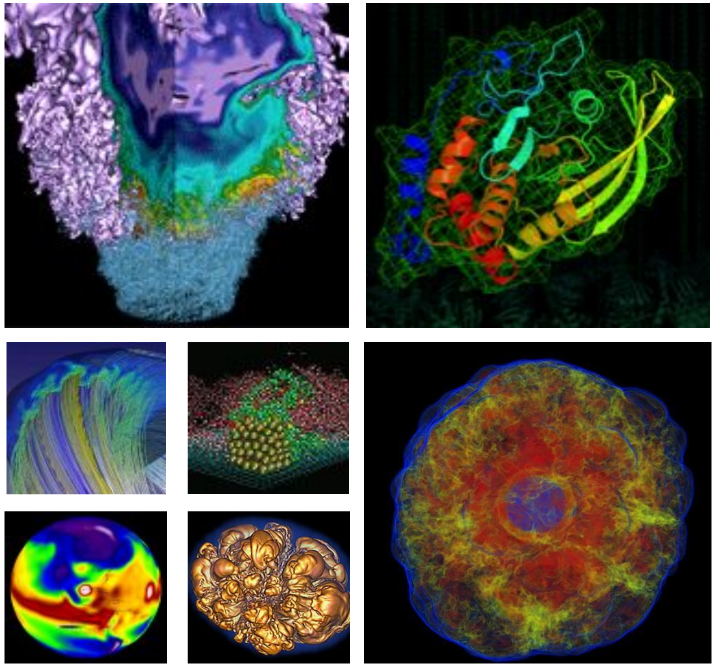
Jupyter at NERSC

Redefining the Interface to HPC



Rollin Thomas

Data and Analytics Services

NERSC User Group Meeting

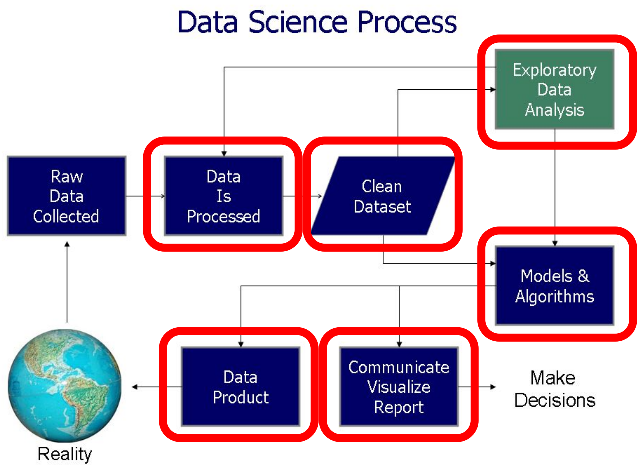
Rockville MD • 2019-07-19

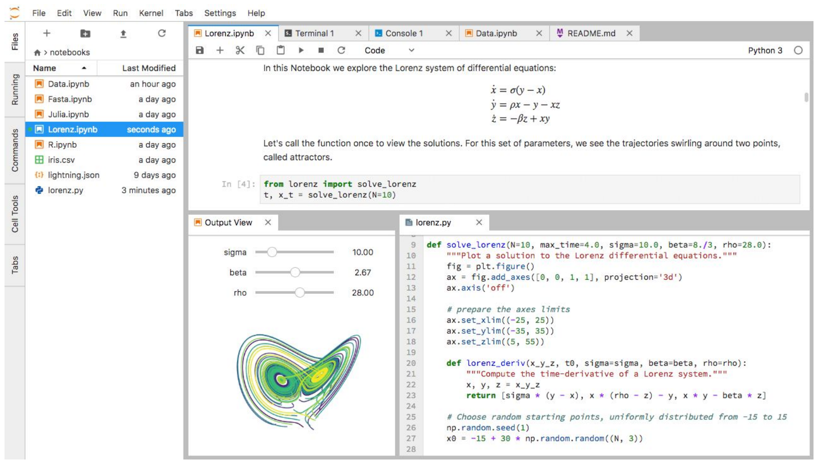
What is Jupyter?

Tool for reproducible, shareable narratives, literate computing:

***Notebook*: Document containing code, comments, outputs.**

Rich text, interactive plots, equations, widgets, etc.





notebooks

Name	Last Modified
Data.ipynb	an hour ago
Fasta.ipynb	a day ago
Julia.ipynb	a day ago
Lorenz.ipynb	seconds ago
R.ipynb	a day ago
iris.csv	a day ago
lightning.json	9 days ago
lorenz.py	3 minutes ago

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

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

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.

```
In [4]: from lorenz import solve_lorenz
        t, x_t = solve_lorenz(N=10)
```

sigma	10.00
beta	2.67
rho	28.00

```
def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential equations."""
    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1], projection='3d')
    ax.axis('off')

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

    def lorenz_deriv(x_y_z, t0, 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
    np.random.seed(1)
    x0 = -15 + 30 * np.random.random((N, 3))

```

Goal: Enable exploratory data analytics, deep learning, workflows, and more through Jupyter on NERSC systems.

Why Jupyter, Why Now at NERSC?

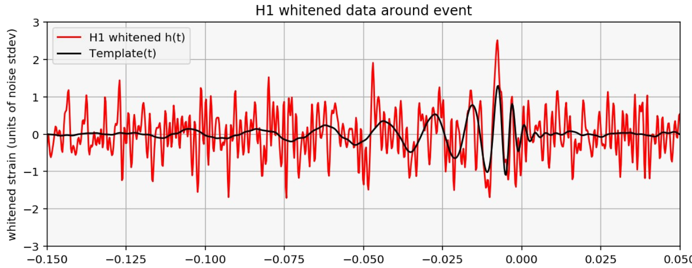
Integral part of Big (Data) Science & Superfacility:
- LSST-DESC, DESI, ALS, LCLS, Materials Project
- NCEM, LUX/LZ, KBase...

Generational shift in analytics for science + more:
- UCB's Data Science 8 course, entirely in Jupyter
- "I'll send you a copy of my notebook"
- Training events adopting notebooks (DL)

Supporting reproducibility and science outreach:
- Open source code and open source science
- Jupyter notebooks alongside publications (LIGO)

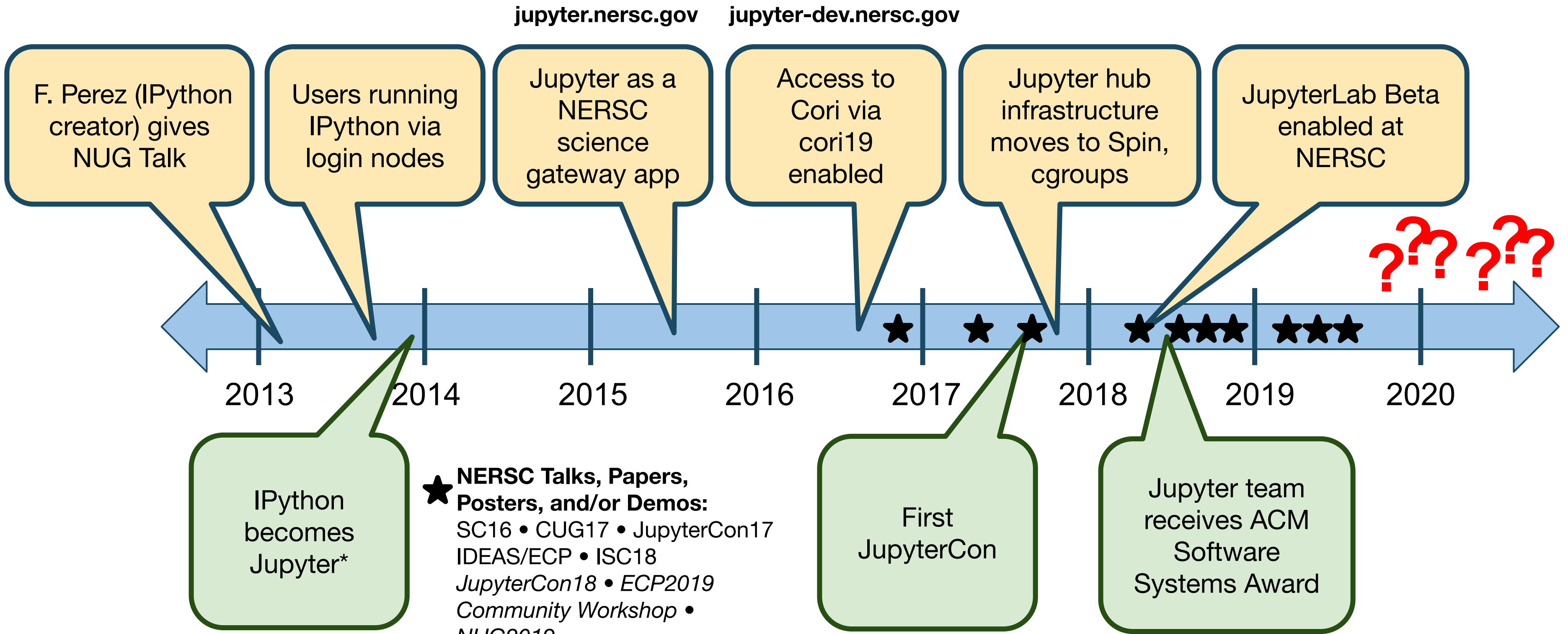
Data 8: Foundations of Data Science, Fall 2018, Zellerbach Hall

2017 ACM Software System Award: "… ***a de facto standard for data analysis in research, education, journalism and industry.*** Jupyter has broad impact across domains and use cases. Today more than ***2,000,000 Jupyter notebooks are on GitHub***, each a distinct instance of a Jupyter application—covering a range of uses from technical documentation to course materials, books and academic publications."



LIGO Binary BH-BH Merger GW Signature
Figure from LIGO EPO/Publication Jupyter Notebook

Jupyter at NERSC Timeline


jupyter.nersc.gov
jupyter-dev.nersc.gov
F. Perez (IPython creator) gives NUG Talk
Users running IPython via login nodes
Jupyter as a NERSC science gateway app
Access to Cori via cori19 enabled
Jupyter hub infrastructure moves to Spin, cgroups
JupyterLab Beta enabled at NERSC
????
2013
2014
2015
2016
2017
2018
2019
2020
IPython becomes Jupyter*
First JupyterCon
Jupyter team receives ACM Software Systems Award

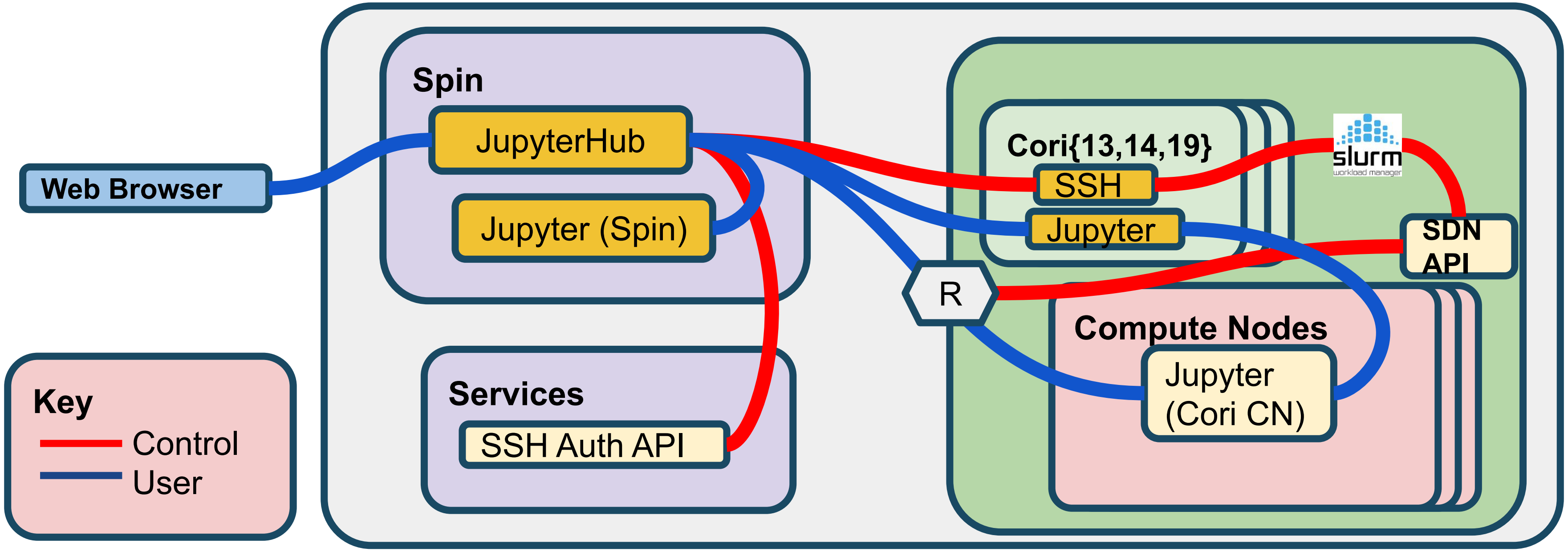
★ **NERSC Talks, Papers, Posters, and/or Demos:**
SC16 • CUG17 • JupyterCon17
IDEAS/ECP • ISC18
JupyterCon18 • ECP2019 Community Workshop • NUG2019

* IPython became Jupyter, de-emphasizing the Python branding.
Jupyter is language-agnostic.

Use Cases & Access Modes @ NERSC

Use Case	Where	Why
Light-weight data analysis and visualization	Spin Container *(In production now.)*	Usable when other systems are down. Simple, interactive access
Workflow execution and medium-scale data analysis	Cori "Login" Nodes *(In production now.)*	Access to batch and scratch Larger memory shared node
Heavy weight computation including task frameworks	Cori Compute Nodes *(In testing now.)*	Dedicated resources (e.g. memory and cores). Ability to launch parallel workloads in the notebook.

Jupyter @ NERSC Architecture



Jupyter Matters to NERSC Users

Users appreciate Jupyter @ NERSC…

[Venkitesh: "… jupyter notebooks are very important for me: ***The 3 most important things in life: food, shelter and jupyter… everything else is optional.***"]

"I really like the jupyter interface."

"New jupyter notebooks are awesome!"

"Great interactive workflow (e.g. for postprocessing) via JupyterHub"

"As mentioned, the ability to access data from the scratch directories through the Jupyter hub is very important to my workflow. The Jupyter hub has been running more and more consistently, but it still seems to lag or stall sometimes. I guess ***my only thought on how to improve (currently)*** would be to improve the stability of the Jupyter hub."

"I absolutely love the fact that I can use the Jupyter hub to access the Cori scratch directory. This allows me to analyze data through the browser … or to quickly check that simulation runs are going as expected without having to transfer data to a different location. ***I actually also have access to other supercomputer clusters, but this is one of the biggest reasons I mainly use Cori and Edison for debugging and production runs.***"

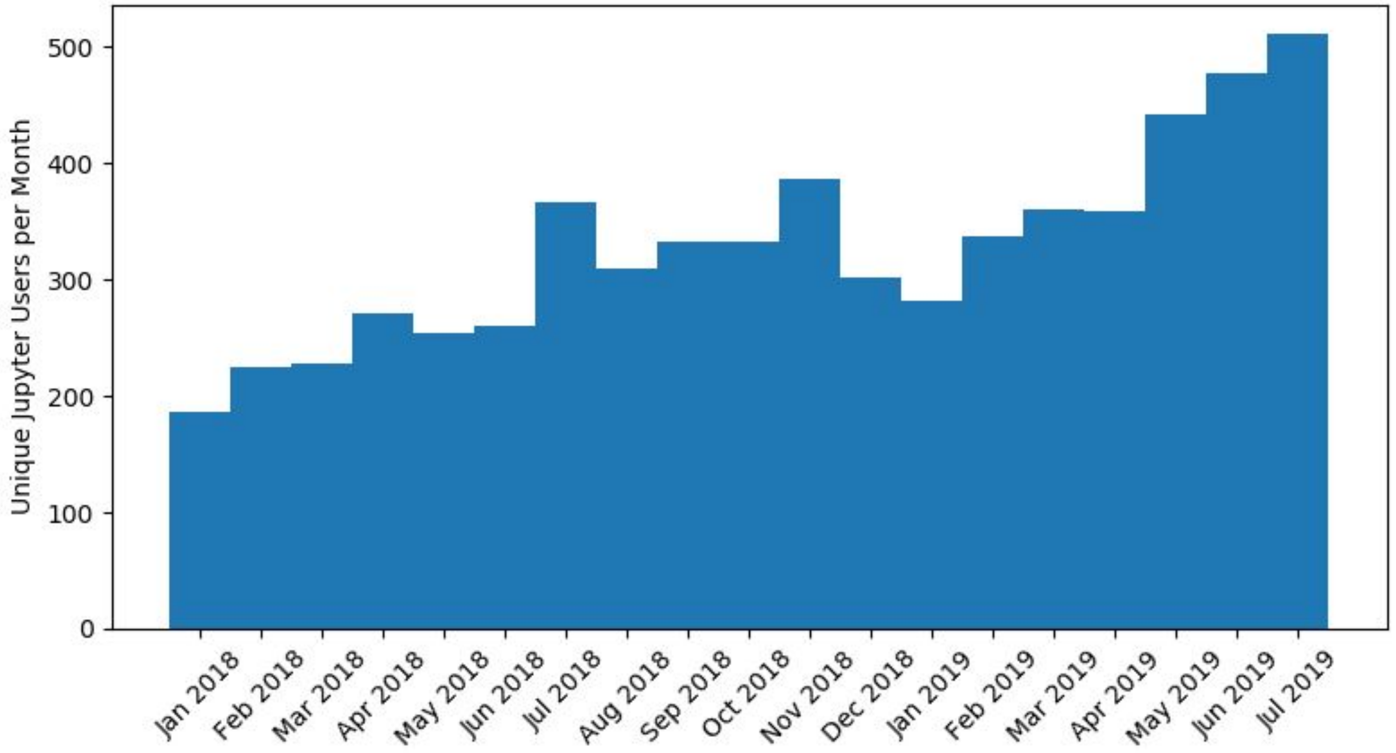
…but need increased stability and to scale up.

(2017 User Survey)

"I would really appreciate it if jupyter.nersc.gov wouldn't go down as much as it does."

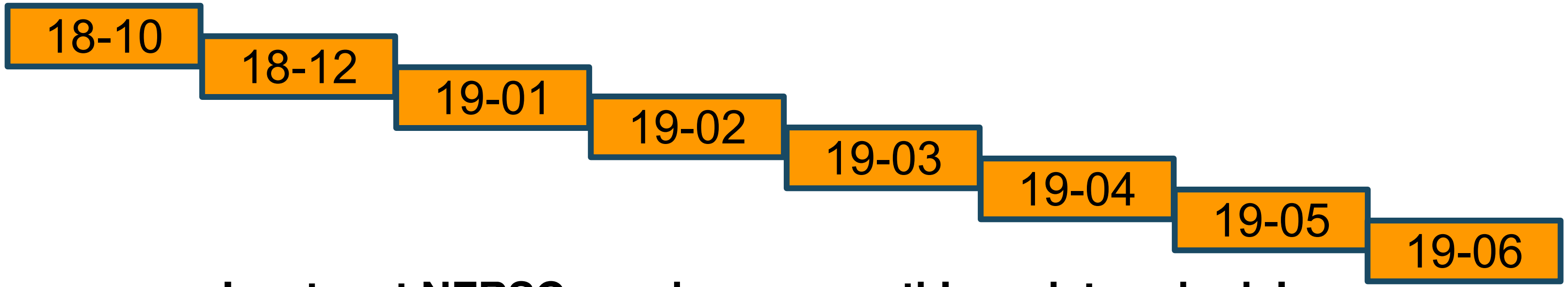
"MPI cannot be used in jupyter notebook as well, where the jupyter hubs run on login nodes (unless when using the compute nodes through SLURM.)"

Jupyter on Cori Usage Numbers

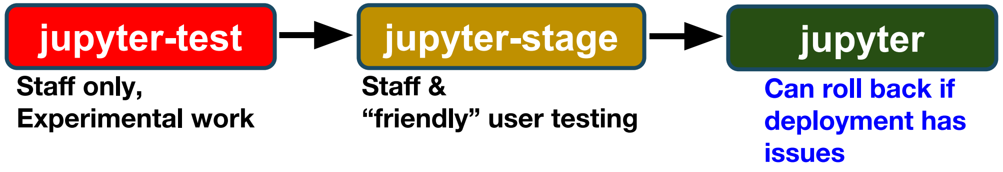


Supporting these users, their current and future needs is a lot of work!

Deployment Schedule

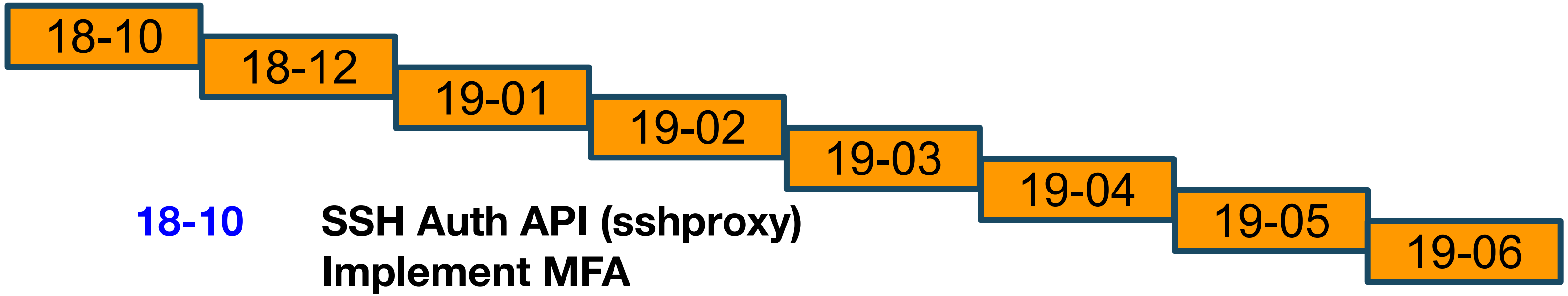


Jupyter at NERSC now has a ~monthly update schedule
Usually follows Cori maintenances by 2-5 days
We post an announcements when it's coming up
Jupyter upgrades themselves are very short (usually…)
Rarely change notebook env, mostly hub changes:



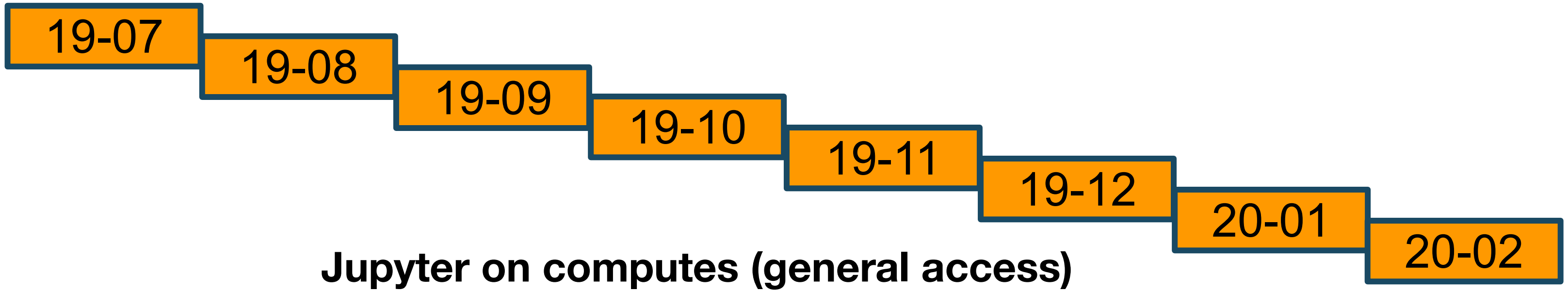
Brought to you by Spin

Deployment Schedule



18-10	**SSH Auth API (sshproxy)** **Implement MFA**
18-12	**Add “offline” communication page** **“Announcements” service for communication**
19-02	**Ability to leverage more Cori nodes**
19-03	**Converged hubs: *jupyter-dev* and *jupyter***
19-05	**Computes and GPU node access (special access)** **Leverage Iris for role-based access control** **JupyterHub 1.0**

Deployment Schedule



Jupyter on computes (general access)

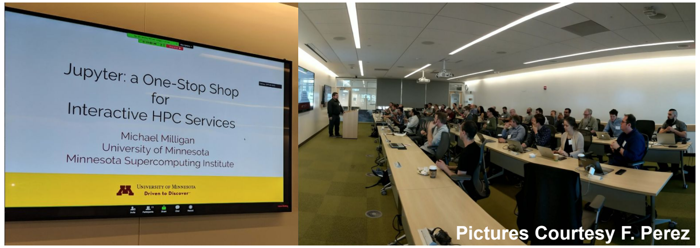
Configurable jobs:

- **Reservation (for training/tutorials/workshops)**
- **Full shifter image support**

Integration with NERSC SSO

Jupyter Slurm integration
Parameterized notebooks
Managing curated notebooks
...

Workshop: Jupyter for Science User Facilities and High Performance Computing


Jupyter: a One-Stop Shop for Interactive HPC Services
Michael Milligan
University of Minnesota
Minnesota Supercomputing Institute
Pictures Courtesy F. Perez

Joint Workshop w/BIDS: June 11-12 at NERSC, June 13 at BIDS

Committee: Rollin Thomas • Shane Canon • Shreyas Cholia • Kelly Rowland
Debbie Bard • Dan Allan (BNL) • Chris Holdgraf (BIDS)

Part of "Jupyter Community Workshop" Series

Competitive application process • Granted up to $20K for travel support
Funds from Bloomberg, managed by NumFOCUS and Project Jupyter

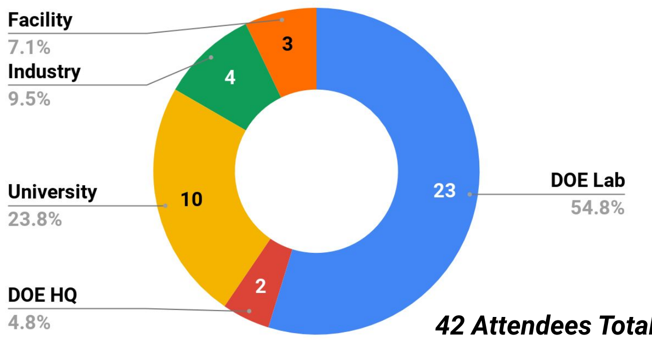
User Facilities, HPC & Data Centers Represented

NSLS-II • LSST • APS • SLAC • JGI • ARM • European XFEL
NERSC • ALCF • TACC • MSI (@UMN) • Compute Canada • ESA

Content

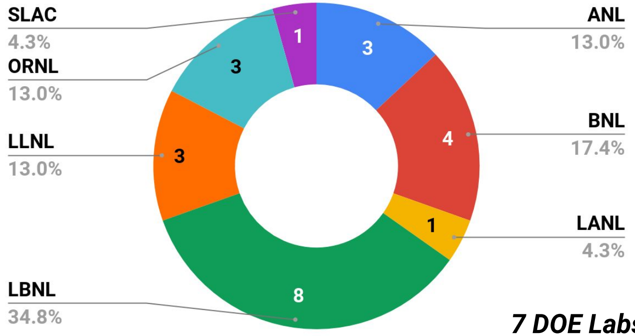
Talks: Deployment • Infrastructure • Extending Jupyter for HPC • Use Cases
Breakouts: Organizing Collaboration • Securing Jupyter • Sharing Notebooks
Reproducibility • Best Practices • Future Plans • Tutorials
Roundtable Meeting with Core Jupyter Developers

Attendance by Category



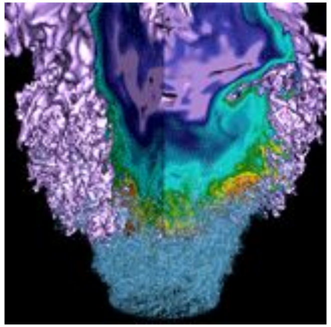
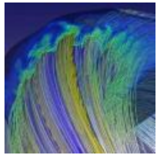
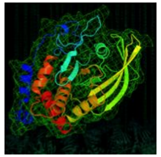
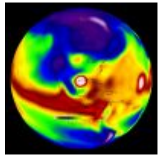
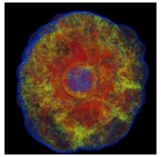
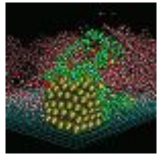
42 Attendees Total
(Facility = Non-University Facility)

DOE Lab Representation



7 DOE Labs

Demo Intro

NCEM Superfacility Highlight Slide

Slide Courtesy Matt Henderson and Shreyas Cholia (LBL CRD)

Scientific Achievement

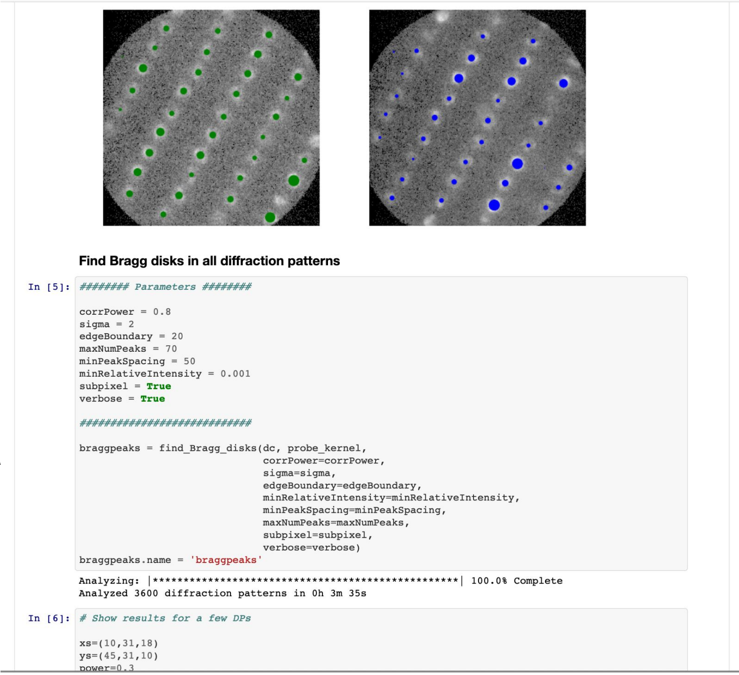
Enable interactive human-in the loop computing for Superfacility Workflows at NERSC

Significance and Impact

Superfacility Projects like ALS and NCEM need a flexible interactive user interface to be able to analyze data from experiments in real time. We use Jupyter Notebooks to enable these projects to combine live code execution, reproducibility, interactive visualizations on HPC resources.

Research Details

- Command and control center for integrating distributed workflows under a common platform
- Enable HPC Tools to allow Jupyter notebooks to execute code in parallel on distributed backend resources using Dask/IPyParallel/MPI
- Realtime rendering of results rendered inline in Jupyter to enable live interaction, visualization and job steering
- Parameterized notebooks can to execute a workflow over multiple datasets and parameters.
- Enable reproducible workflows through project curated notebooks that can be cloned / executed



Find Bragg disks in all diffraction patterns

```
In [5]: ######## Parameters ########

        corrPower = 0.8
        sigma = 2
        edgeBoundary = 20
        maxNumPeaks = 70
        minPeakSpacing = 50
        minRelativeIntensity = 0.001
        subpixel = True
        verbose = True

        ############################

        braggpeaks = find_Bragg_disks(dc, probe_kernel,
                                      corrPower=corrPower,
                                      sigma=sigma,
                                      edgeBoundary=edgeBoundary,
                                      minRelativeIntensity=minRelativeIntensity,
                                      minPeakSpacing=minPeakSpacing,
                                      maxNumPeaks=maxNumPeaks,
                                      subpixel=subpixel,
                                      verbose=verbose)
        braggpeaks.name = 'braggpeaks'

        Analyzing: |**************************************************| 100.0% Complete
        Analyzed 3600 diffraction patterns in 0h 3m 35s

In [6]: # Show results for a few DPs

        xs=(10,31,18)
        ys=(45,31,10)
        power=0.3
```

Achieved 90x speedup on NCEM Py4DSTEM Notebook running on HPC resources at NERSC: https://github.com/py4dstem/py4DSTEM/blob

Team

Rollin Thomas

Shane Canon

Kelly Rowland

Trevor Slaton

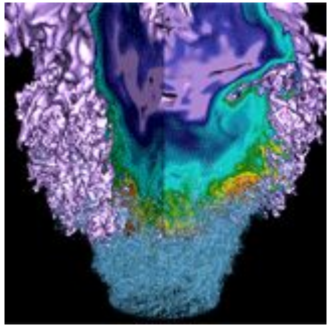
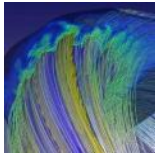
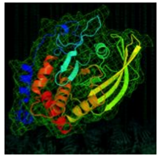
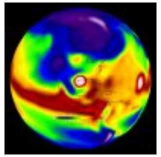
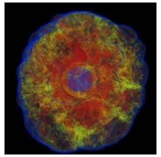
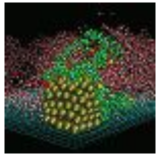
Shreyas Cholia

Matt Henderson

William Krinsman

Jon Hays

Demo

Future Work

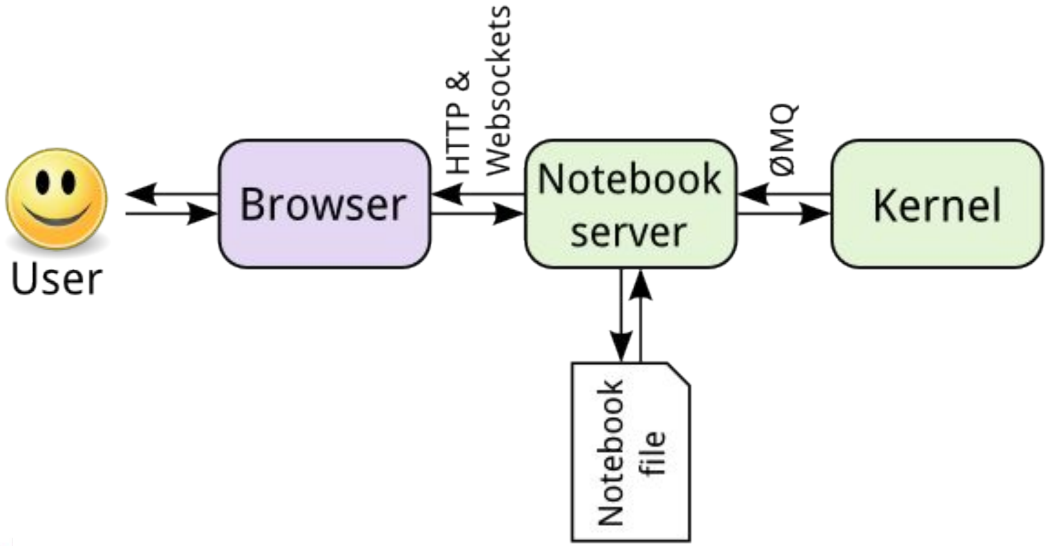
- Streamline notebook ↔ running job interaction
- Exclusive node access (notebook on compute) in real-time-like queue
- Experiment with configurable-http-proxy for dynamic routing (not SDN)
- Support >300 users/day, maybe add 1-2 more shared nodes, manage via Slurm
- Deploy Jupyter solutions at NERSC:
  - Sharing notebooks, parameterized notebooks, curated notebooks
  - Voila and dashboards
  - More HPC-centric extensions and fixes to JupyterLab/Hub.
- Engagement with Jupyter and Jupyter-in-HPC communities
- User engagements through Superfacility and beyond:

  NCEM, ALS, DESI, LSST, …

Thank You

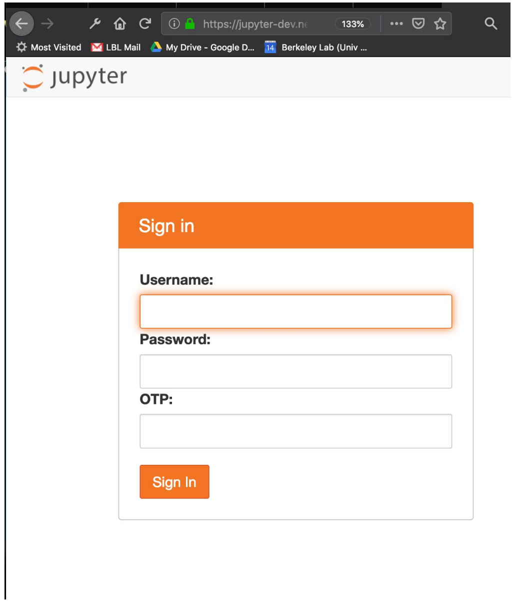
How It Works

The **Notebook Server** sends code (via **ZeroMQ**) to a language "**kernel**" that executes this code. In addition to running your code, it stores code and output, together with markdown notes, in an editable document called a **notebook**, saved as a **JSON** file with a **.ipynb extension**.



JupyterHub: Jupyter as a Service

- Service to deploy notebooks in a multi-user environment
- Manages user authentication, notebook deployment and web proxies



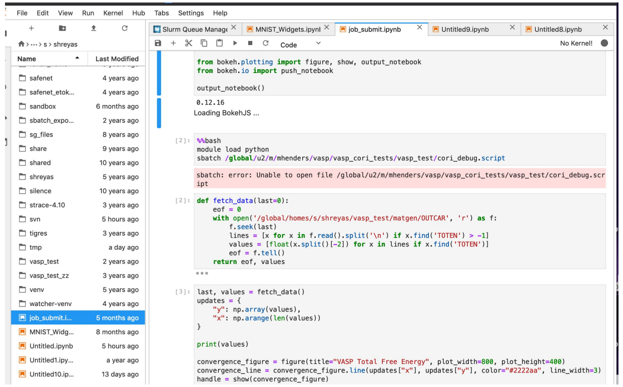
Sign in

Username:

Password:

OTP:

Sign In



Name	Last Modified
safenet	4 years ago
safenet_etok...	4 years ago
sandbox	6 months ago
sbatch_expo...	2 years ago
sg_files	8 years ago
share	9 years ago
shared	10 years ago
shreyas	5 years ago
silence	10 years ago
strace-4.10	3 years ago
svn	5 hours ago
tigres	3 years ago
tmp	a day ago
vasp_test	2 years ago
vasp_test_zz	3 years ago
venv	5 years ago
watcher-venv	4 years ago
job_submit.i...	5 months ago
MNIST_Widg...	8 months ago
Untitled.ipynb	5 hours ago
Untitled1.ipy...	a year ago
Untitled10.ip...	13 days ago

```
from bokeh.plotting import figure, show, output_notebook
from bokeh.io import push_notebook

output_notebook()
```

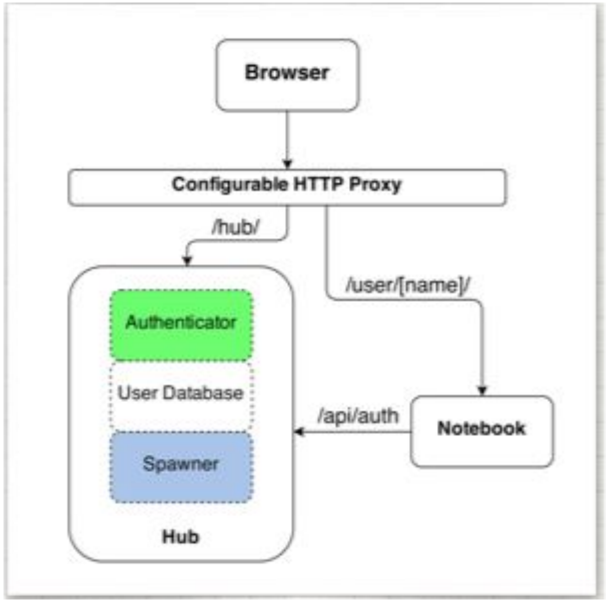
```
0.12.16
Loading BokehJS ...
```

```
[2]: %%bash
module load python
sbatch /global/u2/m/mhenders/vasp/vasp_cori_tests/vasp_test/cori_debug.script
```

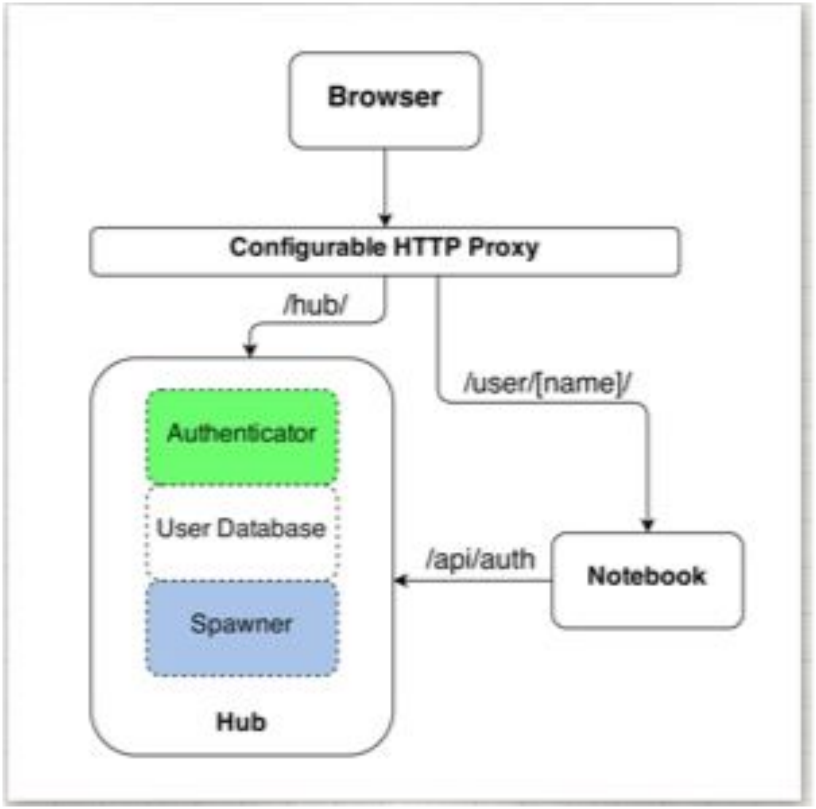
```
sbatch: error: Unable to open file /global/u2/m/mhenders/vasp/vasp_cori_tests/vasp_test/cori_debug.scr
ipt
```

```
[2]: def fetch_data(last=0):
    eof = 0
    with open('/global/homes/s/shreyas/vasp_test/matgen/OUTCAR', 'r') as f:
        f.seek(last)
        lines = [x for x in f.read().split('\n') if x.find('TOTEN') > -1]
        values = [float(x.split()[-2]) for x in lines if x.find('TOTEN')]
        eof = f.tell()
    return eof, values
```

```
[3]: last, values = fetch_data()
updates = {
    "y": np.array(values),
    "x": np.arange(len(values))
}

print(values)

convergence_figure = figure(title="VASP Total Free Energy", plot_width=800, plot_height=400)
convergence_line = convergence_figure.line(updates["x"], updates["y"], color="#2222aa", line_width=3)
handle = show(convergence_figure)
```



JupyterHub Architecture



Components are abstracted:

- Authenticator
- Spawner
- Proxy

Pieces we've created:

- GSIAuthenticator (IT IS NO MORE)
- SSHAPIAuthenticator

- SSHSpawner (*Had* gsissh support)
- NERSCSpawner
- NERSCSlurmSpawner

Pieces we re-use and love:

- WrapSpawner (NERSCSpawner)
- BatchSpawner (NERSCSlurmSpawner)

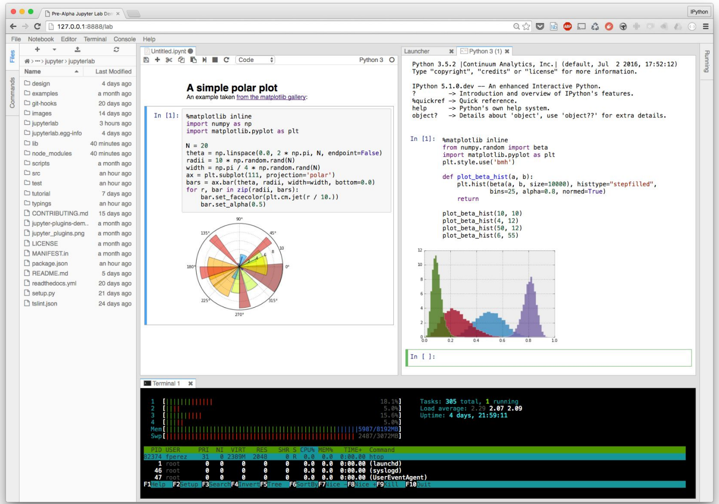
JupyterLab: Notebooks++

Modern Frontend for Jupyter

- Integrated GUI for Jupyter ecosystem (filebrowser, tabbed panes, notebooks, terminal etc.)
- Common framework to integrate multiple applications - **“extensions”**
- eg. connect multiple viewers to common underlying kernel

Extensions for many users?
Right now we manage them all.

… we’ve made a Slurm one I’ll show later.



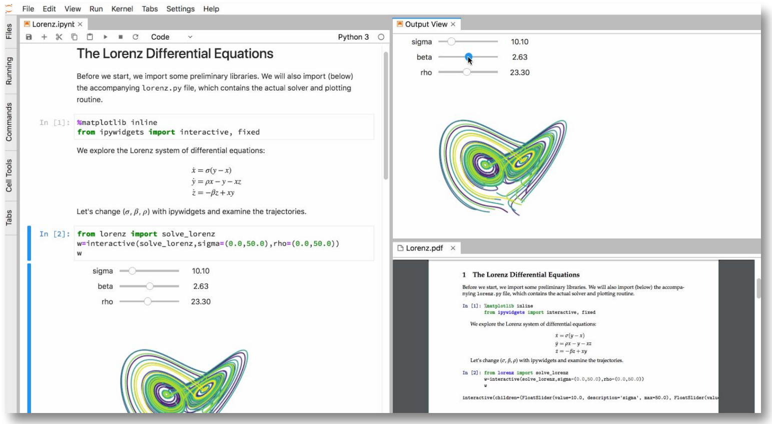
Name	Last Modified
design	4 days ago
examples	a month ago
git-hooks	20 days ago
images	14 days ago
jupyterlab	3 hours ago
jupyterlab.egg-info	4 days ago
lib	40 minutes ago
node_modules	40 minutes ago
scripts	a month ago
src	an hour ago
test	an hour ago
tutorial	7 days ago
typings	an hour ago
CONTRIBUTING.md	15 days ago
jupyter-plugins-dem..	a month ago
jupyter_plugins.png	a month ago
LICENSE	a month ago
MANIFEST.in	a month ago
package.json	an hour ago
README.md	5 days ago
readthedocs.yml	20 days ago
setup.py	21 days ago
tslint.json	24 days ago

A simple polar plot
An example taken from the matplotlib gallery:

```
In [1]: %matplotlib inline
        import numpy as np
        import matplotlib.pyplot as plt

        N = 20
        theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)
        radii = 10 * np.random.rand(N)
        width = np.pi / 4 * np.random.rand(N)
        ax = plt.subplot(111, projection='polar')
        bars = ax.bar(theta, radii, width=width, bottom=0.0)
        for r, bar in zip(radii, bars):
            bar.set_facecolor(plt.cm.jet(r / 10.))
            bar.set_alpha(0.5)
```

```
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  2 2016, 17:52:12)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0.dev -- 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.


In [1]: %matplotlib inline
        from numpy.random import beta
        import matplotlib.pyplot as plt
        plt.style.use('bmh')

        def plot_beta_hist(a, b):
            plt.hist(beta(a, b, size=10000), histtype="stepfilled",
                     bins=25, alpha=0.8, normed=True)
            return

        plot_beta_hist(10, 10)
        plot_beta_hist(4, 12)
        plot_beta_hist(50, 12)
        plot_beta_hist(6, 55)

In [ ]:
```

```
  1  [|||||||||||||                                        18.1%]     Tasks: 305 total, 1 running
  2  [||||                                                  5.0%]     Load average: 2.29 2.07 2.09
  3  [|||||||||||                                          15.6%]     Uptime: 4 days, 21:59:11
  4  [|||||                                                 5.0%]
  Mem[|||||||||||||||||||||||||||||||||||||||||||||||5987/8192MB]
  Swp[||||||||||||||||||||||||||||||||||||||||||||||2487/3072MB]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
82374 fperez     31   0 2389M  2048     0 R  0.0  0.0  0:00.00 htop
    1 root        0   0     0     0     0    0.0  0.0  0:00.00 (launchd)
   46 root        0   0     0     0     0    0.0  0.0  0:00.00 (syslogd)
   47 root        0   0     0     0     0    0.0  0.0  0:00.00 (UserEventAgent)
F1Help  F2Setup F3SearchF4InvertF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```



The Lorenz Differential Equations

Before we start, we import some preliminary libraries. We will also import (below) the accompanying `lorenz.py` file, which contains the actual solver and plotting routine.

```
In [1]: %matplotlib inline
        from ipywidgets import interactive, fixed
```

We explore the Lorenz system of differential equations:

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

Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

```
In [2]: from lorenz import solve_lorenz
        w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
        w
```

sigma	10.10
beta	2.63
rho	23.30

1 The Lorenz Differential Equations

Before we start, we import some preliminary libraries. We will also import (below) the accompanying `lorenz.py` file, which contains the actual solver and plotting routine.

```
In [1]: %matplotlib inline
        from ipywidgets import interactive, fixed
```

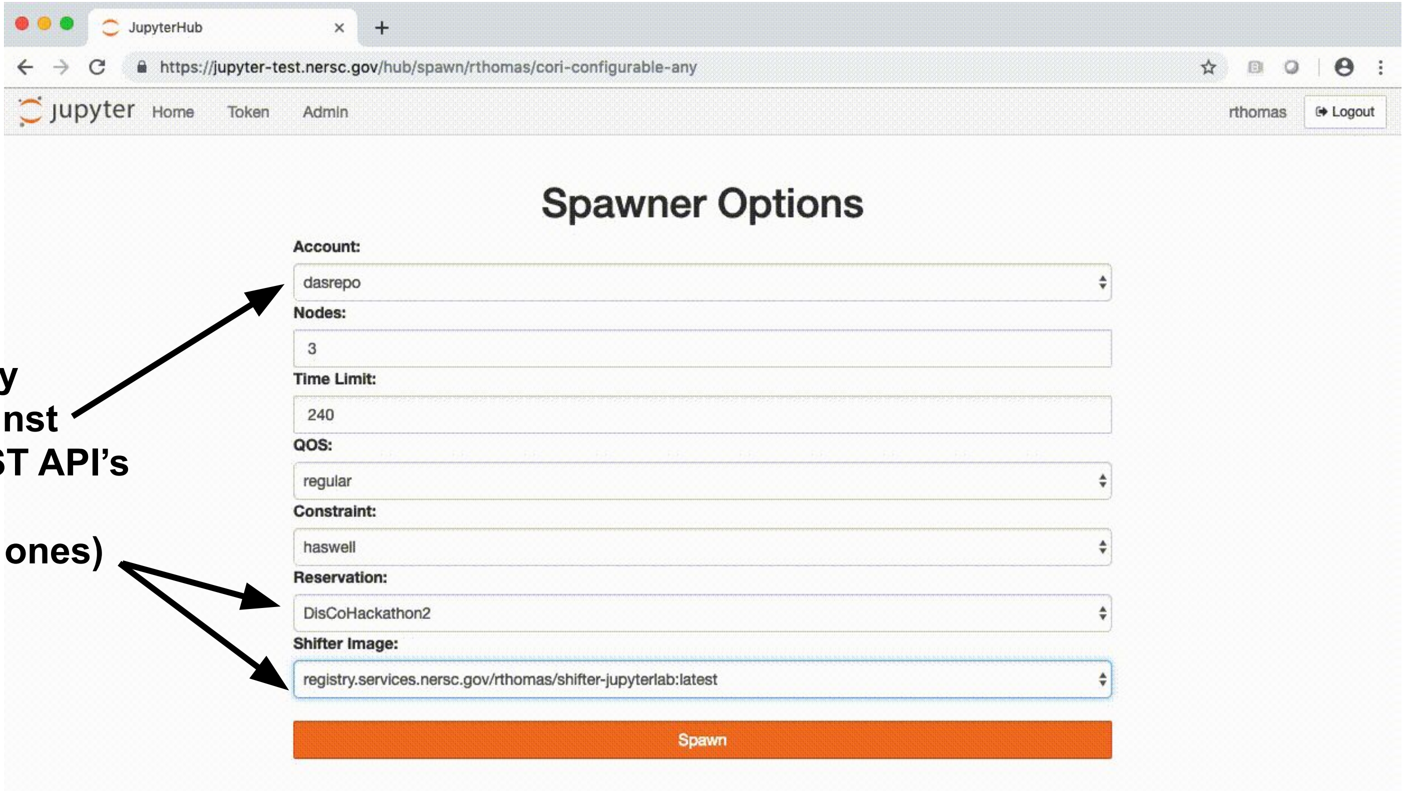
We explore the Lorenz system of differential equations:

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

Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.

```
In [2]: from lorenz import solve_lorenz
        w=interactive(solve_lorenz,sigma=(0.0,50.0),rho=(0.0,50.0))
        w

interactive(children=(FloatSlider(value=10.0, description='sigma', max=50.0), FloatSlider(valu
```

Handling Spawner Options


JupyterHub
https://jupyter-test.nersc.gov/hub/spawn/rthomas/cori-configurable-any
jupyter
Home
Token
Admin
rthomas
Logout

Spawner Options

Account: dasrepo

Nodes: 3

Time Limit: 240

QOS: regular

Constraint: haswell

Reservation: DisCoHackathon2

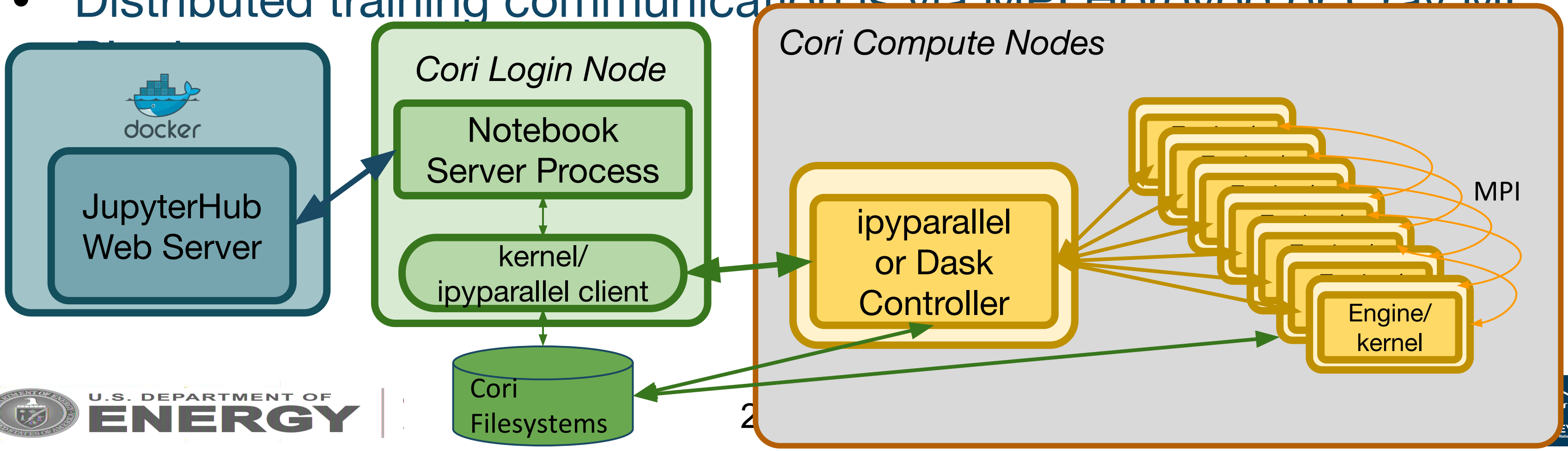
Shifter Image: registry.services.nersc.gov/rthomas/shifter-jupyterlab:latest

Spawn

Populated by queries against internal REST API's at NERSC (or example ones)

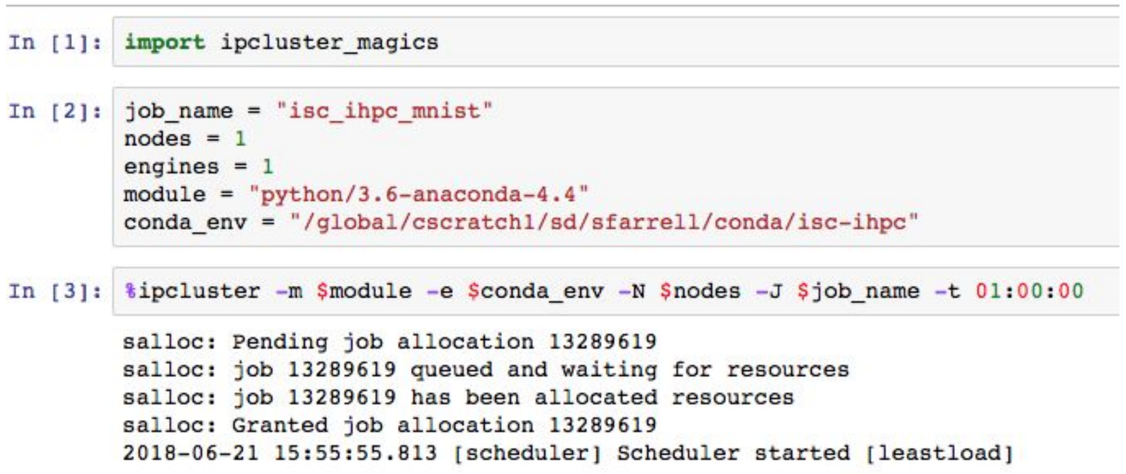
Jupyter architecture

- Allocate nodes on Cori interactive queue and start ipyparallel or Dask cluster
  - Developed %ipcluster magic to setup within notebook
- Compute nodes traditionally do not have external address
  - Required network configuration / policy decisions
- Distributed training communication is via MPI Horovod or Cray ML



Setting up ipyparallel cluster

Via Magic (entire workflow in notebook) or a console script

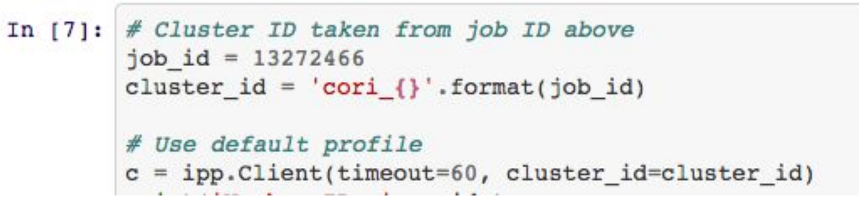


```
In [1]: import ipcluster_magics

In [2]: job_name = "isc_ihpc_mnist"
        nodes = 1
        engines = 1
        module = "python/3.6-anaconda-4.4"
        conda_env = "/global/cscratch1/sd/sfarrell/conda/isc-ihpc"

In [3]: %ipcluster -m $module -e $conda_env -N $nodes -J $job_name -t 01:00:00

        salloc: Pending job allocation 13289619
        salloc: job 13289619 queued and waiting for resources
        salloc: job 13289619 has been allocated resources
        salloc: Granted job allocation 13289619
        2018-06-21 15:55:55.813 [scheduler] Scheduler started [leastload]
```

```
salloc --qos=interactive -N 1 -C haswell
wbhimji@nid00032:~> ./startCluster.sh
```

```
# Use a unique cluster ID for this job
clusterID=cori_${SLURM_JOB_ID}
echo "Launching controller"
ipcontroller --ip="$headIP" \
      --cluster-id=$clusterID &
sleep 20
echo "Launching engines"
srun ipengine --cluster-id=$clusterID
```

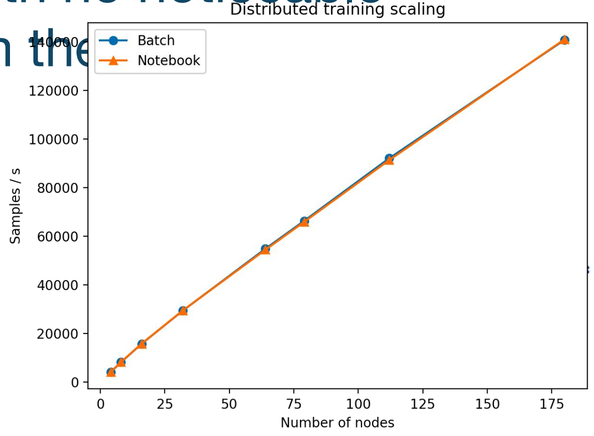
Connect to cluster in notebook

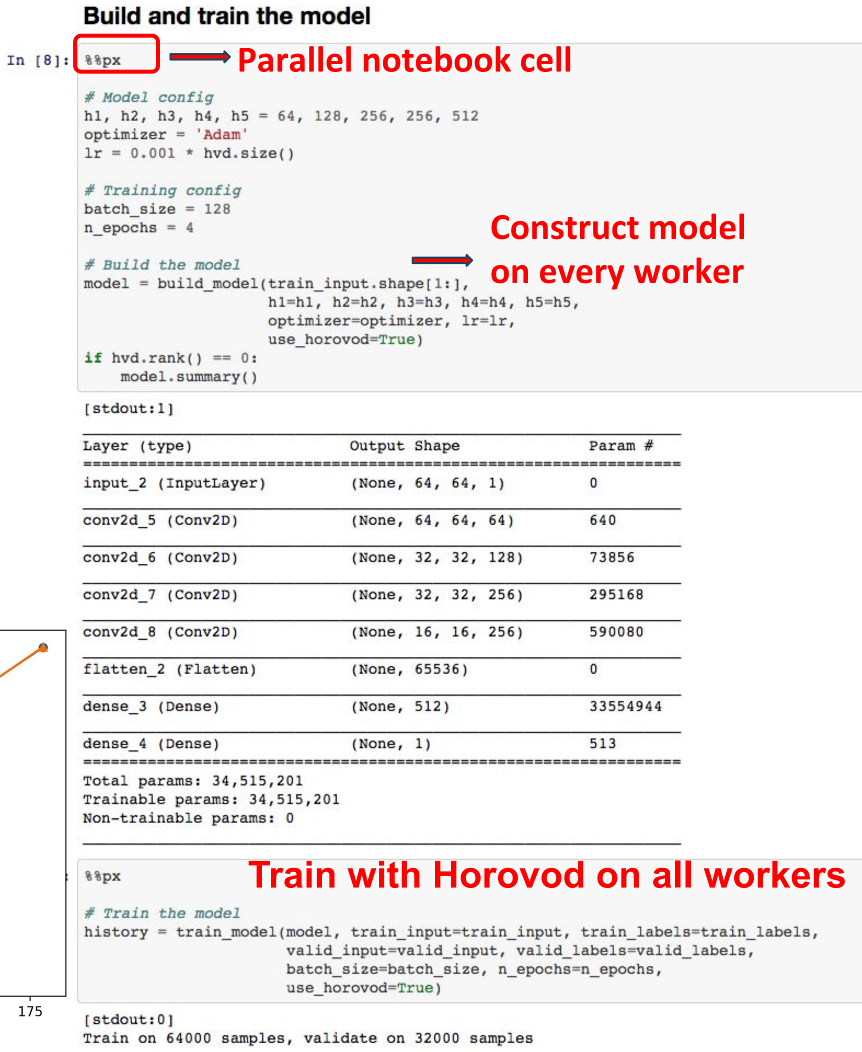


```
In [7]: # Cluster ID taken from job ID above
        job_id = 13272466
        cluster_id = 'cori_{}'.format(job_id)

        # Use default profile
        c = ipp.Client(timeout=60, cluster_id=cluster_id)
```

Distributed Training

- Distributed training in notebooks with IPyParallel and Horovod-MPI
- Notebook cells specified for parallel execution using cell magic
  - MPI code in a notebook
- Scales well with no noticeable overhead from the infrastructure





Build and train the model

```
In [8]: %%px

# Model config
h1, h2, h3, h4, h5 = 64, 128, 256, 256, 512
optimizer = 'Adam'
lr = 0.001 * hvd.size()

# Training config
batch_size = 128
n_epochs = 4

# Build the model
model = build_model(train_input.shape[1:],
                    h1=h1, h2=h2, h3=h3, h4=h4, h5=h5,
                    optimizer=optimizer, lr=lr,
                    use_horovod=True)

if hvd.rank() == 0:
    model.summary()
```

Parallel notebook cell

Construct model on every worker

```
[stdout:1]
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 64, 64, 1)         0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 64, 64, 64)        640
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 32, 32, 128)       73856
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 32, 32, 256)       295168
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 16, 16, 256)       590080
_________________________________________________________________
flatten_2 (Flatten)          (None, 65536)             0
_________________________________________________________________
dense_3 (Dense)              (None, 512)               33554944
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 513
=================================================================
Total params: 34,515,201
Trainable params: 34,515,201
Non-trainable params: 0
_________________________________________________________________
```

```
%%px

# Train the model
history = train_model(model, train_input=train_input, train_labels=train_labels,
                      valid_input=valid_input, valid_labels=valid_labels,
                      batch_size=batch_size, n_epochs=n_epochs,
                      use_horovod=True)
```

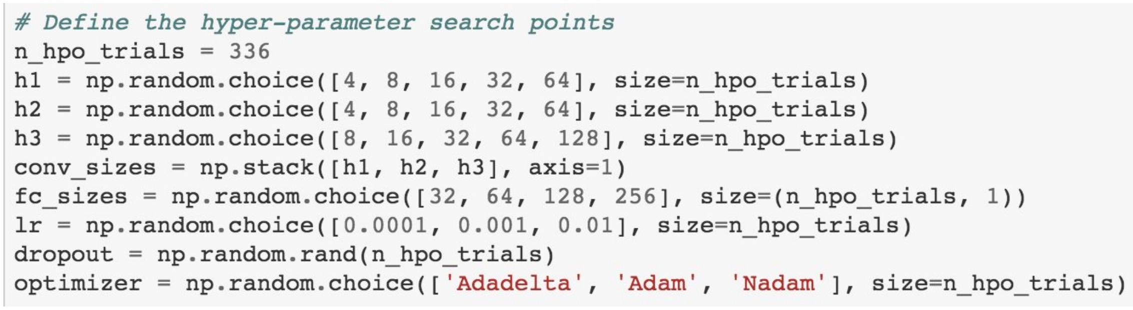
Train with Horovod on all workers

```
[stdout:0]
Train on 64000 samples, validate on 32000 samples
```

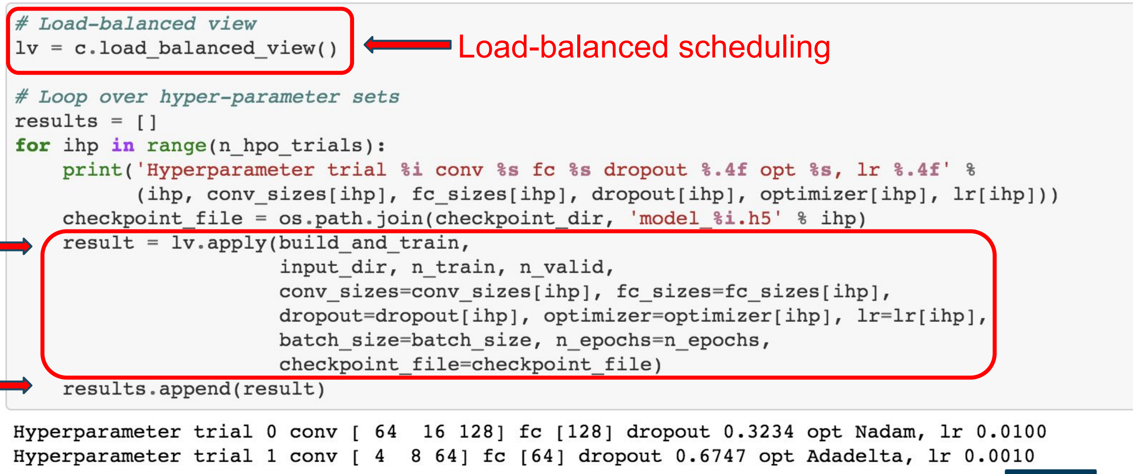
Distributed HPO - Setup

Easy but powerful setup for random search HPO

- Define HP sets to evaluate
- Define model training function
- Run the HPO tasks with load-balanced scheduler



```
# Define the hyper-parameter search points
n_hpo_trials = 336
h1 = np.random.choice([4, 8, 16, 32, 64], size=n_hpo_trials)
h2 = np.random.choice([4, 8, 16, 32, 64], size=n_hpo_trials)
h3 = np.random.choice([8, 16, 32, 64, 128], size=n_hpo_trials)
conv_sizes = np.stack([h1, h2, h3], axis=1)
fc_sizes = np.random.choice([32, 64, 128, 256], size=(n_hpo_trials, 1))
lr = np.random.choice([0.0001, 0.001, 0.01], size=n_hpo_trials)
dropout = np.random.rand(n_hpo_trials)
optimizer = np.random.choice(['Adadelta', 'Adam', 'Nadam'], size=n_hpo_trials)
```



```
# Load-balanced view
lv = c.load_balanced_view()
```

Load-balanced scheduling

```
# Loop over hyper-parameter sets
results = []
for ihp in range(n_hpo_trials):
    print('Hyperparameter trial %i conv %s fc %s dropout %.4f opt %s, lr %.4f' %
          (ihp, conv_sizes[ihp], fc_sizes[ihp], dropout[ihp], optimizer[ihp], lr[ihp]))
    checkpoint_file = os.path.join(checkpoint_dir, 'model_%i.h5' % ihp)
    result = lv.apply(build_and_train,
                      input_dir, n_train, n_valid,
                      conv_sizes=conv_sizes[ihp], fc_sizes=fc_sizes[ihp],
                      dropout=dropout[ihp], optimizer=optimizer[ihp], lr=lr[ihp],
                      batch_size=batch_size, n_epochs=n_epochs,
                      checkpoint_file=checkpoint_file)
    results.append(result)
```

Launch user-defined training function and arguments

AsyncResult objects can be queried for status, outputs

```
Hyperparameter trial 0 conv [ 64  16 128] fc [128] dropout 0.3234 opt Nadam, lr 0.0100
Hyperparameter trial 1 conv [ 4  8 64] fc [64] dropout 0.6747 opt Adadelta, lr 0.0010
```

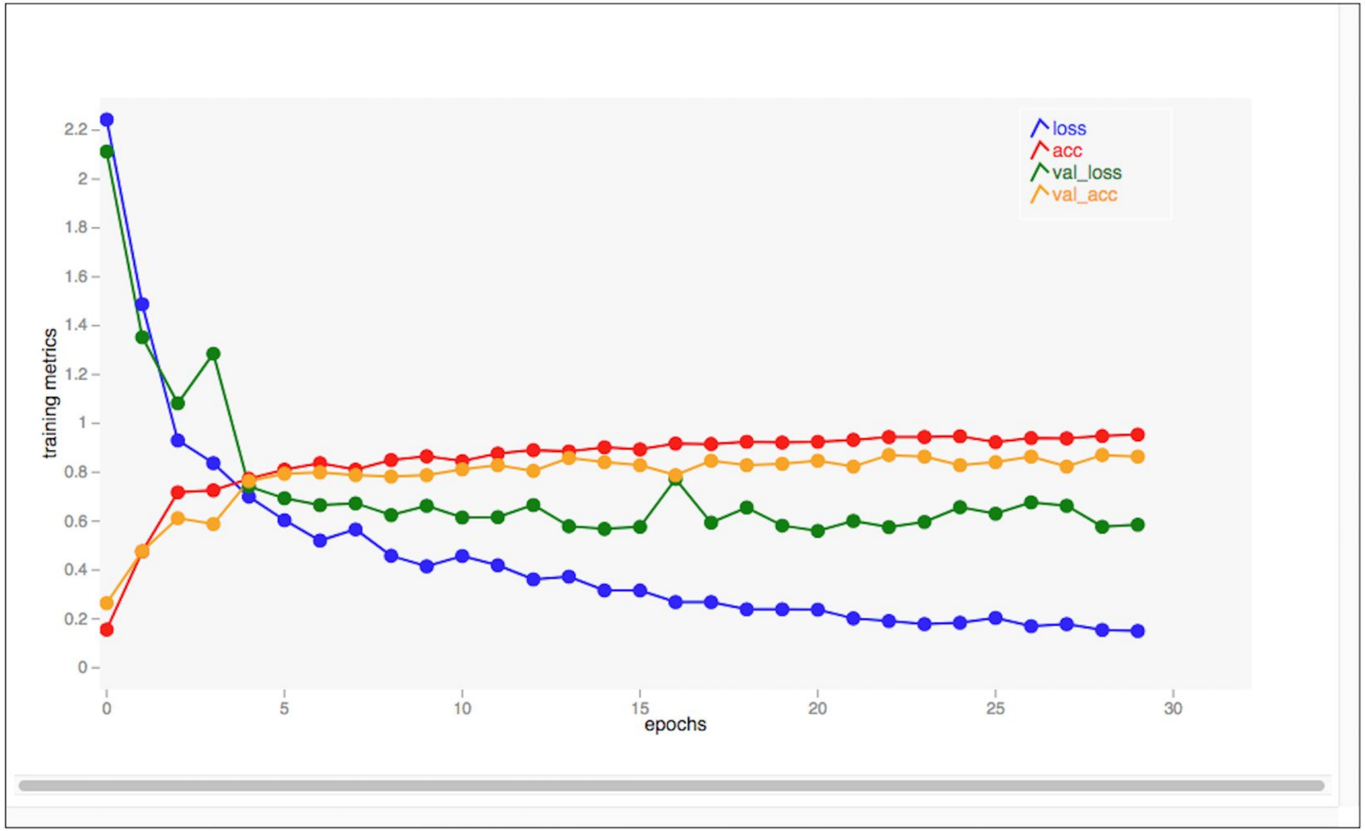
Plots update live

Table shows different configurations:

- Status
- Current loss and accuracy
- Sort

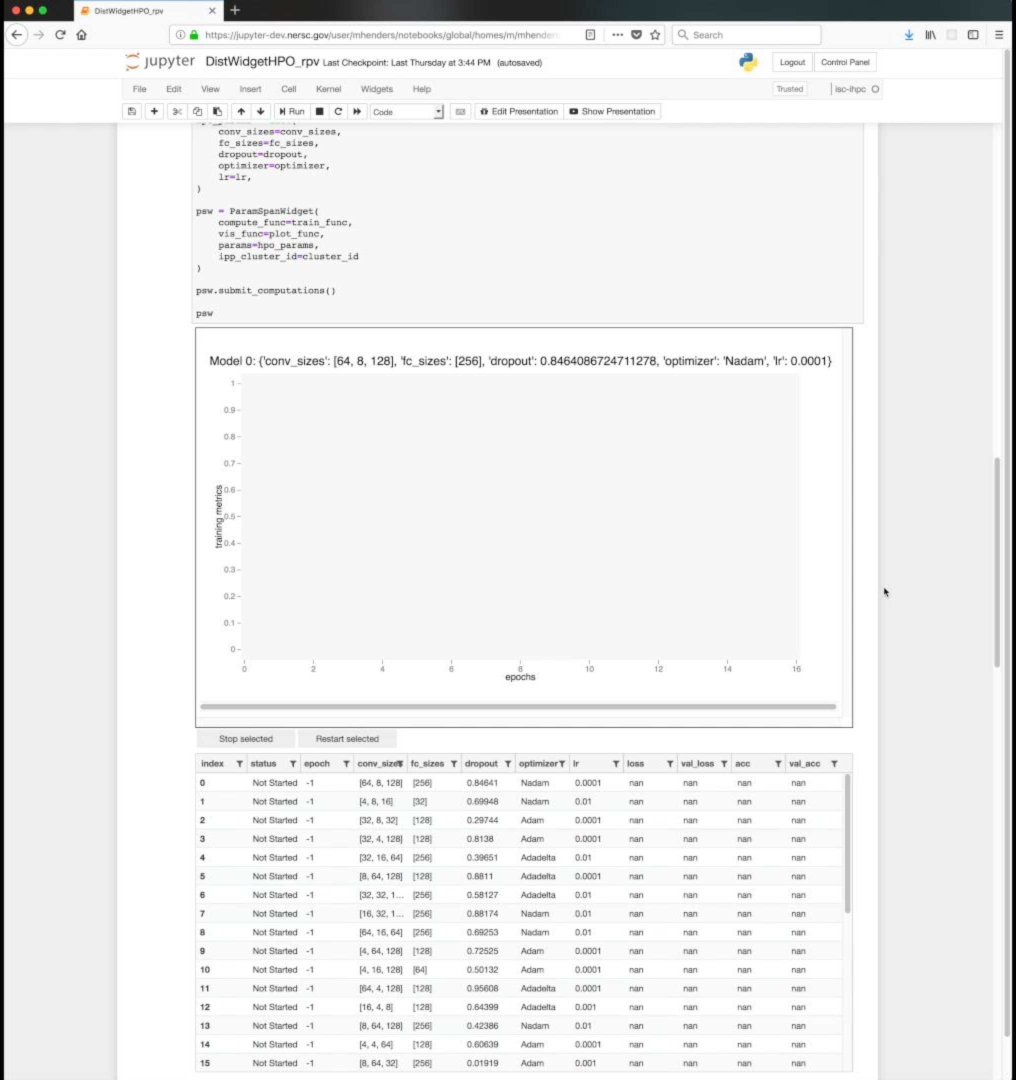
Can add further quantities to plot and interaction buttons

https://github.com/sparticlesteve/cori-intml-examples/



Stop selected Restart selected

index	status	epoch	h1	h2	h3	dropout	optimizer	loss	val_loss	acc	val_acc
3	Ended Tra...	31	16	64	16	0.88467	Adam	2.003565...	1.689878...	0.213253...	0.682352...
2	Ended Tra...	31	16	8	8	0.19765	Adam	0.852827...	0.829521...	0.763855...	0.800000...
0	Ended Tra...	31	64	8	8	0.04836	Adadelta	0.157987...	0.579903...	0.944578...	0.870588...
1	Ended Tra...	31	4	8	16	0.03825	Adadelta	0.151153...	0.585079...	0.954216...	0.864705...

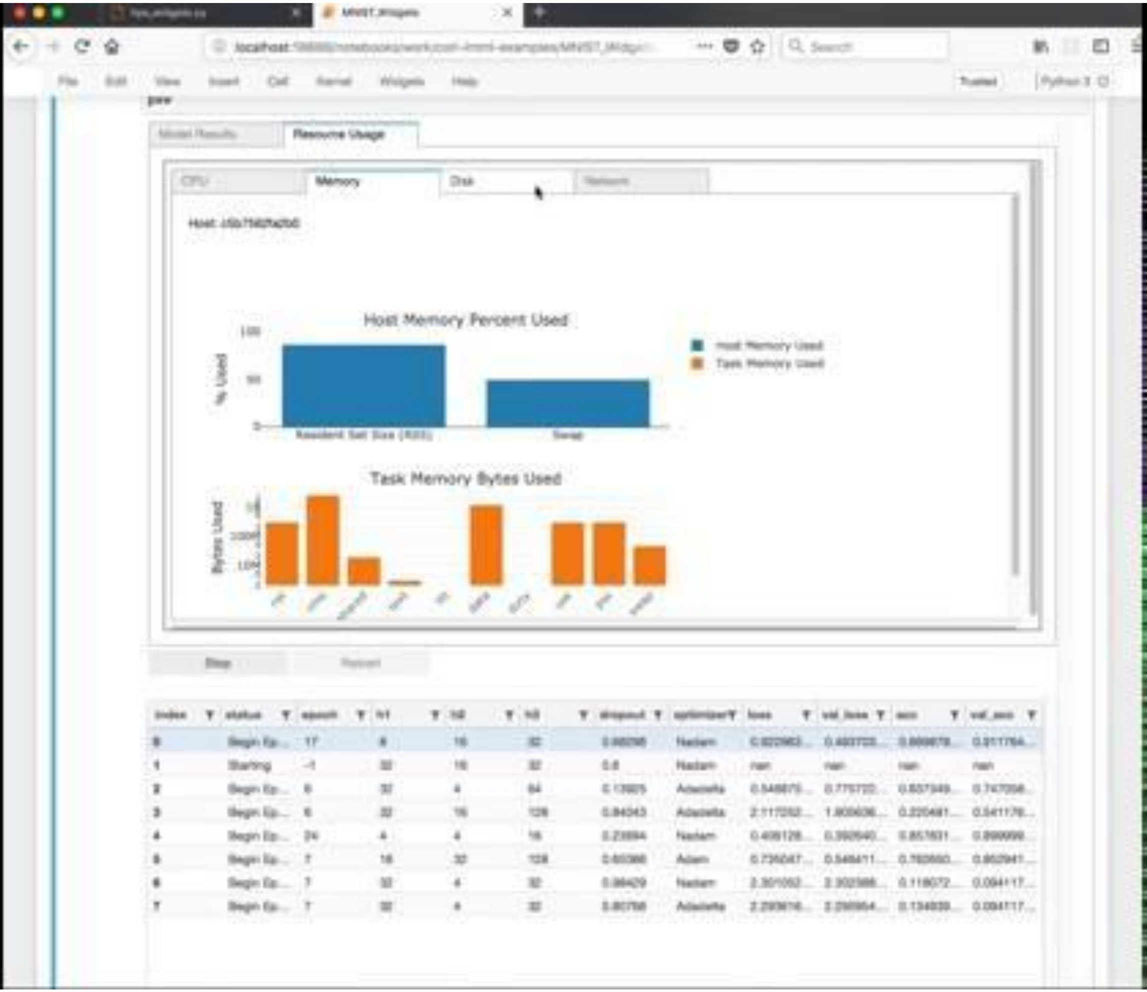

DistWidgetHPO_rpv
https://jupyter-dev.nersc.gov/user/mhenders/notebooks/global/homes/m/mhenders
jupyter DistWidgetHPO_rpv Last Checkpoint: Last Thursday at 3:44 PM (autosaved)
Logout
Control Panel
File Edit View Insert Cell Kernel Widgets Help
Trusted
isc-ihpc
Run
Code
Edit Presentation
Show Presentation

```
    conv_sizes=conv_sizes,
    fc_sizes=fc_sizes,
    dropout=dropout,
    optimizer=optimizer,
    lr=lr,
)

psw = ParamSpanWidget(
    compute_func=train_func,
    vis_func=plot_func,
    params=hpo_params,
    ipp_cluster_id=cluster_id
)

psw.submit_computations()

psw
```

Model 0: {'conv_sizes': [64, 8, 128], 'fc_sizes': [256], 'dropout': 0.8464086724711278, 'optimizer': 'Nadam', 'lr': 0.0001}

training metrics
epochs
Stop selected
Restart selected

index	status	epoch	conv_sizes	fc_sizes	dropout	optimizer	lr	loss	val_loss	acc	val_acc
0	Not Started	-1	[64, 8, 128]	[256]	0.84641	Nadam	0.0001	nan	nan	nan	nan
1	Not Started	-1	[4, 8, 16]	[32]	0.69948	Nadam	0.01	nan	nan	nan	nan
2	Not Started	-1	[32, 8, 32]	[128]	0.29744	Adam	0.0001	nan	nan	nan	nan
3	Not Started	-1	[32, 4, 128]	[128]	0.8138	Adam	0.0001	nan	nan	nan	nan
4	Not Started	-1	[32, 16, 64]	[256]	0.39651	Adadelta	0.01	nan	nan	nan	nan
5	Not Started	-1	[8, 64, 128]	[128]	0.8811	Adadelta	0.0001	nan	nan	nan	nan
6	Not Started	-1	[32, 32, 1…	[256]	0.58127	Adadelta	0.01	nan	nan	nan	nan
7	Not Started	-1	[16, 32, 1…	[256]	0.88174	Nadam	0.01	nan	nan	nan	nan
8	Not Started	-1	[64, 16, 64]	[256]	0.69253	Nadam	0.01	nan	nan	nan	nan
9	Not Started	-1	[4, 64, 128]	[128]	0.72525	Adam	0.0001	nan	nan	nan	nan
10	Not Started	-1	[4, 16, 128]	[64]	0.50132	Adam	0.0001	nan	nan	nan	nan
11	Not Started	-1	[64, 4, 128]	[128]	0.95608	Adadelta	0.0001	nan	nan	nan	nan
12	Not Started	-1	[16, 4, 8]	[128]	0.64399	Adadelta	0.001	nan	nan	nan	nan
13	Not Started	-1	[8, 64, 128]	[256]	0.42386	Nadam	0.01	nan	nan	nan	nan
14	Not Started	-1	[4, 4, 64]	[128]	0.60639	Adam	0.0001	nan	nan	nan	nan
15	Not Started	-1	[8, 64, 32]	[256]	0.01919	Adam	0.001	nan	nan	nan	nan

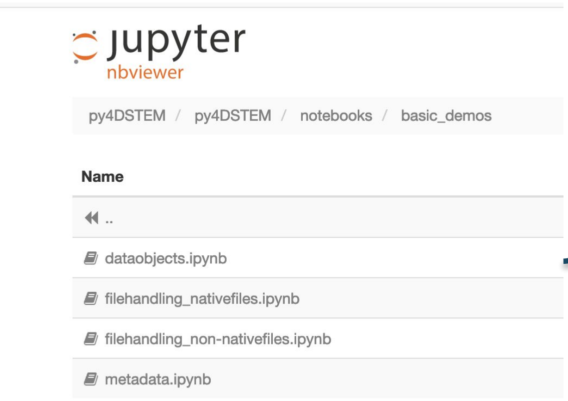


Host: [illegible]

index	status	epoch	h1	h2	h3	dropout	optimizer	loss	val_loss	acc	val_acc
0	Begin Ep...	17	8	16	32	0.66296	Nadam	0.322982...	0.493723...	0.866878...	0.911764...
1	Starting	-1	32	16	32	0.6	Nadam	nan	nan	nan	nan
2	Begin Ep...	6	32	4	64	0.13925	Adadelta	0.548875...	0.775722...	0.837349...	0.747058...
3	Begin Ep...	6	32	16	128	0.84043	Adadelta	2.117252...	1.809636...	0.220481...	0.411176...
4	Begin Ep...	24	4	4	16	0.23894	Nadam	0.408128...	0.392640...	0.857831...	0.899999...
5	Begin Ep...	7	16	32	128	0.60388	Adam	0.735047...	0.548411...	0.782650...	0.852941...
6	Begin Ep...	7	32	4	32	0.96429	Nadam	2.301052...	2.302388...	0.118072...	0.094117...
7	Begin Ep...	7	32	4	32	0.80768	Adadelta	2.293616...	2.290954...	0.134939...	0.094117...

Curated Notebook Environments

- Browse curated examples managed by project using tools like nbviewer
- Clone and launch notebook into users workspace with appropriate conda environment
- Reproducible Notebooks – Similar to Binder

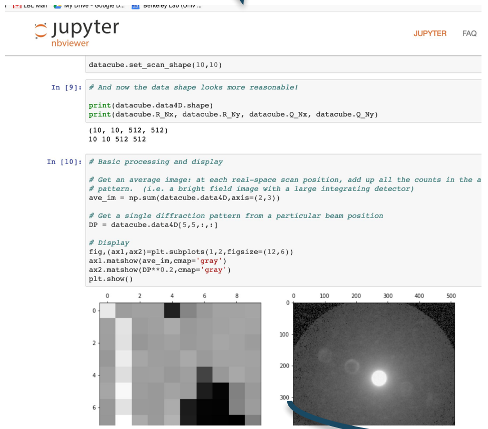


py4DSTEM / py4DSTEM / notebooks / basic_demos

Name
..
dataobjects.ipynb
filehandling_nativefiles.ipynb
filehandling_non-nativefiles.ipynb
metadata.ipynb

Browse

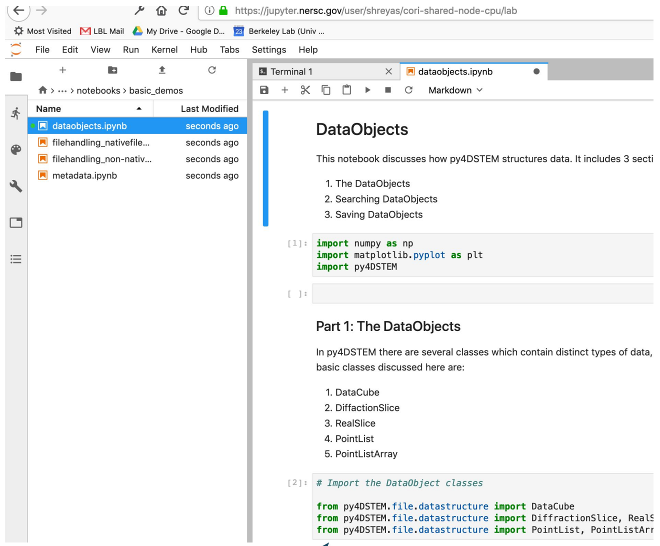
View



```
datacube.set_scan_shape(10,10)
```

```
In [9]: # And now the data shape looks more reasonable!

        print(datacube.data4D.shape)
        print(datacube.R_Nx, datacube.R_Ny, datacube.Q_Nx, datacube.Q_Ny)

        (10, 10, 512, 512)
        10 10 512 512
```

```
In [10]: # Basic processing and display

         # Get an average image: at each real-space scan position, add up all the counts in the a
         # pattern.  (i.e. a bright field image with a large integrating detector)
         ave_im = np.sum(datacube.data4D,axis=(2,3))

         # Get a single diffraction pattern from a particular beam position
         DP = datacube.data4D[5,5,:,:]

         # Display
         fig,(ax1,ax2)=plt.subplots(1,2,figsize=(12,6))
         ax1.matshow(ave_im,cmap='gray')
         ax2.matshow(DP**0.2,cmap='gray')
         plt.show()
```



... > notebooks > basic_demos

Name	Last Modified
dataobjects.ipynb	seconds ago
filehandling_nativefile...	seconds ago
filehandling_non-nativ...	seconds ago
metadata.ipynb	seconds ago

DataObjects

This notebook discusses how py4DSTEM structures data. It includes 3 secti

1. The DataObjects
2. Searching DataObjects
3. Saving DataObjects

```
[1]: import numpy as np
     import matplotlib.pyplot as plt
     import py4DSTEM

[ ]:
```

Part 1: The DataObjects

In py4DSTEM there are several classes which contain distinct types of data, basic classes discussed here are:

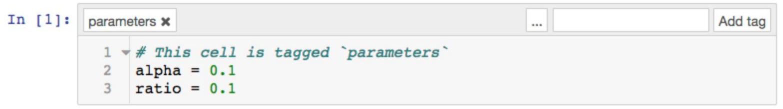
1. DataCube
2. DiffactionSlice
3. RealSlice
4. PointList
5. PointListArray

```
[2]: # Import the DataObject classes

     from py4DSTEM.file.datastructure import DataCube
     from py4DSTEM.file.datastructure import DiffractionSlice, RealS
     from py4DSTEM.file.datastructure import PointList, PointListArr
```

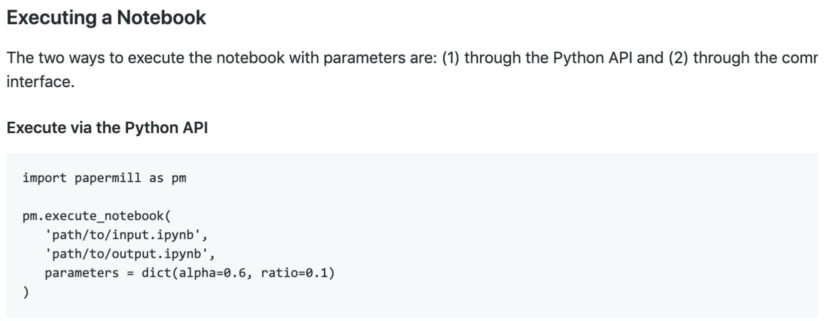
Launch

Parameterized Notebooks

- Run the same notebook against
  - different sets of parameters
  - Different datasets
  - Think “data parallel”
- Running as notebook gives you a live document of each task
- Save successful runs



```
# This cell is tagged `parameters`
alpha = 0.1
ratio = 0.1
```



Executing a Notebook

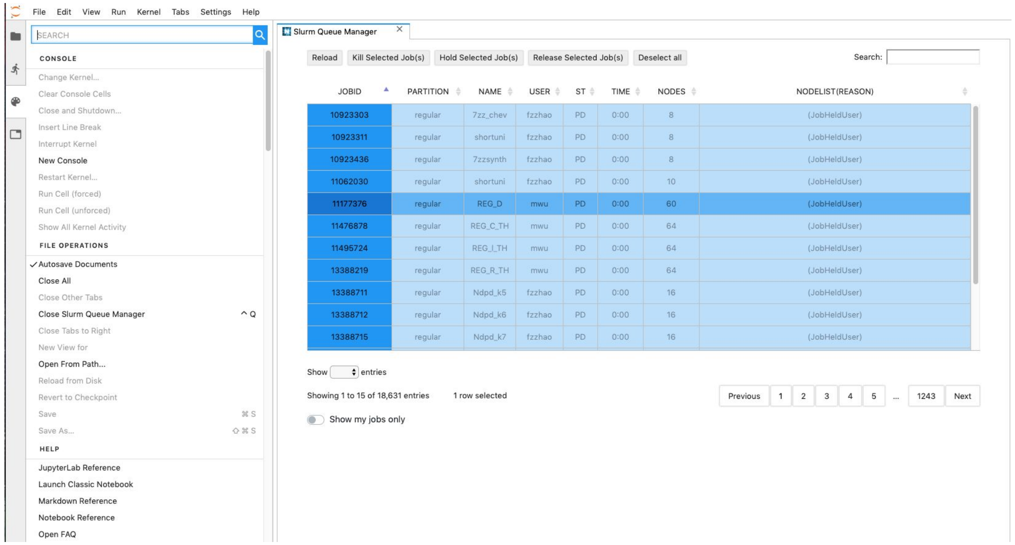
The two ways to execute the notebook with parameters are: (1) through the Python API and (2) through the comr
interface.

Execute via the Python API

```
import papermill as pm

pm.execute_notebook(
   'path/to/input.ipynb',
   'path/to/output.ipynb',
   parameters = dict(alpha=0.6, ratio=0.1)
)
```

Jupyterlab Extensions

- **SLURM Extension**
- **Resource Usage Monitoring Extension**



JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
10923303	regular	7zz_chev	fzzhao	PD	0:00	8	(JobHeldUser)
10923311	regular	shortuni	fzzhao	PD	0:00	8	(JobHeldUser)
10923436	regular	7zzsynth	fzzhao	PD	0:00	8	(JobHeldUser)
11062030	regular	shortuni	fzzhao	PD	0:00	10	(JobHeldUser)
11177376	regular	REG_D	mwu	PD	0:00	60	(JobHeldUser)
11476878	regular	REG_C_TH	mwu	PD	0:00	64	(JobHeldUser)
11495724	regular	REG_I_TH	mwu	PD	0:00	64	(JobHeldUser)
13388219	regular	REG_R_TH	mwu	PD	0:00	64	(JobHeldUser)
13388711	regular	Ndpd_k5	fzzhao	PD	0:00	16	(JobHeldUser)
13388712	regular	Ndpd_k6	fzzhao	PD	0:00	16	(JobHeldUser)
13388715	regular	Ndpd_k7	fzzhao	PD	0:00	16	(JobHeldUser)

Show entries

Showing 1 to 15 of 18,631 entries   1 row selected

Previous 1 2 3 4 5 ... 1243 Next

Show my jobs only