Project Jupyter: From Computational Notebooks to Large Scale Data Science with Sensitive Data

Brian Granger
Cal Poly, Physics/Data Science
Project Jupyter, Co-Founder

ACM Learning Seminar
September 2018
Outline

- Jupyter + Computational Notebooks
- Data Science in Large, Complex Organizations
  - JupyterLab
  - JupyterHub
Project Jupyter exists to develop open-source software, open-standards and services for interactive and reproducible computing.
The Jupyter Notebook

- Project Jupyter ([https://jupyter.org](https://jupyter.org)) started in 2014 as a spinoff of IPython
- Flagship application is the Jupyter Notebook
- Interactive, exploratory, browser-based computing environment for data science, scientific computing, ML/AI
- Notebook document format (.ipynb):
  - Live code, narrative text, equations (LaTeX), images, visualizations, audio
  - Reproducible Computational Narrative
- ~100 programming languages supported
- Over 500 contributors across 100s of GitHub repositories.
- 2017 ACM Software System Award.

Example notebook from the LIGO Collaboration
Before Moving On: Attribution?
Who Builds Jupyter?

- Jupyter Steering Council:
  - Fernando Perez, Brian Granger, Min Ragan-Kelley, Paul Ivanov, Thomas Kluyver, Jason Grout, Matthias Bussonnier, Damian Avila, Steven Silvester, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Carol Willing, Sylvain Corlay, Peter Parente, Ana Ruvalcaba, Afshin Darian, M Pacer.

- Other Core Jupyter Contributors:
  - Chris Holdgraf, Yuvi Panda, M Pacer, Ian Rose, Tim Head, Jessica Forde, Jamie Whitacre, Grant Nestor, Chris Colbert, Cameron Oelsen, Tim George, Maarten Breddels, 100s others.

- Dozens of interns at Cal Poly

- Funding
  - Alfred P. Sloan Foundation, Moore Foundation, Helmsley Trust, Schmidt Foundation

- NumFOCUS: Parent 501(c)3 for Project Jupyter and other open-source projects

How to think about the contributions of different people? What is the right narrative?
Jupyter is not the heroic work of one person, or even a small number of people.
Jupyter is created by a large number of people with different strengths working in diverse teams.
Onwards!
As of Summer 2018, Asia is the most represented continent in Jupyter’s web traffic.

<table>
<thead>
<tr>
<th>Country</th>
<th>Sessions</th>
<th>% New Sessions</th>
<th>New Users</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>United States</strong></td>
<td>326,770</td>
<td>53.90%</td>
<td>176,128</td>
</tr>
<tr>
<td><strong>China</strong></td>
<td>82,152</td>
<td>58.00%</td>
<td>47,652</td>
</tr>
<tr>
<td><strong>India</strong></td>
<td>77,828</td>
<td>52.25%</td>
<td>40,664</td>
</tr>
<tr>
<td><strong>United Kingdom</strong></td>
<td>44,005</td>
<td>55.12%</td>
<td>24,255</td>
</tr>
<tr>
<td><strong>Germany</strong></td>
<td>37,732</td>
<td>56.24%</td>
<td>21,221</td>
</tr>
<tr>
<td><strong>Russia</strong></td>
<td>37,502</td>
<td>43.94%</td>
<td>16,479</td>
</tr>
<tr>
<td><strong>Canada</strong></td>
<td>31,976</td>
<td>57.75%</td>
<td>18,465</td>
</tr>
<tr>
<td><strong>France</strong></td>
<td>26,359</td>
<td>59.48%</td>
<td>15,678</td>
</tr>
<tr>
<td><strong>Japan</strong></td>
<td>24,004</td>
<td>56.51%</td>
<td>13,564</td>
</tr>
<tr>
<td><strong>Brazil</strong></td>
<td>23,434</td>
<td>54.62%</td>
<td>12,795</td>
</tr>
</tbody>
</table>
Over 2.5M Public Notebooks on GitHub

https://github.com/parente/nbestimate

# of Public Notebooks on GitHub

https://github.com/trending/jupyter-notebook

https://github.com/parente/nbestimate
Organizational Usage

We are seeing strong organizational adoption, driven by JupyterHub and other cloud based deployments.

- Data science platforms (Teradata, Google, Microsoft, IBM, AWS, Anaconda, Domino, CoCalc, Dataiku, data.world, Kaggle, ...)
- Data journalism (LA Times, Chicago Tribune, BuzzFeedNews, ...)
- Publishing (Springer, O’Reilly)
- K-12, University Education (Berkeley, Cal Poly, ...)
- Data Science/ML/Al Teams (1000’s)
- Large scale scientific collaborations (LSST, CERN, LIGO/VIRGO, PIMS, NASA JPL, Pangeo, ...)

... and 100s - 1000s more
An Amazing Community of Users
Example: LSST

- Large Synoptic Survey Telescope (https://www.lsst.org/)
- 27ft primary mirror
- 10 year operating period
- Each image covers 40 moons worth of the sky
- 15 TB of data every night!
- Computational platform based on JupyterHub + JupyterLab:
  - User base: “every astronomer on the planet” (~7,500)
  - “Next-to-the-data” analysis
  - Data access (3 PB Database, 4 PB files)
  - Scalable compute (2,400 cores)
  - Interactive analysis, modeling, simulation, visualization
  - Collaboration

https://www.slideshare.net/MarioJuric/what-to-expect-of-the-lsst-archive-the-lsst-science-platform
The foundation of Jupyter is a set of open standards for interactive computing.

Jupyter Notebook format ([https://github.com/jupyter/nbformat](https://github.com/jupyter/nbformat))

- JSON based document format for code, data, narrative text, equations, output
- Independent of user interface, programming language


- JSON based network protocol for interactive computing user interfaces (Jupyter Notebook) to talk to kernels that runs code interactively in a given programming language.
- Transport layer over ZeroMQ or WebSockets.

Jupyter Notebook Server ([https://github.com/jupyter/jupyter_server](https://github.com/jupyter/jupyter_server))

- A set of WebSocket and HTTP APIs for remote access to building blocks of interactive computing:
  - File system
  - Terminal
  - Kernels
Open-Source Software for Interactive Computing

- **Jupyter Notebook**: the original Jupyter notebook server and user interface.

- **JupyterLab**: next generation user interface for Jupyter notebooks.

- **JupyterHub**: deploy Jupyter to large organizations in a scalable, secure and maintainable manner.

- **IPython**: the Python kernel for Jupyter.

- **Jupyter Widgets**: interactive user interfaces within Jupyter notebooks.

- **nbconvert**: convert notebooks to other formats (HTML, Markdown, LaTeX).
Building Blocks for Interactive Computing

- Jupyter’s open standards and open-source software provides a set of building blocks that can be used to build a wide range of interactive computing systems.

- LEGO for interactive computing!

- Examples: JupyterLab, nteract, Google Colaboratory, Binder
JupyterLab is Jupyter’s next-generation user interface. It uses the same notebook format, server and network protocols.

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

\[
\begin{align*}
    x &= \sigma(y - x) \\
    y &= \rho x - y - xz \\
    z &= -\beta z + xy
\end{align*}
\]

Let’s call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

```python
from Lorenz import solve_lorenz
```

```
t, x_t = solve_lorenz(N=10)
```

nteract

nteract is an alternate user interface for working with Jupyter notebooks, focused on simplicity.

Open-source and sponsored by Netflix.

Uses the same notebook document format, server and network protocols.

https://nteract.io/
Google Colaboratory

Colaboratory is an alternate user interface for working with Jupyter notebooks, integrated with Google Drive.

Uses the same notebook format and network protocols.

https://colab.research.google.com/
Binder turns any Git repo with notebooks into a live notebook server for anyone in the world. It works with any Jupyter user interface and programming language (kernel).

https://mybinder.org/
Data Science in Large, Complex Organizations
Human Centered Design

• If you don’t design for humans, you will design for computers and humans will be miserable.

• Examples of such failures:
  
  • The primary “user interface” for working on a remote computer is still SSH
  
  • Tracebacks used to communicate to users when a program raises an exception

• See Alan Cooper’s “The Inmates Are Running the Asylum”

• Scientific computing and data science, are, by definition, human-centered activities that involve iterative exploration, analytical reasoning, visualization, mathematical abstraction, model building, moral and ethical reasoning, and decision making.

• In large organizations, there are a diverse range of individuals working with code and data: data scientists, data engineers, analytics, marketing, sales, product managers, university administrators, teachers, statisticians, etc.

• Not everyone who works with data wants or needs to write or look at code.
Collaboration is Essential

- Large organizations have complex human networks of people that need to work together.

- Individuals have different skill sets, responsibilities, access permissions, roles, priorities.

- Yet everyone needs to **look at** and **make decisions** based on the same overall data.

- GitHub is an effective collaboration tools only for people that live and breath code.
Datasets are Often Sensitive, Confidential

- The development of data science, ML/AI have been driven by open-source software and freely available, open, public datasets.

- However, most datasets of value to organizations are sensitive and confidential and require differing levels of protection.

- A range of different regulations: HIPAA, FERPA, GDPR, FedRAMP, Title 13, Title 26, SOX, GLBA, California Consumer Privacy Act, A.B. 375 (https://www.caprivacy.org/)

- Five Safes (Desai, Ritchie, Welpton 2016)
  - http://www2.uwe.ac.uk/faculties/BBS/Documents/1601.pdf
  - Framework for “‘designing, describing and evaluating access systems for data, used by data providers, data users, and regulators.”
  - Safe Projects, Safe People, Safe Data, Safe Settings, Safe Outputs

- Open-source tools can’t take a “not our problem” attitude.
  - Jupyter and other open-source tools were almost certainly used by Cambridge Analytica, SCLElections, to build models with Facebook user profiles for the 2016 US election.
How is Jupyter Tackling These Challenges?
JupyterLab

JupyterLab is the next-generation web-based user interface for Project Jupyter
• Next-generation user-interface for Project Jupyter

• Full support for Jupyter Notebooks

• Notebooks, terminals, text editor, file browser, code console

• Extension architecture enables anyone to add capabilities to JupyterLab using modern web technologies (npm, react,...)

• Integration between built-in components and extensions through public APIs

• Rich handling of different data types

• Ready for use! JupyterLab is now out of Beta.

• [http://jupyterlab.readthedocs.io/](http://jupyterlab.readthedocs.io/)

• Real-time collaboration on the way!
In this Notebook we explore the Lorenz system of differential equations:

\[
\begin{align*}
    \dot{x} &= \sigma(y - x) \\
    \dot{y} &= \rho x - y - xz \\
    \dot{z} &= -\beta z + xy
\end{align*}
\]

Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

```python
In [4]: from Lorenz import solve_lorenz
t, x, y = solve_lorenz(N=10)
```

```python
def solve_lorenz(k=10, max_time=4.0, sigma=10.0, beta=0.7, rho=28.0):
    """Plot a solution to the Lorenz differential equations."""
    fig = plt.figure()
    ax = fig.add_axes([0.1, 0.3, 0.8, 0.8], projection='3d')
    ax.set_xlim([-25, 25])
    ax.set_ylim([-35, 35])
    ax.set_zlim([5, 55])

    # prepare the axes limits
    ax.set_xlim([-25, 25])
    ax.set_ylim([-35, 35])
    ax.set_zlim([5, 55])

    x, y, z = x_t
    return x, y, z - x

def lorenz_deriv(x_y_z, t, sigma=sigma, beta=beta, rho=rho):
    """Compute the time-derivative of a Lorenz system."""
    x, y, z = x_y_z
    return (sigma*(y - x), x*(rho - z) - y, x*y - beta*z)

# Choose random starting points, uniformly distributed from -15 to 15
x0 = np.random.randn(15)
x3 = -15 + 30 * np.random.random((N, 3))
```
JupyterHub

Scaling interactive computing with Jupyter to organizations
In the Jupyter architecture, each user gets a dedicated Notebook/JupyterLab server, with containerized* compute and persistent* storage for files.

JupyterHub scales this model to multiple users and large organizations:

- Authenticator: extensible API for identifying and authenticating users (OAuth, LDAP, PAM,...)
- Spawner: extensible API for managing single user servers (subprocess, docker, kubernetes,...)
- Proxy: Dynamically map URLs to single user servers


*Usually, not required*
JupyterHub for Sensitive Data

- Organizational Data Model
  - Users, groups, roles, resources (compute, docker images, datasets,...)
  - Integration with directory services (Keycloak, Active Directory, LDAP), SAML, OIDC)

- Projects for JupyterHub
  - Shared workspace for text files, compute, Jupyter Notebooks
  - Well defined scope for collaboration and data access/security

- Telemetry and event logging
  - Needed for monitoring, auditing and compliance

- Reliable, Secure, Maintainable Deployments
  - Encryption in-transit and at-rest in the Jupyter architecture
  - Declarative, immutable, continuous deployments using Helm, Kubernetes

- With Julia Lane (NYU), Fernando Perez (Berkeley), funded by the Sloan and Schmidt Foundations.
Thank you!

Questions?