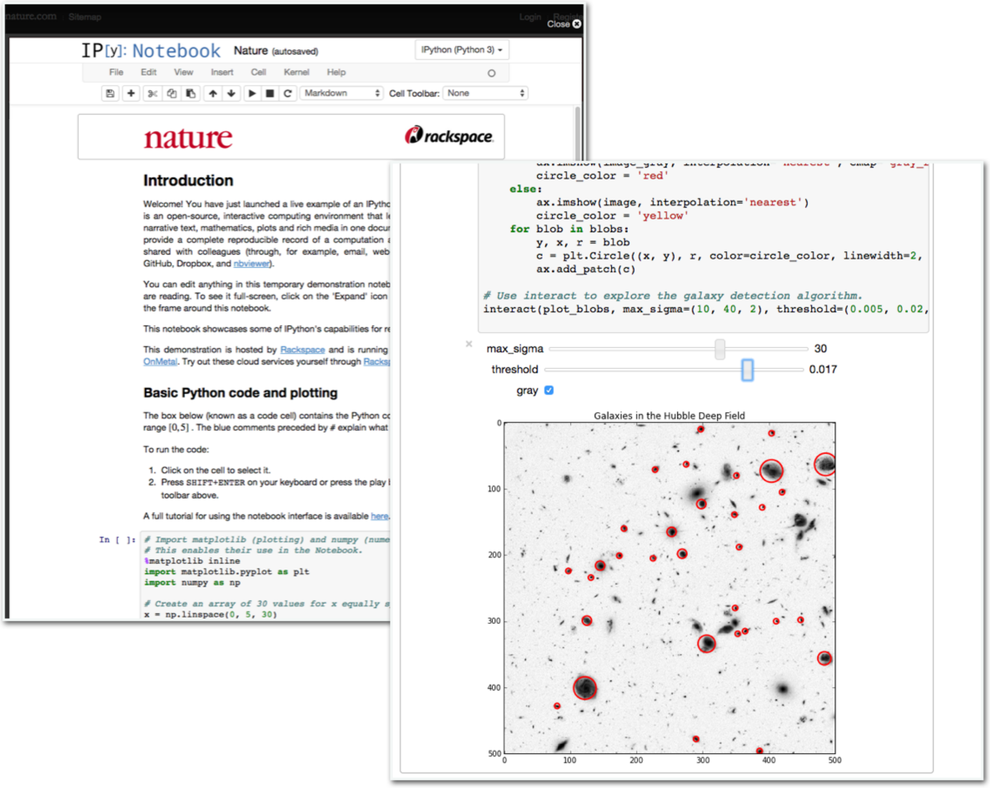
This demo contains snippets of realistic code illustrating three scenarios:

1. A simple static plot of a quadratic function, $y = x^2$.
2. A demonstration of aliasing effects in the Fourier transform.
3. A galaxy-detection algorithm in an image from the Hubble Deep Field.

The purpose of this demonstration is to validate the idea of a live, executable document combining narrative, code, data and scientifically relevant content. We exposed it to the readers so they could directly interact with the source document, modify it, re-execute it, explore the parameters beyond the inputs originally provided by the authors and obtain new results.

The second and third examples are interactive widgets using our `ipywidgets` library, allowing the readers of the demo to drag sliders and directly see the results of their changes, such as modifying algorithm parameters.

This demo was enormously successful, breaking readership records for Nature for this class of article (based on what the information the staff had given to us to prepare server capacity prior to release).



Introduction

Welcome! You have just launched a live example of an IPytho
is an open-source, interactive computing environment that le
narrative text, mathematics, plots and rich media in one docu
provide a complete reproducible record of a computation a
shared with colleagues (through, for example, email, web
GitHub, Dropbox, and nbviewer).

You can edit anything in this temporary demonstration noteb
are reading. To see it full-screen, click on the 'Expand' icon
the frame around this notebook.

This notebook showcases some of IPython's capabilities for re

This demonstration is hosted by Rackspace and is running
OnMetal. Try out these cloud services yourself through Racksp

Basic Python code and plotting

The box below (known as a code cell) contains the Python co
range [0, 5]. The blue comments preceded by # explain what

To run the code:

1. Click on the cell to select it.
2. Press SHIFT+ENTER on your keyboard or press the play b
toolbar above.

A full tutorial for using the notebook interface is available here.

```
In [ ]: # Import matplotlib (plotting) and numpy (nume
        # This enables their use in the Notebook.
        %matplotlib inline
        import matplotlib.pyplot as plt
        import numpy as np

        # Create an array of 30 values for x equally s
        x = np.linspace(0, 5, 30)
```

```
        ax.imshow(image_gray, interpolation='nearest', cmap='gray_r
        circle_color = 'red'
    else:
        ax.imshow(image, interpolation='nearest')
        circle_color = 'yellow'
    for blob in blobs:
        y, x, r = blob
        c = plt.Circle((x, y), r, color=circle_color, linewidth=2,
        ax.add_patch(c)

# Use interact to explore the galaxy detection algorithm.
interact(plot_blobs, max_sigma=(10, 40, 2), threshold=(0.005, 0.02,
```

max_sigma 30
threshold 0.017
gray ☑

5 Talks, seminars and publications

Our team has been busy engaging multiple communities, from academia to industry, by delivering talks at domain-specific conferences, seminars, lecture series and tutorials.

In this reporting cycle, we want to highlight that we have diversified our team's presence, and we have a broader cross-section of our team members participating at multiple events world-wide, not only the lead PIs taking this role.

The following is a summary of our activities on this front:

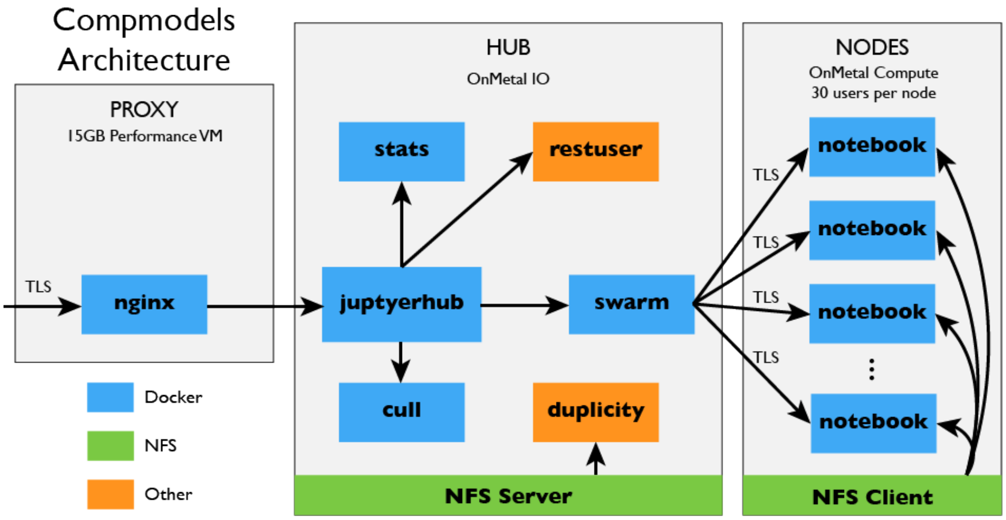
- G. Wilson, F. Perez, P. Norvig, *Teaching computing with the IPython notebook*, Proceedings of the 45th ACM technical symposium on Computer science education, pp. 740-740. ACM, 2014.
- F. Perez, *Jupyter & KBase: reproducibility and collaboration in open science*, invited talk, Kavli symposium on microbiome research, National Academies, Washington DC, September 2015.
- J. Hamrick, M. Ragan-Kelley, K. Kelley, *Teaching with IPython/Jupyter Notebooks and JupyterHub*, SciPy 2015, Austin, July 2015.
- J. Frederic, *Past, Present, and Future of Jupyter*, PyData 2015, Seattle, July 2015.
- M. Bussonnier, J. Frederic, T. Kluyver, *Jupyter Advanced Topics Tutorial*, Tutorial at SciPy 2015, Austin, July 2015.
- F. Perez, Keynote, UCAR Software Engineering Assembly Conference, Boulder, April 2015.
- F. Perez, M. Ragan-Kelley, *Project Jupyter: a language-independent architecture for CSE, from interactive computing to reproducible publications*, SIAM CSE 2015, Salt Lake, March 2015.
- F. Perez, M. Ragan-Kelley, *Jupyter widgets: interactive computing through the browser in any programming language*, poster, SIAM CSE 2015, Salt Lake, March 2015.
- J. Hamrick, M. Ragan-Kelley, K. Kelley, *Using Docker with JupyterHub for Education*, invited talk, Docker HQ, San Francisco, March 2015.
- M. Ragan-Kelley, *Jupyter Widgets: Interactive Computing Through the Browser in Any Programming Language* (poster), SIAM CSE, Salt Lake, March 2015.
- T. Kluyver, M. Ragan-Kelley, *Learn about the latest developments in IPython*, San Francisco Python Meetup, February 2015.
- J. Frederic, K. Kelley, *Introduction to the Jupyter Notebook*, Tutorial at Strata 2015, San Jose, February 2015.
- B. Granger, *Jupyter/IPython: Scaling Data Science to Organizations and the Open Web*, Strata Silicon Valley, February 2015.
- F. Perez, *IPython: from interactive computing to computational narratives*, AAAS meeting, San Jose, February 2015.
- F. Perez, *Project Jupyter: computational narratives as tools for communication in a data-driven world*, invited talk at Kavli symposium on scientific journalism, San Jose, February 2015.

- M. Ragan-Kelley, *The Jupyter/IPython architecture: a unified view of computational research, from interactive exploration to communication and publication*, AGU Fall Meeting 2014, San Francisco, December 2014.
- F. Perez, *Big Data, Science and Society: passing fad or lasting change?*, memorial lecture, Physics Dept, Universidad de Antioquia, Colombia, November 2014.
- F. Perez, Keynote, PyCon Brazil, Porto de Galhinas, Brazil, November 2014.
- B. Granger, F. Perez, *IPython tutorial*, PyData at Strata NYC 2014, New York, October 2014.
- F. Perez, *IPython & Project Jupyter*, Invited talk at CodeNeuro 2014, San Francisco, October 2014.
- B. Granger, F. Perez, *Multi-language Data Science with IPython, IJulia, IR, and Friends*, Strata NYC 2014, New York, October 2014.
- F. Perez, *IPython and Jupyter From a REPL to an architecture for web-enabled data science*, Invited talk, Python at Microsoft Day, Redmond, Sept. 2014.
- M. Bussonnier, T. Kluyver, *IPython: protocol, kernels and new features*, Euroscipy 2014, Cambridge, August 2014.
- F. Perez, *Software Sustainability? Lessons from IPython*, Invited talk at Workshop on Sustainable Software for Science: Practices and Experiences, SciPy'14 conference, Austin, TX, July 2014.
- J. Frederic, B. Granger, *Fundamentals of the IPython Display Architecture and Interactive Widgets*, Tutorial, Scipy 2014, Austin, July 2014.
- F. Perez, M. Ragan-Kelley, *Interactive Parallel Computing with IPython*, Tutorial, Scipy 2014, Austin, July 2014.
- P. Ivanov, *putting the v in IPython: vim-ipython and ipython-vimception*, Scipy 2014, Austin, July 2014.
- F. Perez, *IPython: a language-independent framework for computation and data*, Workshop on Algorithms for Modern Massive Data Sets (MMDS 2014), Berkeley, June 2014.
- J. Frederic, B. Granger, *IPython Interactive Widgets*, Tutorial at PyData SV 2014, Palo Alto, May 2014.
- P. Ivanov, T. Kluyver, *The IPython protocol, frontends and kernels*, PyData SV 2014, Palo Alto, May 2014.
- M. Ragan-Kelley, *IPython: what's new, what's cool, and what's coming*, PyData SV 2014, Palo Alto, May 2014.
- M. Ragan-Kelley, *Parallel Computing with IPython*, Bay Piggies meeting, April 24, 2014.
- F. Perez, Keynote, PyCon APAC, Taipei, May 2014.
- F. Perez, Keynote, PyCon, Montréal, April 2014.
- F. Perez, *IPython: from interactive computing to computational narratives*, Invited talk, Weathering the Data Storm (Third Annual Symposium On The Future Of Computation In Science And Engineering), Institute for Applied Computational Science, Harvard, January 2014.

6 Broader impacts

6.1 JupyterHub in Education

Jessica Hamrick, a UC Berkeley PhD student, core Jupyter developer and Steering Committe member, used the flexible JupyterHub architecture described above to lead one of the most sophisticated deployments of the system to date. Using a combination of Docker, nginx and NFS servers on Rackspace-donated services, in collaboration with other members of our team, she built a JupyterHub system for a 220-student class called Computational Models of Cognition (COGSCI 131) where she was a graduate student assistant. This sophisticated deployment is described in this blog post, and she presented the results of this work at the SciPy 2015 conference. The architectural summary of the system is illustrated here:



Furthermore, in the Summer of 2015, UC Berkeley adopted and modified this deployment as the basis for the new campus-wide Data Science curriculum being offered to the incoming freshman class, called Data8.org. This course offers a newly-developed online data science textbook called Computational and Inferential Thinking, hosted on a live JupyterHub installation run by Berkeley staff (who coordinated with Jess and adapted her original work to run on Microsoft Azure infrastructure at ds8.berkeley.edu), and is part of the new campus-wide initiative in data science education that includes discipline-oriented connector courses, and a broader effort to consider the role of data science in the undergraduate curriculum.

6.2 University courses using Jupyter

The uses of Jupyter in education continue to rise. At the SciPy'2015 conference, Jess Hamrick's presentation about her deployment of JupyterHub at UC Berkeley was extremely well received, and a "Birds of a Feather" session she organized to discuss the educational uses of the platform led to the creation of a dedicated mailing list for the discussion of educational topics.

On that discussion group, anyone who is interested can participate, and we are now tracking courses being taught with Jupyter via a collaboratively edited spreadsheet, that can continue to be updated as more people report new courses. At the time of this writing, the spreadsheet lists

41 courses being taught with IPython/Jupyter notebooks as their main educational materials, and many of them are also using JupyterHub as the deployment platform.

We want to highlight one course in particular: Lorena Barba and her colleagues deployed an open MOOC, "Practical Numerical Methods with Python", based on the OpenEdX platform, entirely written as a collection of Jupyter Notebooks and fully available on GitHub. This course was taught collaboratively by Lorena and two other faculty, one at U. Southampton and one at UPC Chile. Lorena will be a visiting scholar at BIDS during the Spring 2016 semester, where we will draw on her extensive expertise using our tools in education to improve our system.

6.3 Special interview in Nature - cover mention

The interactive demo described above accompanied an article Nature did about IPython as a tool for sharing scientific analysis and research reproducibility; this piece was included in their November 5, 2014 issue and was highlighted in that issue's cover:

nature
FULL SPEED AHEAD
Plasma wakefield machines — the particle accelerators of the future?
LIFE AFTER THE WALL
CASH, CONFLICT, CONSERVATION
SHARE AND SHARE ALIKE

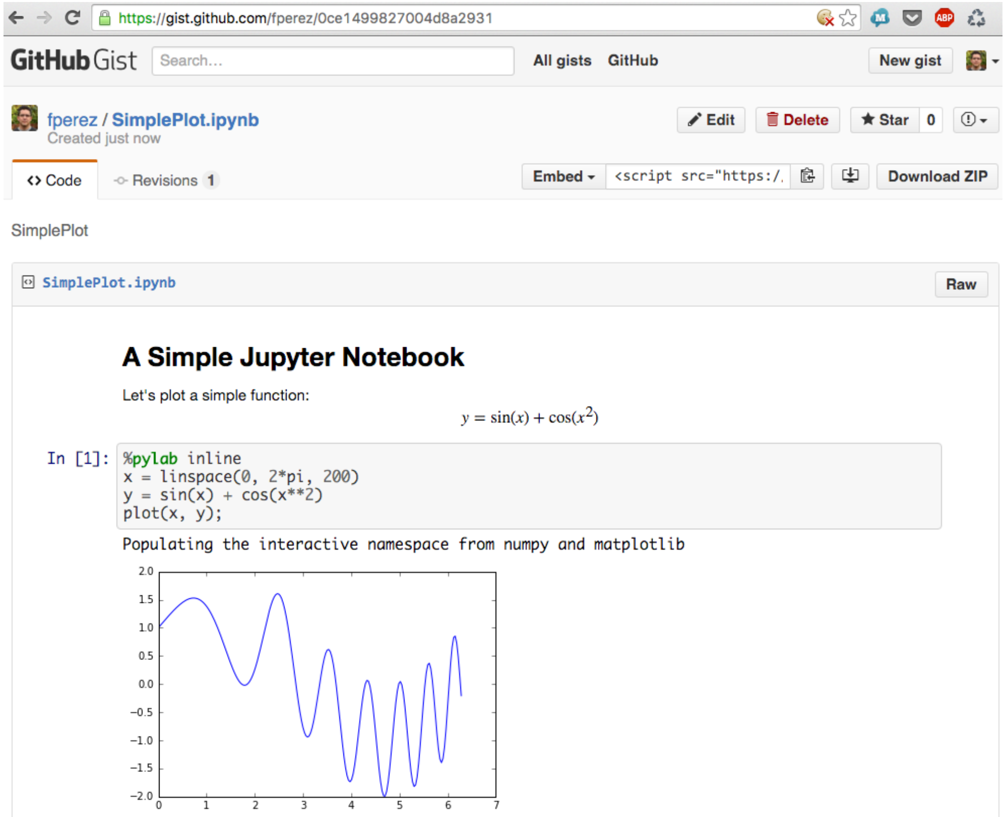
6.4 NBViewer and native rendering of Notebooks on GitHub

In our 2013 progress report, we described how we operate the IPython Notebook Viewer web service that renders static HTML views of notebooks, enabling easy sharing of notebooks with a simple URL. As mentioned above, we now offer the notebook viewer also over the Jupyter domain, and that is the main one we advertise, though the IPython one remains available to avoid breaking any existing URLs.

But in May 7, 2015, GitHub announced publicly that it would begin rendering Jupyter Notebooks natively. This was an extremely important and impactful achievement for us; we had been in discussions with the technical team at GitHub responsible for this implementation for a long time, and we were delighted once this announcement was made. From a technical standpoint, they are using our own tools (the `nbconvert` libraries we discussed in our previous report, and which we continue to develop and improve), deploying them inside of their own infrastructure.

At the time of release of this feature, there were over 200,000 public notebooks on GitHub, and this meant that users would immediately see their notebooks rendered, making the Jupyter Notebook a first-class citizen in GitHub and a much more natural part of the workflow.

The figure below shows how Notebooks are rendered on GitHub.



A Simple Jupyter Notebook

Let's plot a simple function:

$$y = \sin(x) + \cos(x^2)$$

```
In [1]: %pylab inline
        x = linspace(0, 2*pi, 200)
        y = sin(x) + cos(x**2)
        plot(x, y);
```

```
Populating the interactive namespace from numpy and matplotlib
```

6.5 Commercial projects

Since IPython is licensed under a liberal license (BSD) that allows commercial use, we have seen a rise in commercial vendors that incorporate IPython/Jupyter in their products. In this reporting cycle, we have received two new corporate donations, one from Google ($100,000) and one from Continuum Analytics ($100,000). Both of these have been made as unrestricted research gifts, which allow us to allocate these funds to tasks beyond the specific targets of existing grants, and are thus extremely valuable resources for the project.

We have seen significant releases and engagement from commercial entities, such as:

- In July 2015, Microsoft introduced support for Jupyter Notebooks in Azure ML Studio, their cloud computing platform for machine learning and data science.
- In Summer 2015, IBM launched a new product, the Data Scientist Workbench, at datascientistworkbench.com, that provides hosted Jupyter Notebooks on the IBM cloud, backed by IBM resources, with additional features for sharing and collaboration. Additionally, IBM is actively participating in the development of new open source projects meant to become part of Jupyter itself, with several projects currently part of our formal incubation process.

- In May 2015, O'Reilly Media announced they were "making IPython Notebooks a first-class authoring environment for our publishing program, on par with Word or our Atlas platform". We have engaged the technical leadership at O'Reilly, and they have developed the necessary backend resources to support this as an open-source tool called Thebe.

- In September 2015, Bloomberg released bqplot, an interactive plotting library directly designed for the Jupyter notebook and based on the `ipywidgets` architecture. We have worked very closely with the Bloomberg team behind this project, and they are now major contributors to the project in multiple areas, making continuous and significant contributions throughout.

- Plot.ly, a company that develops interactive visualization tools, has developed extensive support for IPython Notebooks on their website and documentation. [Full disclosure: F. Perez serves on the board of advisors of the company, and has filed the required conflict of interest disclosures with both UC Berkeley and Lawrence Berkeley National Laboratory].

7 Evaluation against original success metrics

We will now briefly assess our overall results against our proposed project deliverables and assessment metrics (§ 5 of our original grant proposal):

7.1 Multiuser Notebook Server

As discussed above, JupyterHub has met our goals, and we now continue its development based on the foundation laid so far.

7.2 Interactive widgets

Similarly, `ipywidgets` is currently a successful foundation, though by no means finished and under active development.

7.3 File format specification and conversion tools

The combination of the `nbconvert` tools, our Notebook viewer service, its adoption by entities as important as GitHub, testify to the success of this part of our development.

We still have important work to do on this front, however: the creation of very complex scientific documents in the notebook is still hampered by limitations of the markdown toolchain for features like cross-referencing, bibliographic management, figure captioning or complex tables, for example. Some of these are outside of our scope to develop, and there is active work in the community addressing them, but once progress is made by others, we expect we will be able to better integrate those improvements into our workflow.

In our new funding cycle, we are dedicating resources to also considering how our architecture can support document models that may go beyond the Notebook format in the context of scientific and scholarly publishing, something which we consider still to be an open and important question.

7.4 Domain-specific use case: executable lecture notes to accompany a course in Applied Statistics

These notes are now available online as part of the Stats 191 course at Stanford University, taught by Professor Jonathan Taylor, and the source material is also publicly available on GitHub.

7.5 Project stewardship

We have maintained a reasonable, if not perfect, schedule of releases and developer meetings, we have introduced our open developer meetings on YouTube and have scaled our communication channels to match the growth of our community. This growth continues to be a challenge for the project, and it hasn't always been easy, but we are aware of the importance of maintaining a balance between the addition of features and contributions (financial or technical) and growing a long-term, robust and resilient community that will give the project a healthy outlook.

7.6 Success metrics and assessment

In our original proposal, we stated we would use the following assessment metrics:

1. Deployment of 5 instances of the multiuser notebook server in research groups, classroom or company settings (by groups not affiliated to us).
2. Development of 5 websites or projects that integrate the interactive widgets capability for educational or data exploration purposes.
3. Adoption of the IPython Notebook as the teaching tool for 3 university courses.
4. Release online of 5 sets of learning materials or tutorials by independent parties.
5. Publication of 2 scientific articles by authors beyond our team, using IPython Notebooks to provide reproducible results.

Regarding #1, as we discussed in §3, we have already listed 9 deployments outside Berkeley and CalPoly, and this is only a small list of high-profile and scientifically siginificant ones (on our mailing list, we have frequent queries form many more users reporting activity).

Regarding #2, there are both high-profile projects from commercial actors, like:

- Bloomberg's bqplot
- IBM's Declarative Widget system (currently under incubation for eventual adoption as an official Jupyter project)
- Plot.ly has developed support for our widgets combined with their interactive JavaScript visualization tools

And also specialized ones, such as scikit-spectra, a toolkit for building GUIs for the analysis of spectroscopic data with widgets.

Finally, multiple educators are using the widget system for courses, and making critical use of these interactive capabilites to illustrate concepts. We highlight this with a recent quote from a Mechanical Engineering professor who sent us this message:

> I just wanted to express my gratitude for your creation and continued development of iPython-> jupyter. I just started using it in May, after just embracing Python in January. I just completed an entire Nonlinear Dynamics class (for Mechanical and Aerospace engineering students) based on Nayfeh and Balachandran (1995) and made all of my

> notes in iPython, including creation of interactive displays showing sources and sinks the stability and locations of which are functionally dependent on parameters. I don’t think the examples would be sufficiently simply understood as a general example, but the interactivity certainly brought the concept to life for my students.
>
> Professor Joseph C. Slater
>
> College of Engineering and Computer Science, Wright State University
>
> http://cecs.wright.edu/˜jslater

In summary, we think this target has been successfully met, and we know from traffic on our mailing list and other discussion channels (Gitter chat rooms and GitHub issues) that many more are using the system than the ones we list here.

Regarding #3, as we discuss in §6.2, there are now over 40 courses (that we know of) using the platform.

Finally, both #4 and #5 are covered in our “Gallery of Interesting IPython Notebooks”, where we maintain an informal index of notable materials that are publicly available in Notebook format. There are far more than 5 sets of educational materials and 2 scientific articles (the section on reproducible publications has 25 listed at the time of this writing) listed on that page.

8 Conclusion

We believe that the project has successfully met its goals for the funding period, and we would like to sincerely thank the Sloan Foundation for its generous support, and its flexibility in allowing us to extend beyond the original period with a no-cost extension.

Important intellectual, technical and community development challenges remain, and we are addressing those in the new funding cycle that the Foundation is also supporting in its new partnership with the Gordon and Betty Moore Foundation and the Helmsley Trust.

But thanks to the support for this period, we were able to grow from a small and struggling open source effort into a large project that is now having significant impact in academic research, scholarly publishing, journalism, education and industry. We could not have done this without the Foundation’s support, and we hope to continue builiding upon this towards a better future of open, collaborative and reproducible research in all scientific disciplines, for a better society.

Colophon

This report was prepared entirely as a Jupyter Notebook. The final PDF was produced using the `nbconvert` tool developed with the support of this grant, using the command `jupyter nbconvert --to pdf <notebook-filename>`, without any further manual modification.