

April 24th,  
2019

Jupyter Team.

Presented by:  
Matthias Bussonnier

bussonniermatthias@gm  
ail.com

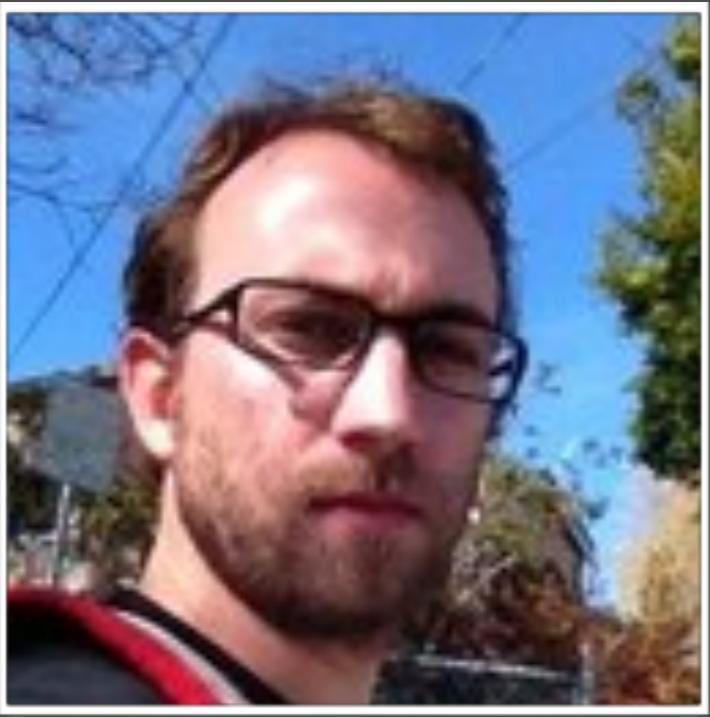
GitHub: @carreau  
Twitter: @mbussonn



Blue Waters Webinar

Slides <https://github.com/carreau/talks>

# Research Facilitator at UC Merced



Core-Dev and Founding  
Member of Jupyter



# IPython – 2001



IPython

```
$ ipython
Python 3.6.0
Type 'copyright', 'credits' or 'license' for more information
IPython 6.0.0.dev -- An enhanced Interactive Python. Type '?' for help.

In [1]: from string import hexdigits
....: from random import choice
....:
....: def randhex(length=10):
....:     return '0x'+''.join([choice(hexdigits) for x in range(10)]).l
ljust
lower
lstrip
```

(BTW, IPython is uppercase I)

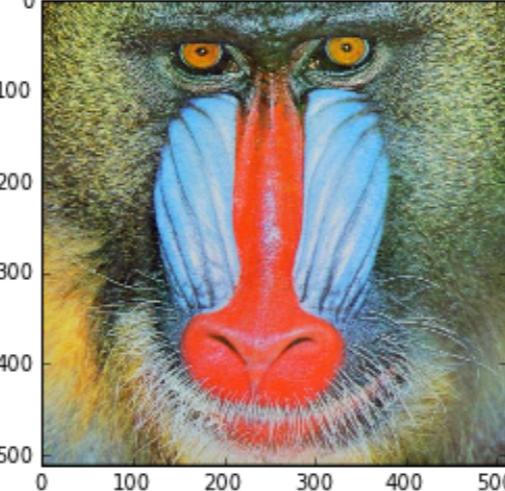


# QtConsole 2010-2011

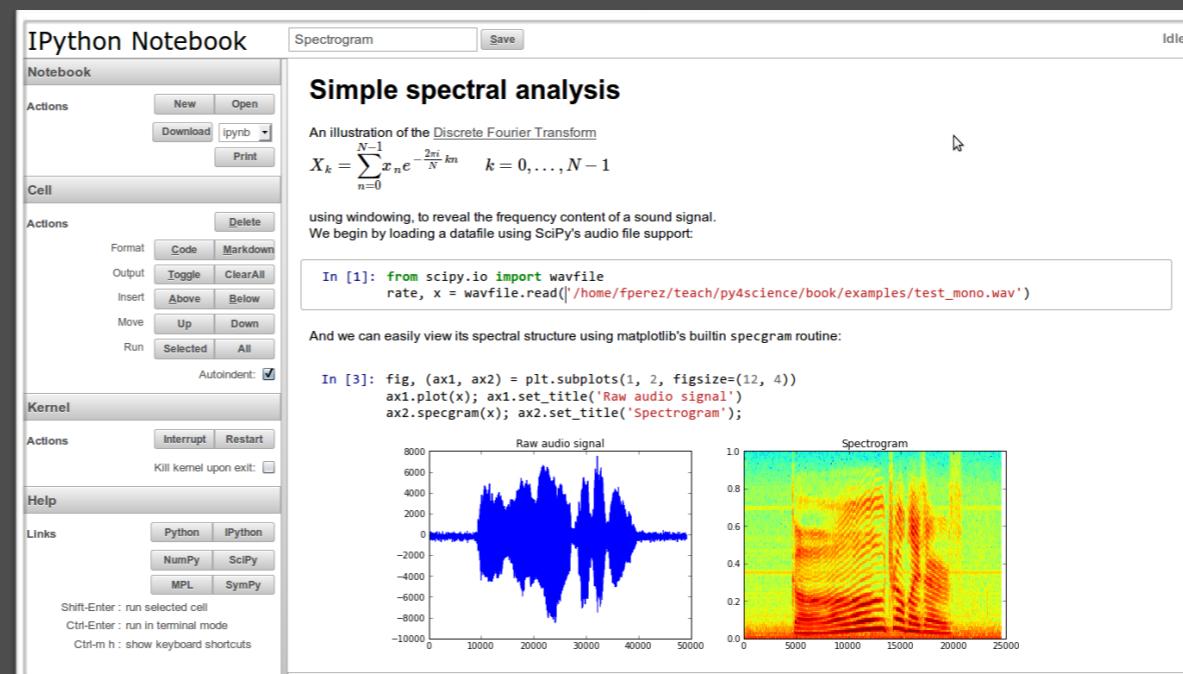
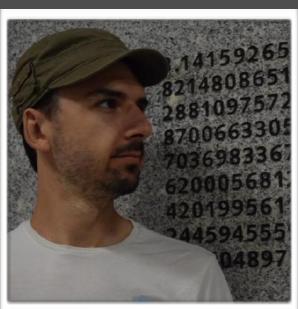


```
IPython 0.11 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
%guiref   -> A brief reference about the graphical user interface.

In [1]: imshow(imread("baboon.png"))
Out[1]: <matplotlib.image.AxesImage at 0x9fe274c>

  
In [2]:
```

# The IPython Notebook – 2012



# Jupyter – 2014

Renames the Python Agnostic Part to “Jupyter” – an homage to Galileo first Notebooks.



jupyter Welcome to P

This Notebook Server was created by

**WARNING**  
Don't rely on this server

Your server is hosted there

Run some Python code

To run the code below:

1. Click on the cell to select it
2. Press SHIFT+ENTER

A full tutorial for using the Jupyter Notebook is available at

In [ ]: `#matplotlib inline`  
`import pandas as pd`  
`import numpy as np`  
`import matplotlib`

jupyter Lorenz Differential Equations (autosaved)

File Edit View Insert Cell Kernel Help

Python 3

### Exploring the Lorenz System

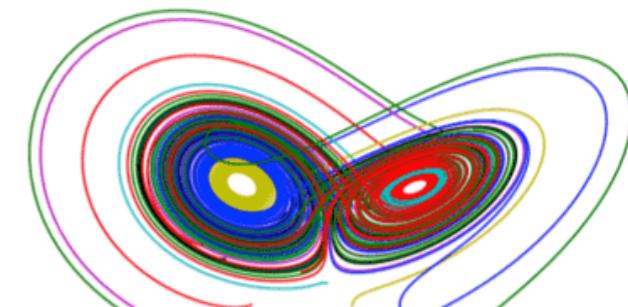
In this Notebook we explore the [Lorenz system](#) of differential equations:

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

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters ( $\sigma$ ,  $\beta$ ,  $\rho$ ) 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.), sigma=(0.0,50.0), beta=(0.,5), rho=(0.0,50.0))`

angle max\_time sigma beta rho



# Jupyter: 2019



1000+ Contributors

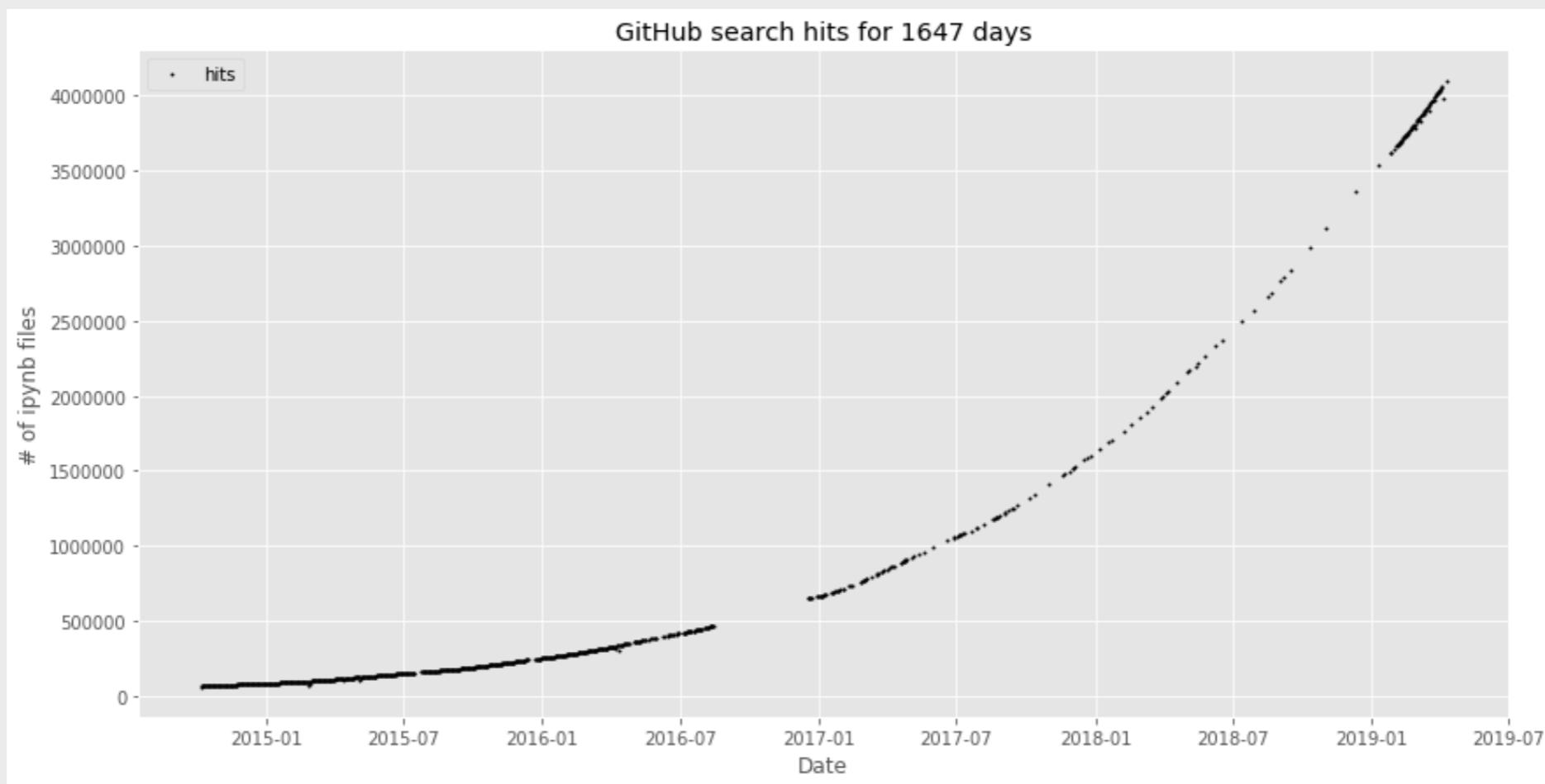
Dozens of Projects

2017 ACM System Software Award

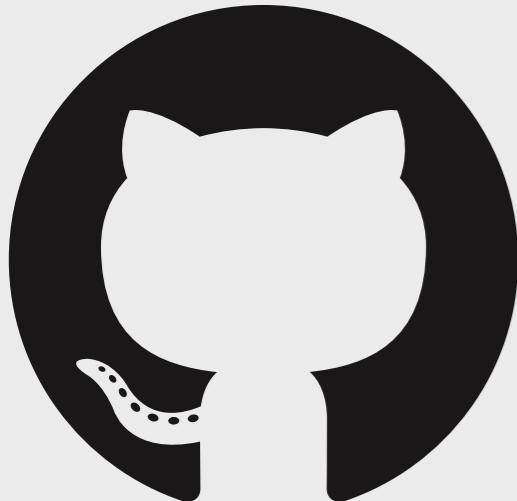


# A few Numbers

~4M on GitHub



<https://github.com/parente/nbestimate>



150+ repositories across multiple organizations  
(IPython, Jupyter, JupyterHub, JupyterLab, ...)  
at 2 release/year that's ~ 1 release per day

1000+ Contributors

8+ Millions Users,  
(with conservative estimates)

Worldwide ~21M developers – North America ~4.4M  
VS Code ~2.6 M Active Users  
GitHub 24M Users

# Core Contributors



- 1000+ Open source contributors, Majority Volunteer
- Organization with Open Governance (currently restructuring)



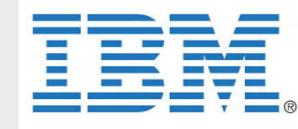
# Sponsors



ALFRED P. SLOAN  
FOUNDATION



SIMONS FOUNDATION



Bloomberg



# How Jupyter came to be



# Life cycle of a Scientific Idea

- Individual exploratory work (Repl, Scripts)
- Collaborative development (Dropbox/ Google Doc / emails / git )
- Parallel production runs (MPI, rewrite C++, batch jobs)
- Publication & communication (Word, Latex, ppt...)
- Education
- Goto 1



# Tools Overhead

Each Tool brings (cognitive) overhead, time to install, deploy, and master.

Can we create a (set of) tools, with minimal overhead end enough flexibility ?

Parallel with popular DataScience languages

Fortran/C/C++ are fast, but take significant development time and skills

Python/R/Julia are (usually) slower, but are useful immediately.



# Rise of Jupyter

- An increasing number of discipline have a fast growing amount of data
- Technology is **a tool** that should
  - Empower **the User**
  - Amplify **Domain Knowledge and Expertise**
  - Facilitate **Sharing and Collaboration**

Jupyter provide a framework that can be use in all the step in the cycle of a scientific idea

- BSD Licensed (Free to use and redistribute even Commercially)
- Open Source, Community Maintained
- Important for sustainability, diversity, and equal access

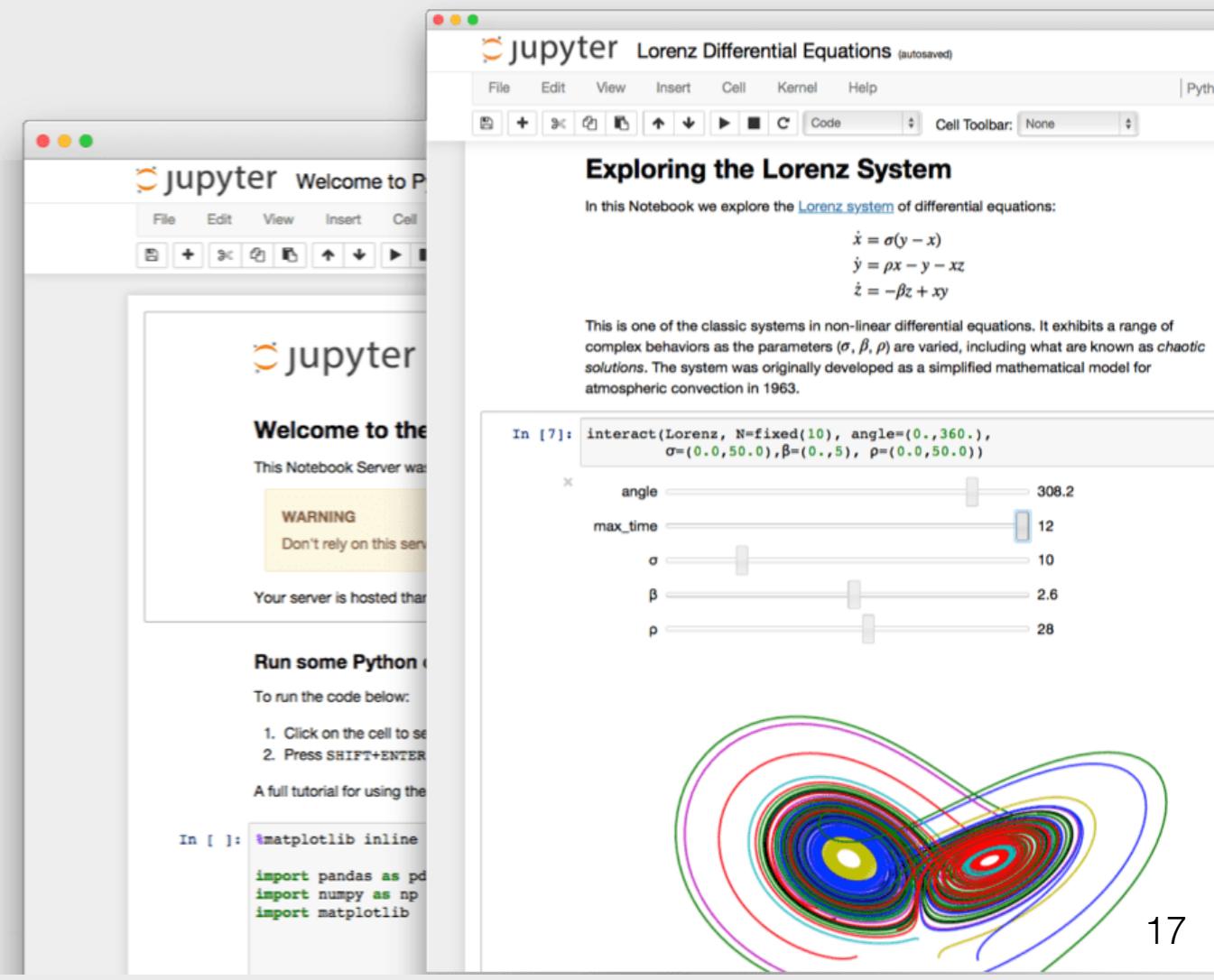
# Life cycle of a Scientific Idea in 2020

- **Exploratory work:** not "small" anymore.
- **Collaboration:** a rich, dynamic network.
- **Scholarly output:** new and diverse types.
- **Consumers of output:** from academia and education to decision making and the public.



# What is Jupyter

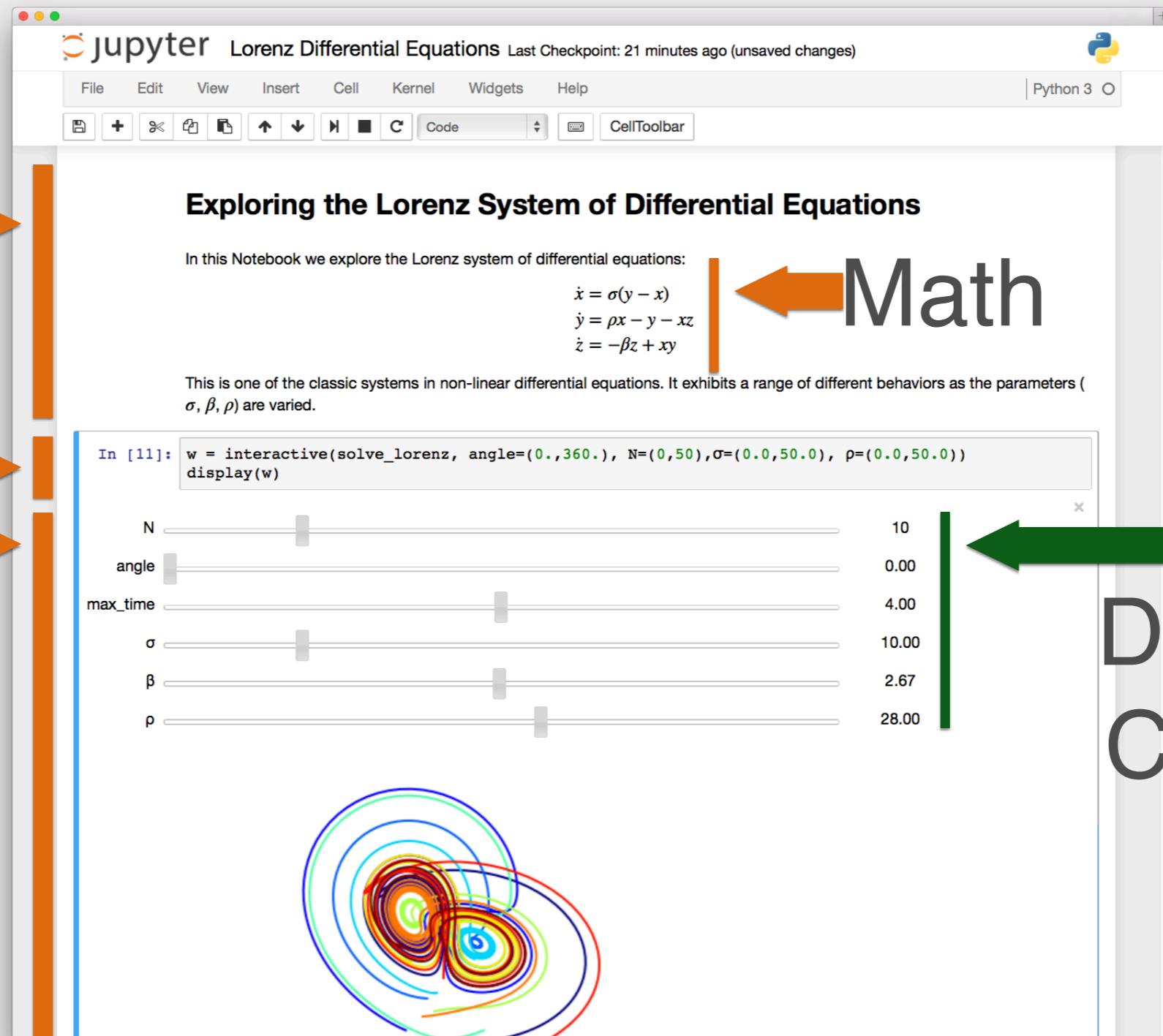
- Mainly Known for The Notebook
  - Web server, a web app, containing code, narrative, math and results.
  - Attached to a Kernel doing computation.
- Results can be:
  - Static
  - Interactive (client side)
  - Dynamic (trigger compute)



# Narrative



# Code Result



# Math

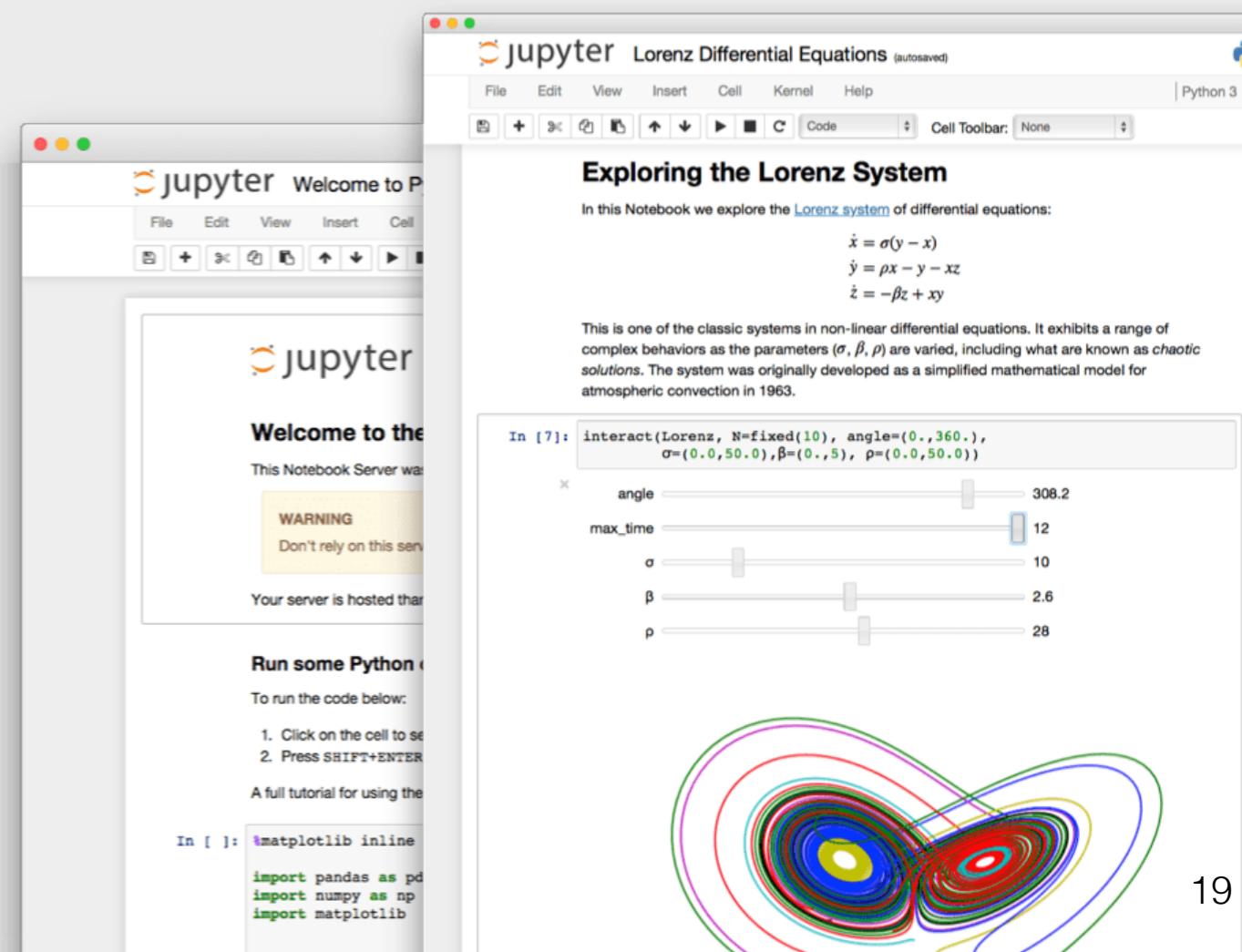
# Dynamic Controls

aka “widgets”

# Web Based Notebook Application

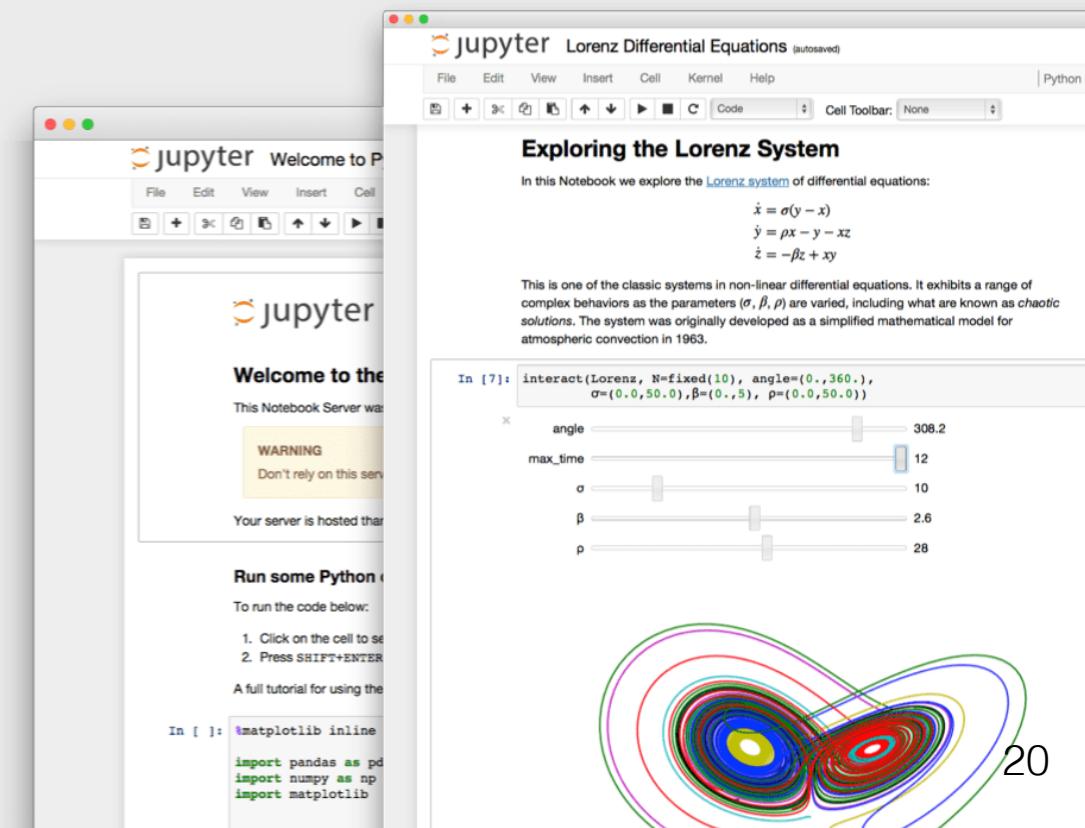
- Web technologies are accessible.
  - Only need a web browser to control an HPC Cluster
  - Familiar to users
- Rapid increase in performance and functionality
  - V8, 3D, Wasm, ...

- Identical for local and remote use.
- Allow multiple domain collaboration



# Open Notebook Document Format

- Notebooks get saved as JSON documents, which contain narrative, code, and results
  - Ubiquitous, JSON is readable in ~all languages.
  - Result embedding ensure trust (no Copy Past errors)
  - Make it easy to share and modify (Nbviewer, Binder)



# JupyterLab

The image shows the JupyterLab interface, a modern web-based development environment for data science. It features a central workspace with several tabs open:

- color\_scatterpk**: A notebook tab showing code for generating a scatter plot with colored points. The output cell shows the plot, which is a dense cloud of points with a color gradient from purple to yellow.
- Terminal 1**: A terminal window showing a bash shell with the command `output\_notebook()` run, resulting in the message "BokehJS successfully loaded".
- transit.ipynb**: A notebook tab showing code for plotting passenger load at a specific stop. The output cell shows a histogram titled "Passenger Load at Rosengartenstrasse stop" with the x-axis labeled "Number of passengers" and the y-axis labeled "Frequency".
- routes.json**: A file viewer tab showing a map of a city area with blue lines representing routes. Below it is a JSON object for a specific route stop.
- stops.json**: A file viewer tab showing a list of stops with their IDs, names, and coordinates.
- passenger.csv**: A file viewer tab showing a table of passenger data with columns for stop ID, stop name short, and stop name.

On the left, there is a sidebar with navigation links and a file browser showing a list of files and notebooks. On the right, there is a large "lab" logo.

<https://blog.jupyter.org/jupyterlab-is-ready-for-users-5a6f039b8906>

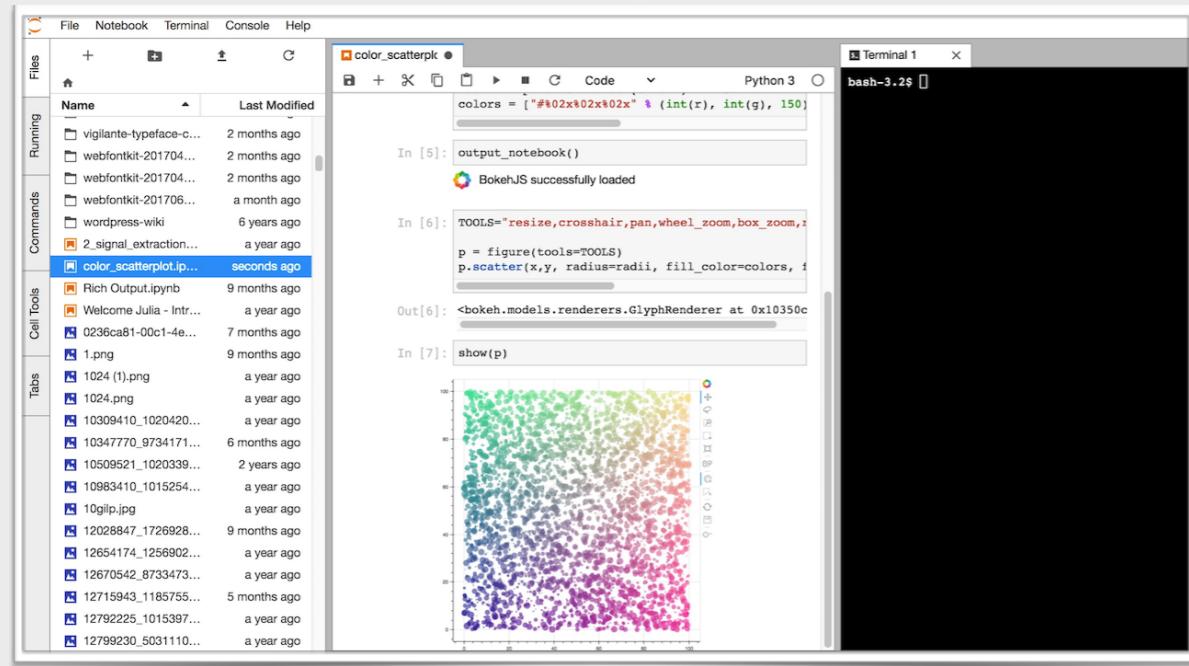
# JupyterLab

- Install Side by Side with Classic Notebook



- No Change in File Format, or protocol

- Better Architecture (all extensions are first class)



Bojan Marković  
Feb 20

You'll only take Spyder from my cold, dead... Ooooh, pretty shiny colors, inline graphics.. Does it come in fuchsia? :)

- Classic Notebook will be deprecated at some point

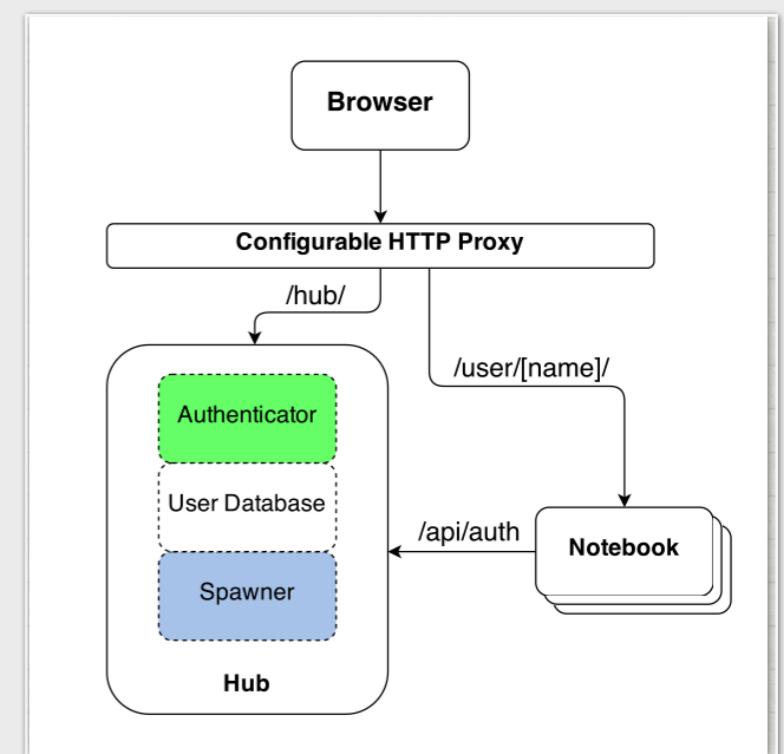
# Many languages



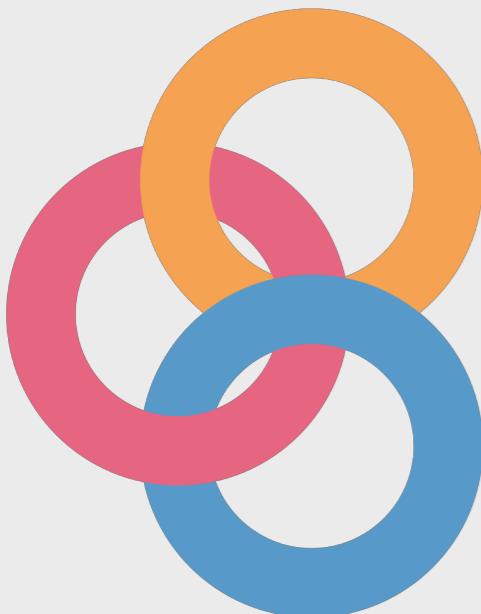
# Easy, Scalable Deployment

# JupyterHub

- A notebook application is a Single User application
- Quick and easy multi-user deployments are critical to lower overhead.
- JupyterHub Provides way a simple way to deploy Jupyter at scale.
  - <https://z2jh.jupyter.org/> for a guide.



# Binder (And MyBinder.org)



- Technology which takes any GitHub repository with Jupyter notebooks
- Turn it into a Docker image to ensure reproducibility and quick deployment.
- Starts an isolated, ephemeral server in a few seconds, for user to interact with.

\* Not limited to GitHub, Notebooks, Jupyter, Docker, or Ephemeral

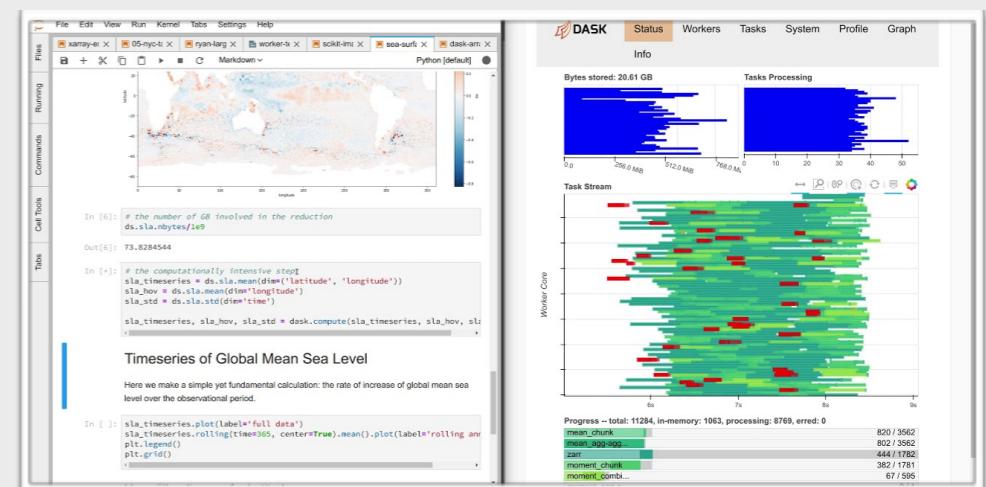
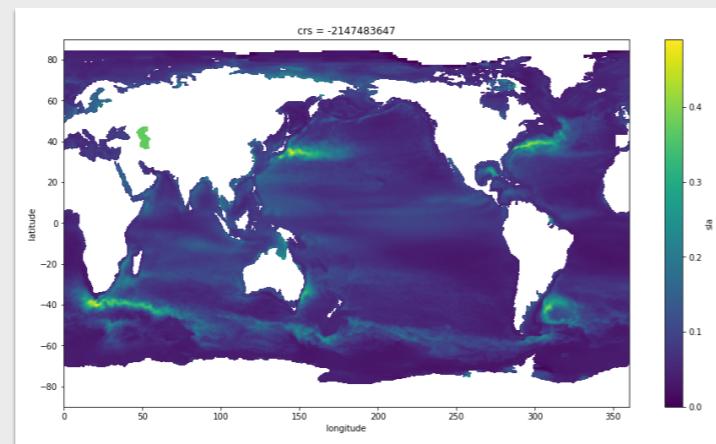


# In the Cloud



**PANGE**O  
<http://pangeo-data.org/>

1. Foster collaboration around the open source scientific python ecosystem for ocean / atmosphere / land / climate science.
2. Support the development with domain-specific geoscience packages.
3. Improve scalability of these tools to handle petabyte-scale datasets on HPC and cloud platforms.



# In the Cloud

- Completely managed JupyterHub on

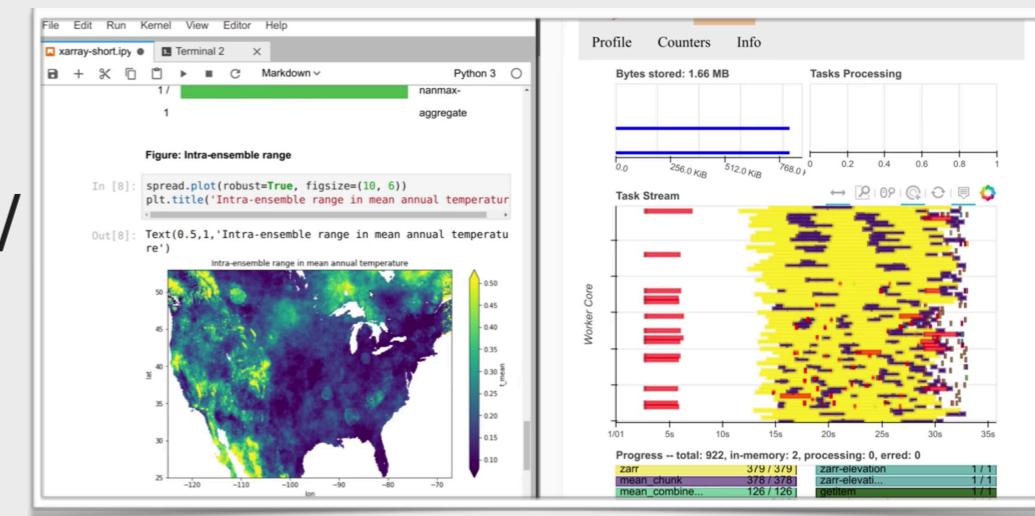
## Kubernetes



**PANGE**O

<http://pangeo-data.org/>

- <http://pangeo.pydata.org/>
- Login via GitHub
- Customized for GeoScience
- Persisting servers on Google Cloud,
- Large amount of Ram/CPU/Nodes
- Dynamically scalable



# Future

What will You build ?

Questions ?

