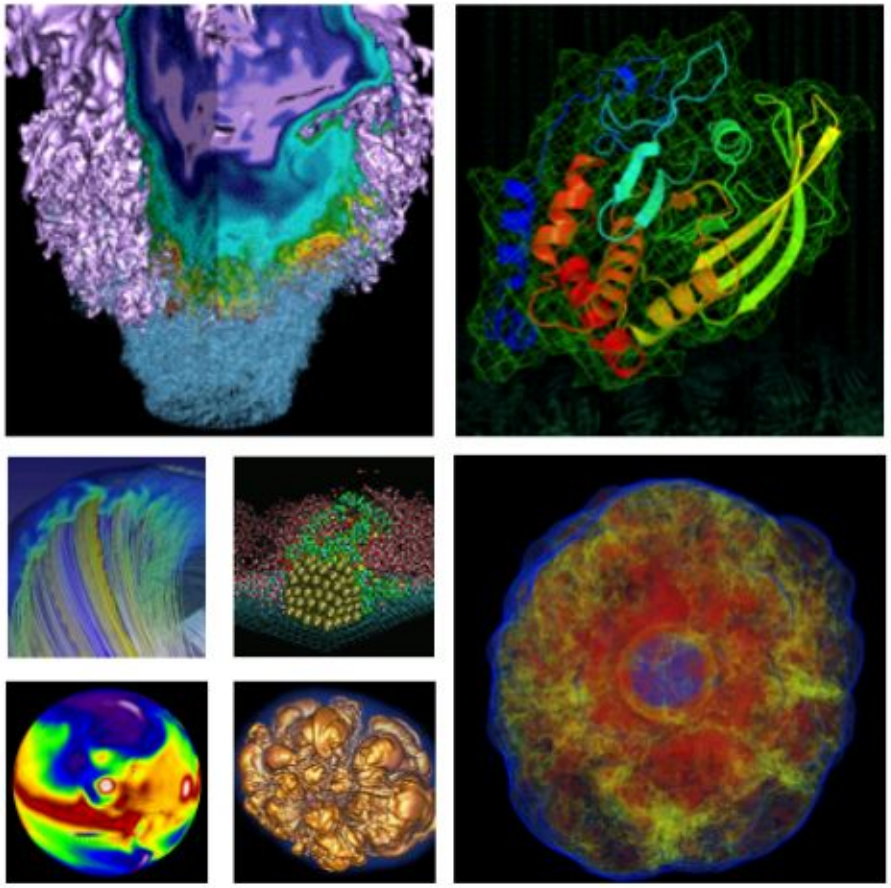
Jupyter at NERSC

Interactive Supercomputing: HOWTO



Rollin Thomas
Data and Analytics Services
NERSC, LBNL

Shreyas Cholia
Usable Software Systems
Computational Research Division/NERSC, LBNL

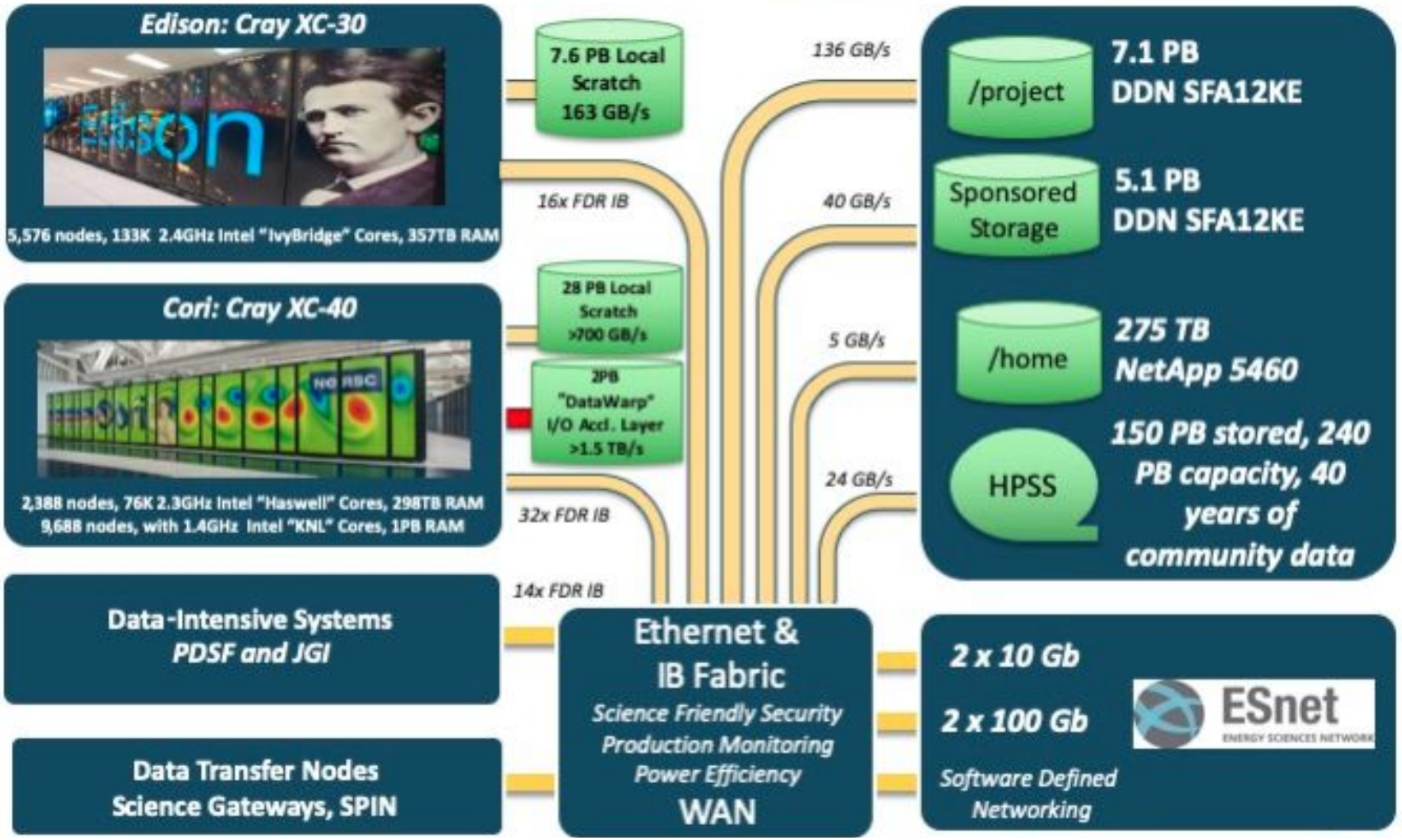
Blue Waters Webinar
April 24 2019

What is NERSC?

National Energy Research Scientific Computing Center

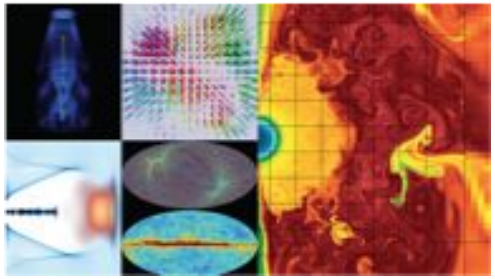
The production user facility for high performance computing and data for the Department of Energy's Office of Science.

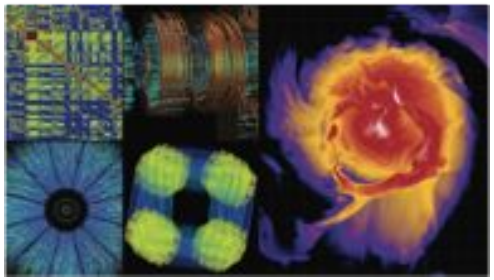
NERSC Resources

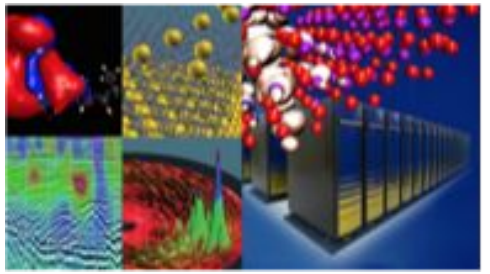


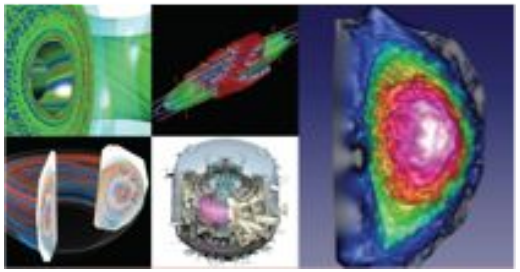
Who Uses NERSC?

7000 users
700 projects
2500+ publications/yr
10 billion "NERSC" hours
Diverse workload areas:

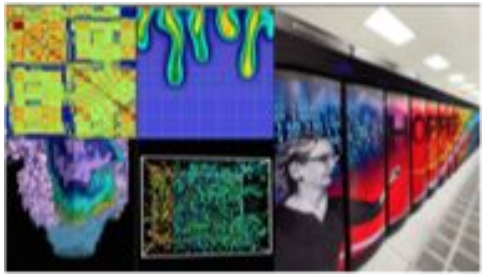

High Energy Physics


Nuclear Sciences


Materials, Chemistry, Geophysics


Fusion, Plasma Physics

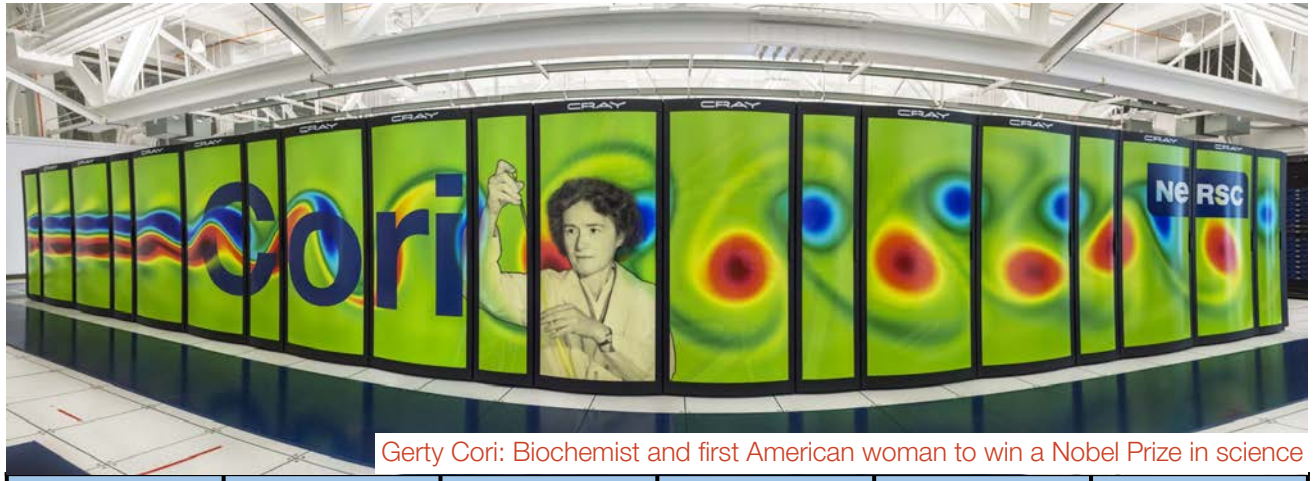

Bio Energy, Environment


Advanced Computing

Experimental and Observational Data

Increased user presence at NERSC from EOD facilities.
Challenge: Supporting this workload alongside simulations.
Real-time, dynamic workflows, HITL analysis/steering.

Cori: Friendly for "Data Users"



Gerty Cori: Biochemist and first American woman to win a Nobel Prize in science

Processor Type	Speed/Cores per Node	Peak Performance	# Nodes	Aggregate Memory	Memory per Node
Haswell	2.3/32	1.92 PF/s	2388	305 TB	128 GB
KNL	1.4/68	28 PF/s	9688	1.1 PB	96+16 GB

slurm workload manager

SHIFTER

CRAY DATAWARP

globus

swift

FireWorks

jupyter

NVRAM Burst Buffer for I/O acceleration
Shared, real-time, interactive queues
Shifter for containerized HPC
Special-purpose large memory "extra login" nodes (512 & 768 GB):

Workflows, data management, bigmem queue, … and Jupyter!

Motivation for a JupyterHub Service

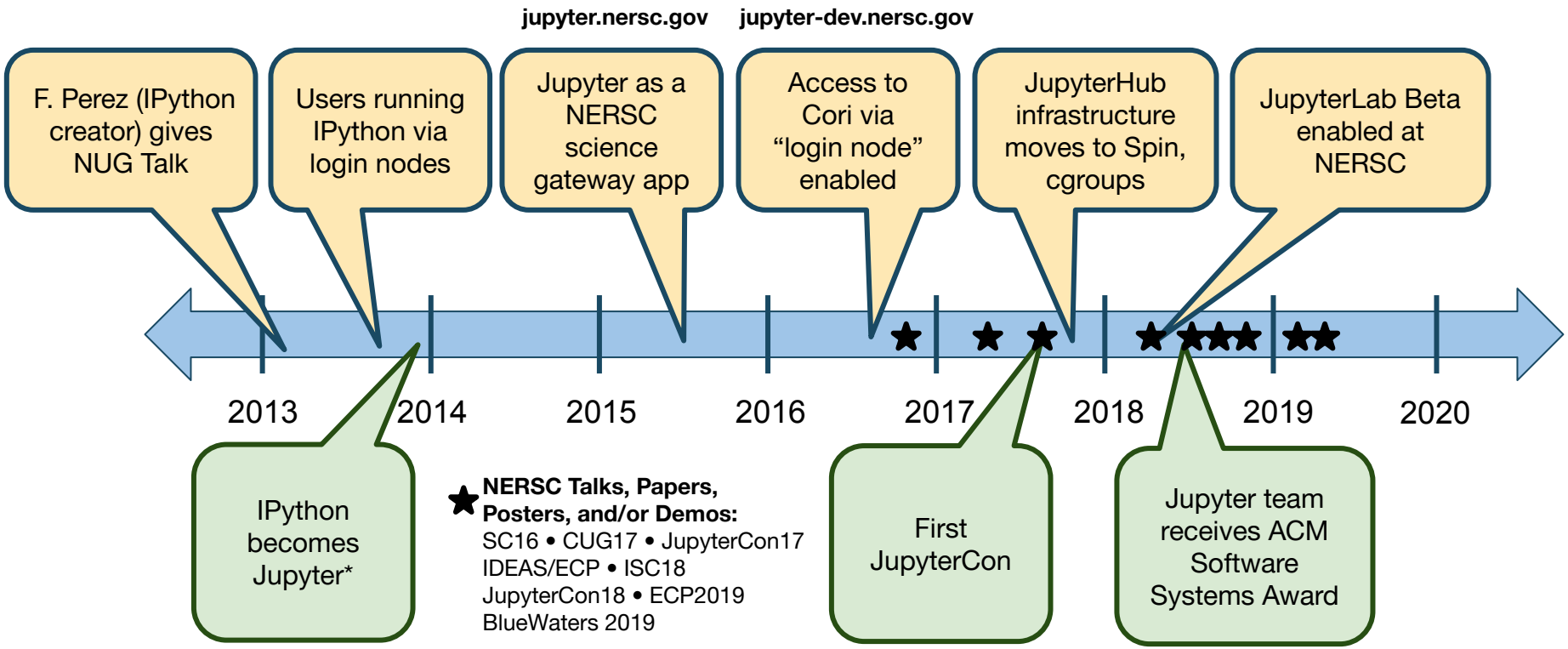
Users could run on login nodes and ssh tunnel:

- ✗ **Users running their own notebook servers on a supercomputer can be a security concern.**
- ✗ **Difficult to support and manage different kernels and environments.**
- ✗ **Setting up ssh tunnels is inconvenient and annoying.**

JupyterHub to rescue:

- ✓ **Centralized service to deploy notebooks in a standard authenticated manner.**
- ✓ **Package known kernels out of the box (Anaconda).**
- ✓ **Access to NERSC resources through single interface: Filesystems, Batch Queue, Network, DBs**
- ✓ **Clarify expectations to users about what we bless & support.**

History



This year:

More compute resources for Jupyter
Access to computes in production
Extensions for HPC and NERSC

Jupyter Matters to Our Users

Users appreciate Jupyter @ NERSC...

"I really like the jupyter interface."

"New jupyter notebooks are awesome!"

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

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

"... 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.

"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.)"

NERSC Annual User Survey Comments (2018, for CY17)
User Comments

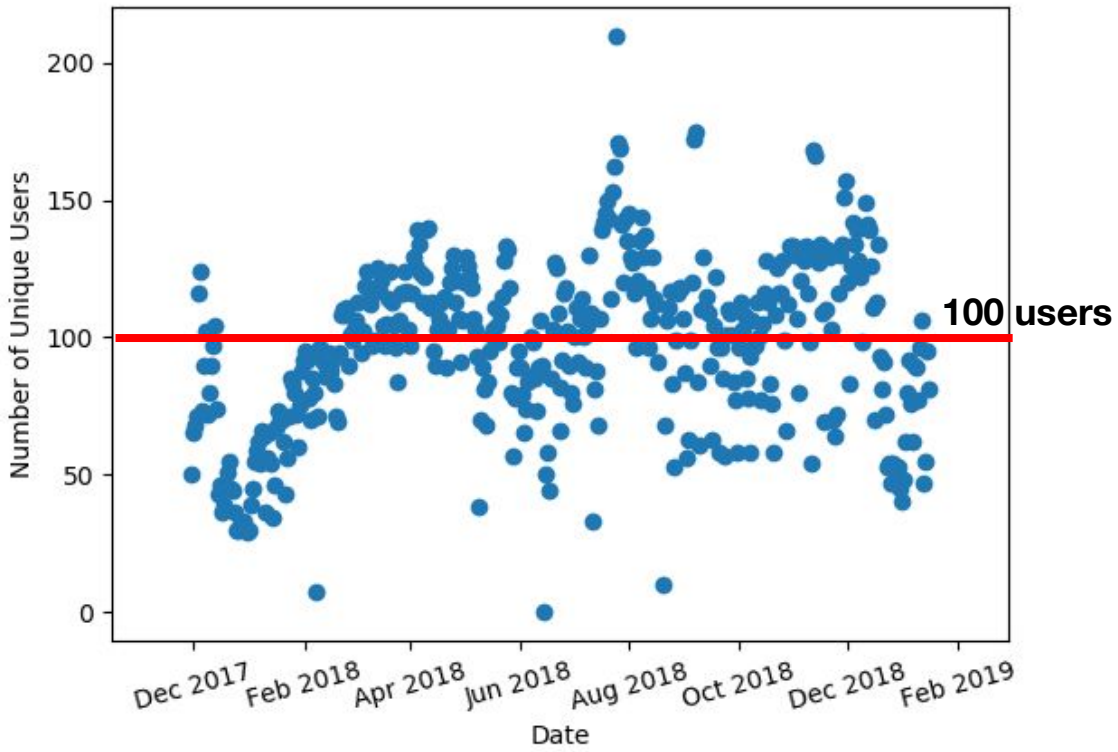
Single Dedicated Cori Node Usage

Hundreds of unique users of Jupyter on Cori per month.

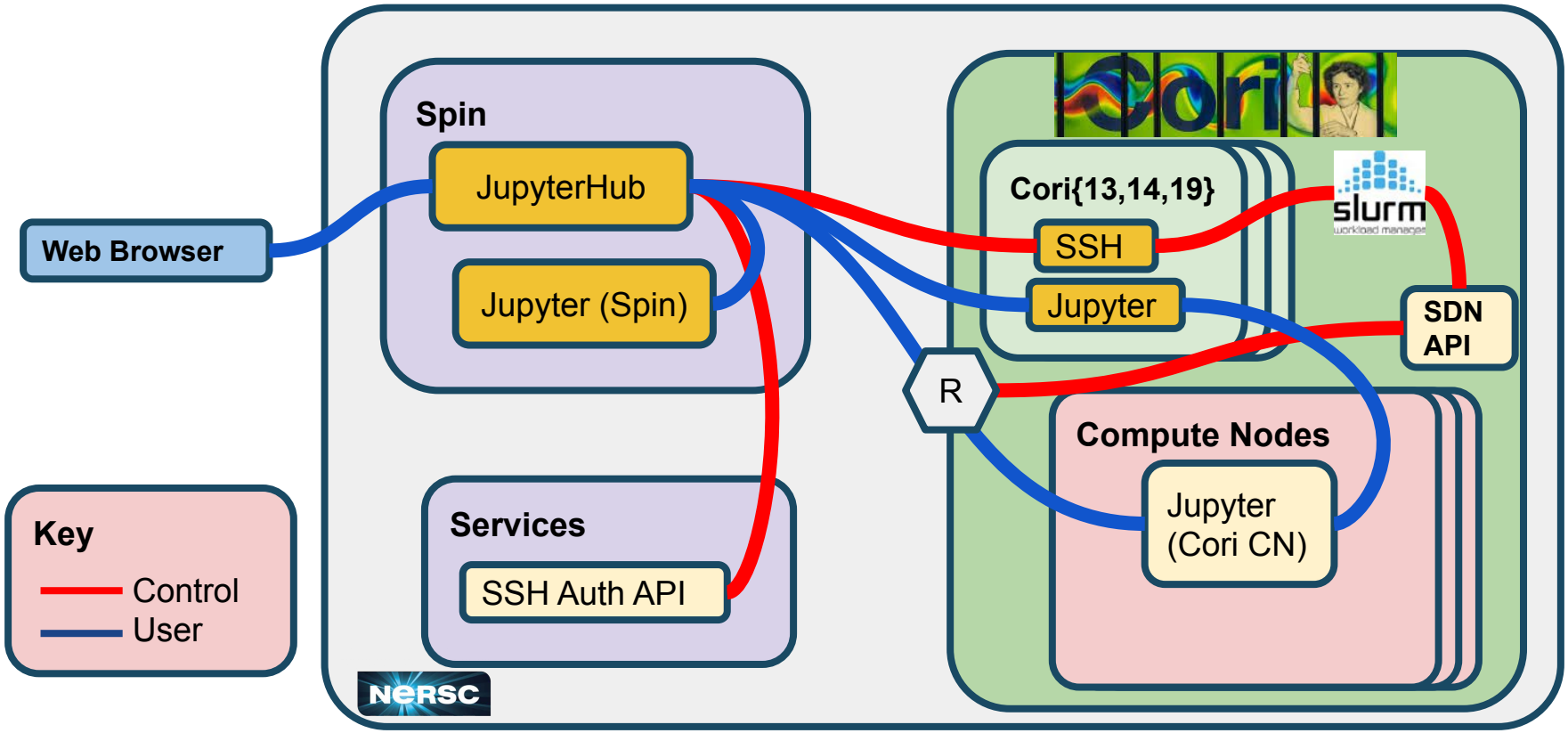
Every 5 minutes we sampled process table & memory use on the Cori Jupyter node.

30% of the time less than 50 GB (10%) of memory free.

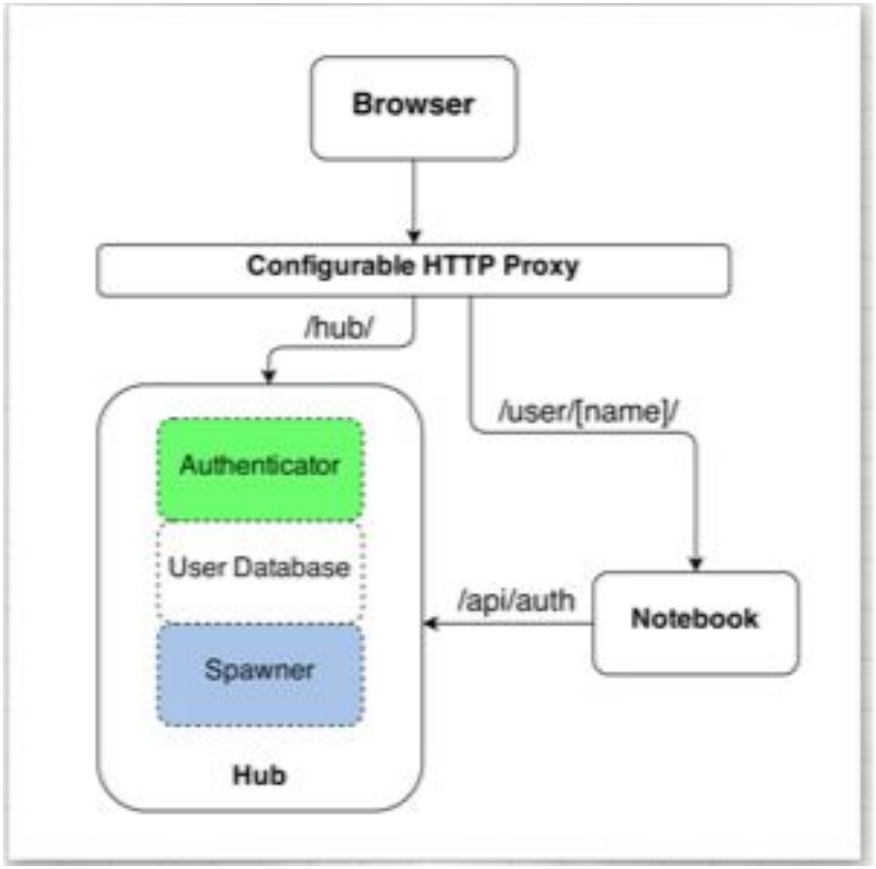
Needed new+more resources. Needed better management of the service.



Jupyter @ NERSC Architecture



JupyterHub Architecture



Components are highly abstracted:

- Authenticator
- Spawner
- Proxy

Pieces we've created:

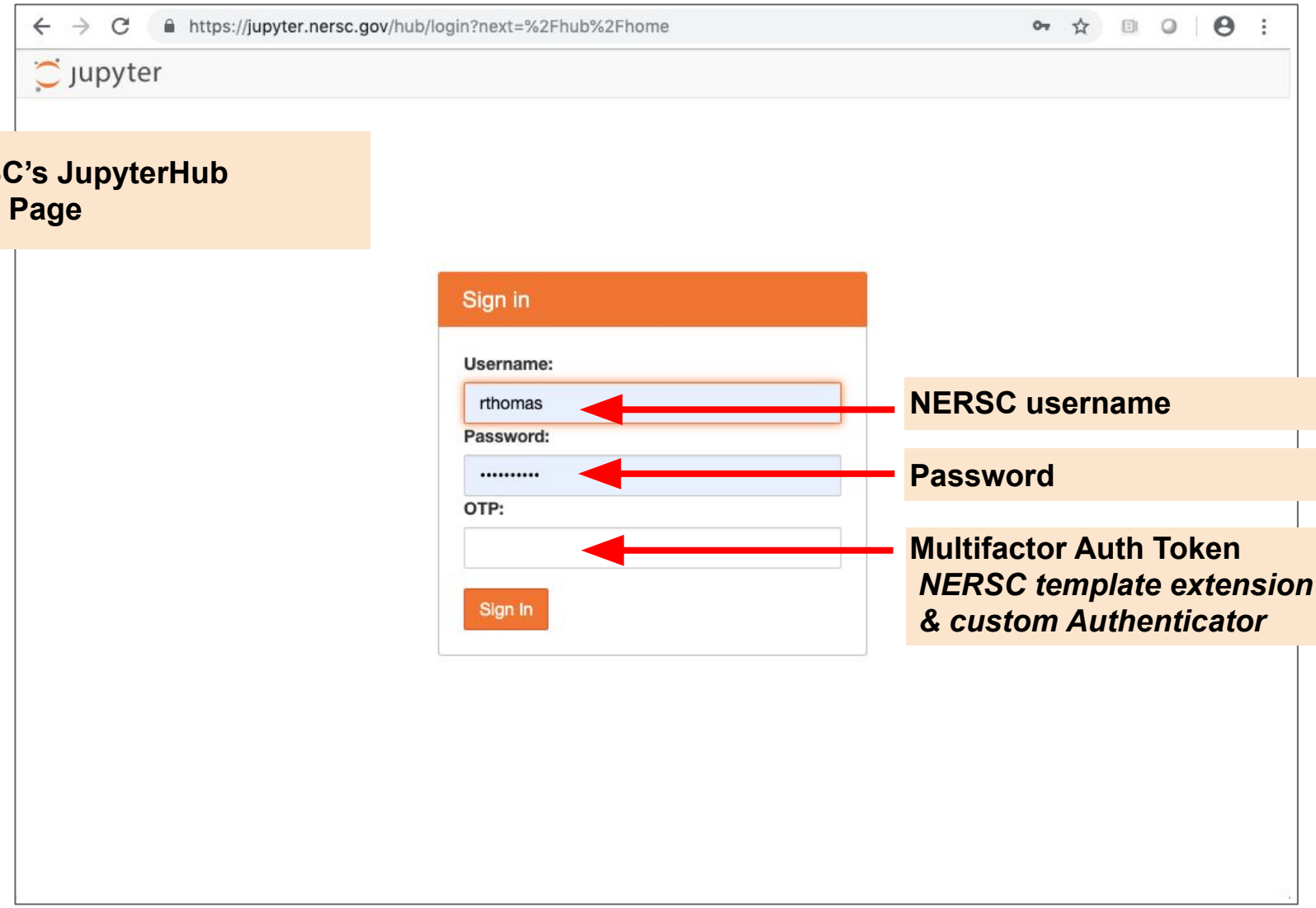
- GSIAuthenticator (retired)
- SSHAPIAuthenticator

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

Pieces we re-use/adapt and love:

- WrapSpawner (NERSCSpawner)
- BatchSpawner (NERSCSlurmSpawner)

Jupyter @ NERSC Architecture



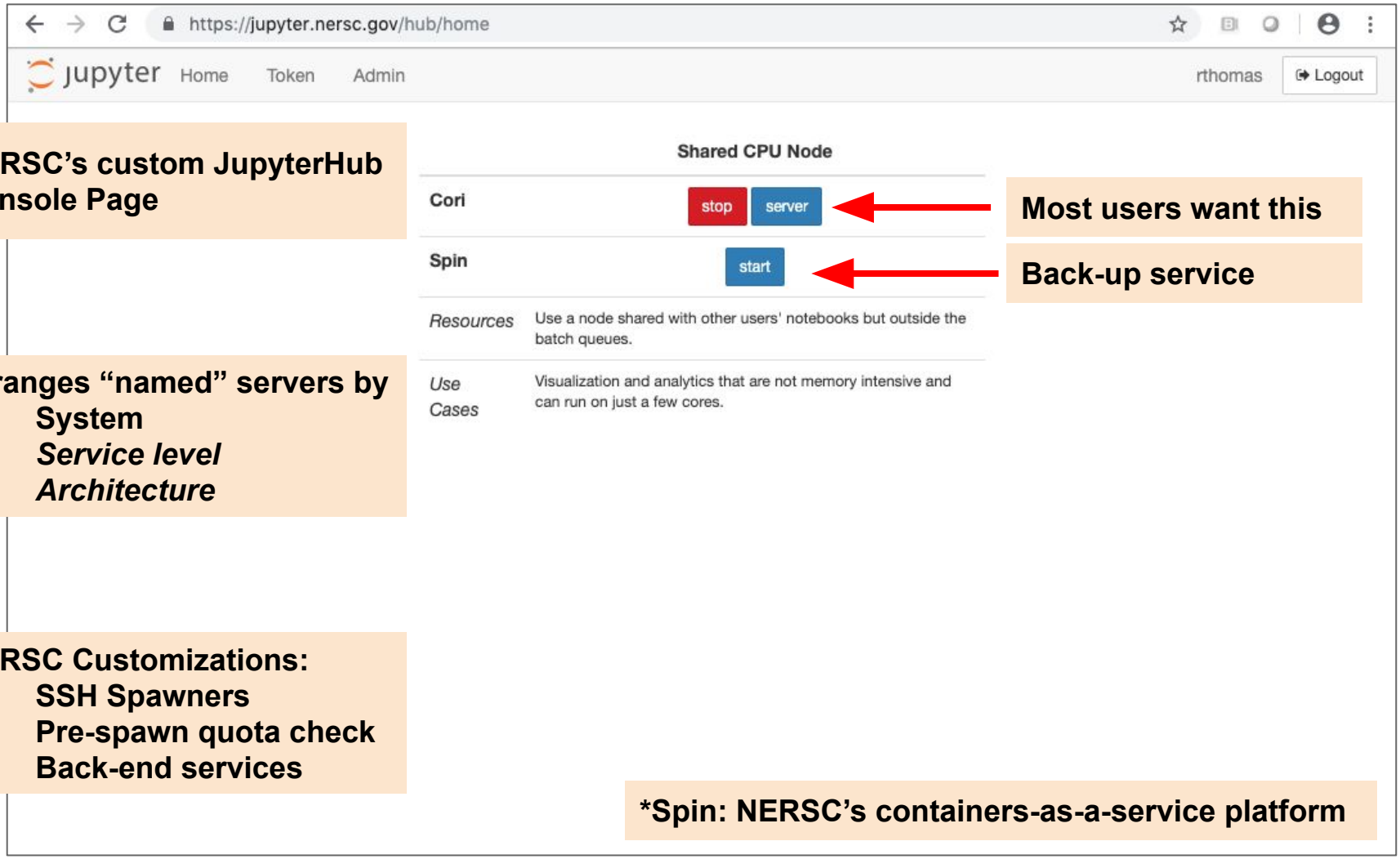
NERSC's JupyterHub Login Page

NERSC username

Password

Multifactor Auth Token
NERSC template extension & custom Authenticator

Jupyter @ NERSC Architecture



NERSC's custom JupyterHub Console Page

Most users want this

Back-up service

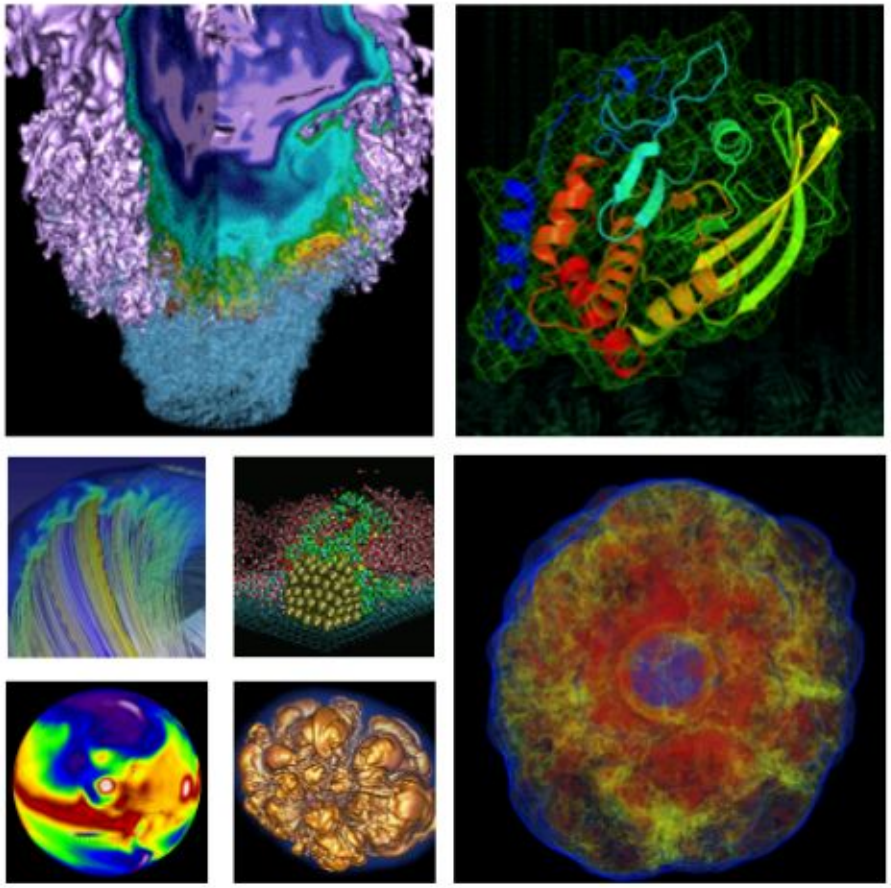
Arranges "named" servers by
- **System**
- ***Service level***
- ***Architecture***

NERSC Customizations:
- **SSH Spawners**
- **Pre-spawn quota check**
- **Back-end services**

***Spin: NERSC's containers-as-a-service platform**

Jupyter at NERSC

Use cases
Extensions
Customizations



Rollin Thomas
Data and Analytics Services
NERSC, LBNL

Shreyas Cholia
Usable Software Systems
Computational Research Division/NERSC, LBNL

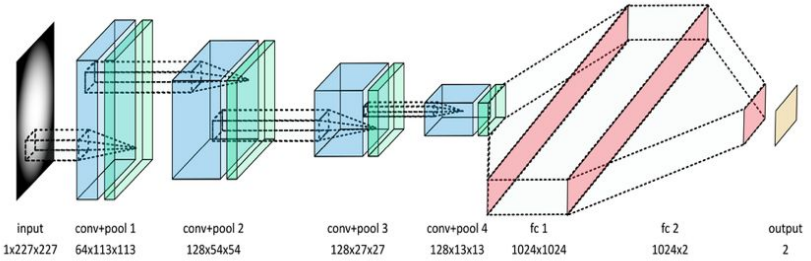
Blue Waters Webinar
April 24 2019

Jupyter in Distributed HPC

Use Cases

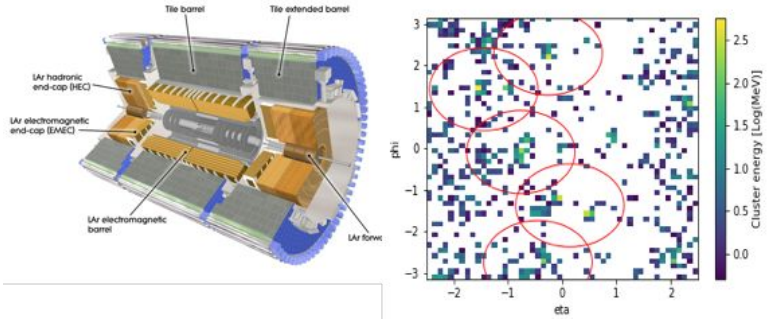
- ATLAS - Distributed training and hyper-parameter optimization
- ALS and NCEM - Image processing pipelines
- TARDIS - Supernova Data exploration and comparison of simulation / experimental data

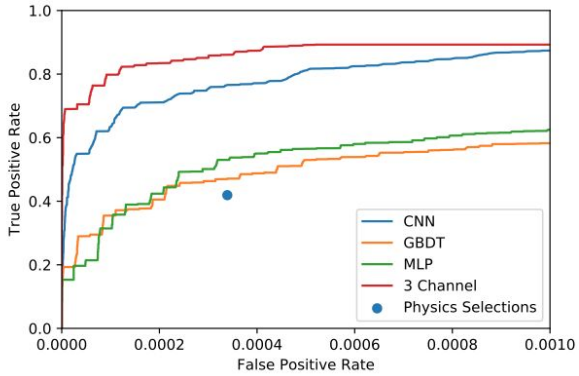
Deep learning Use Case



- **Neutral Networks (NN) with multiple layers**
  - Highly non-linear; many weights
  - Enabled by computing and architectures (e.g.(CNN))
  - Leverage Industry/academic progress for science
- **Use case for this work: whole LHC detector CNN classification**
  - Bin detector signals to form 64x64(x3) image
  - Classification problem: New Physics RPV Susy (signal) vs. Known physics (QCD) (background)

https://arxiv.org/abs/1711.03573





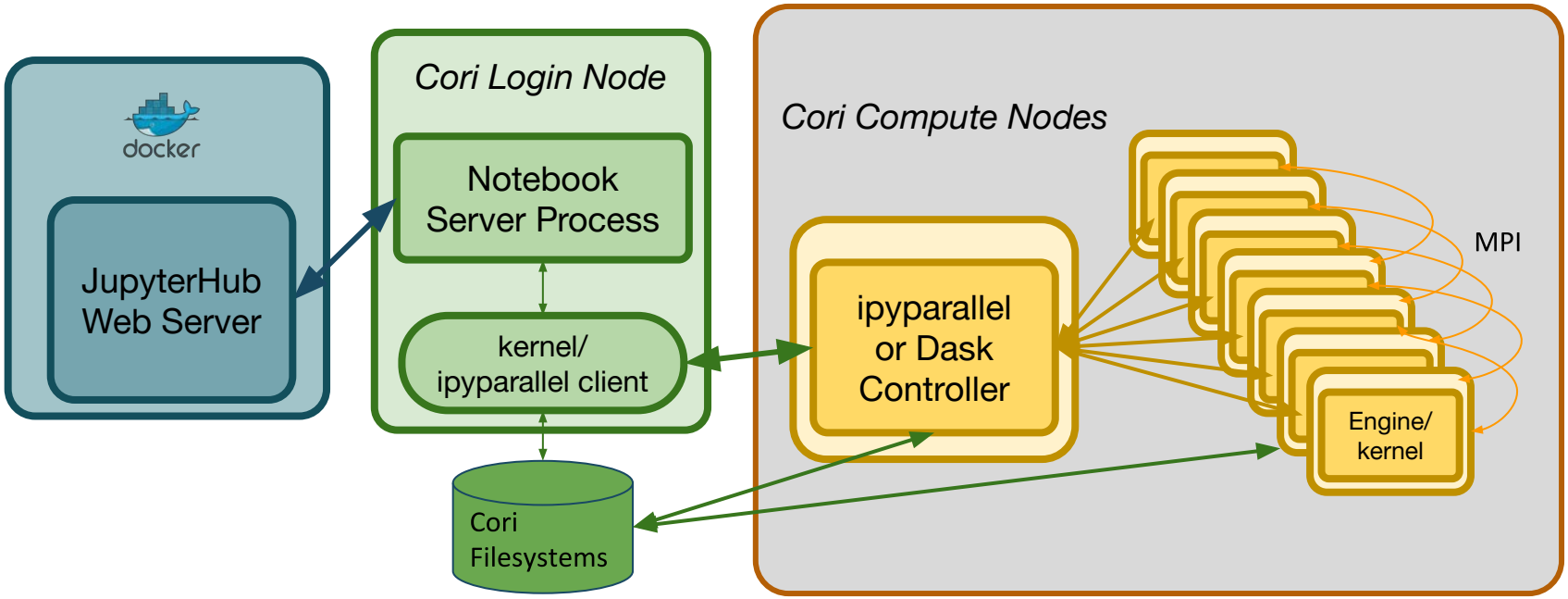
Why Interactive Distributed Deep Learning?

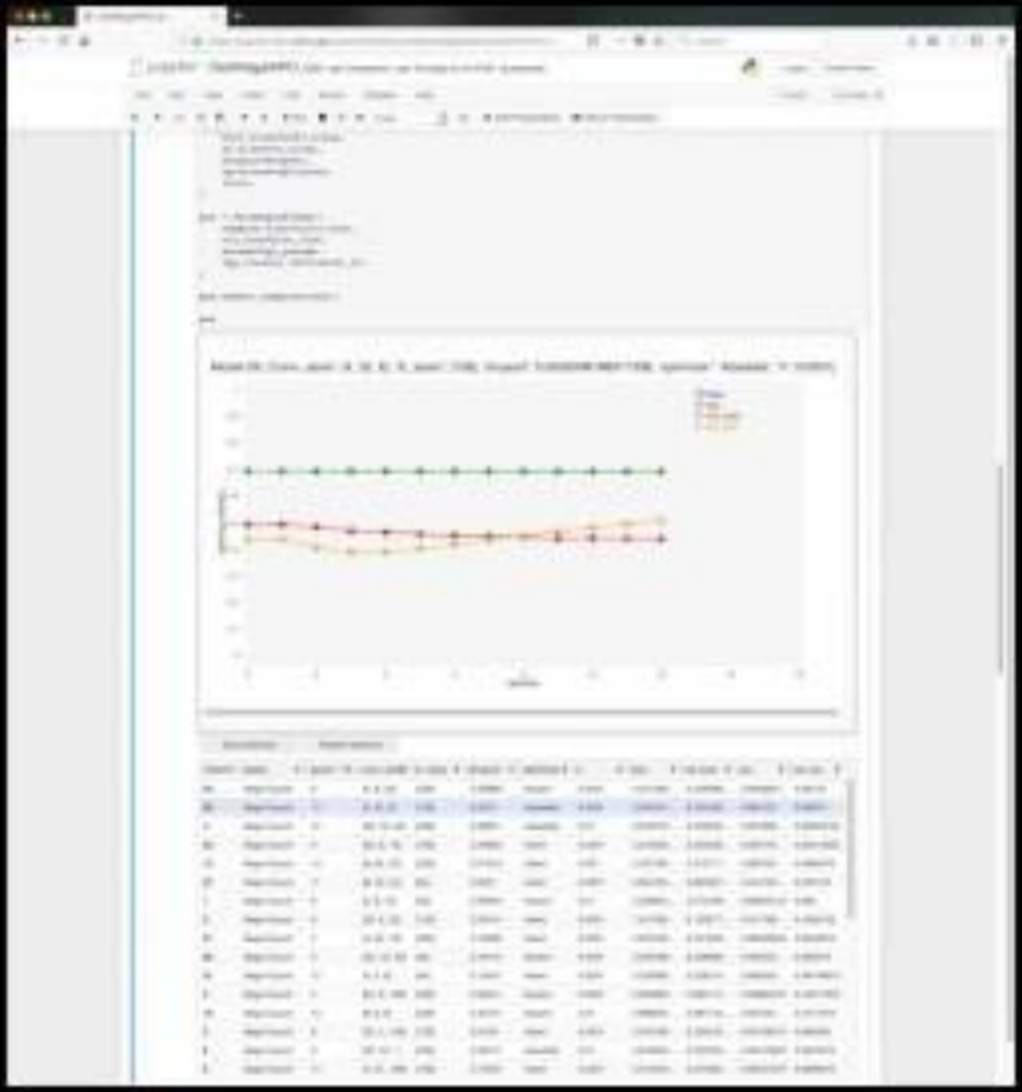
- Deep learning can enable scientific discovery
- Training of complex networks can take days
- Architecture design and parameter choice is an iterative process aided by human intuition, brute-force scans and automated optimization
- Batch HPC submission means many slow iteration cycles
- Most DL frameworks are python-based: iterating within Jupyter notebooks popular development environment for analytics

=> Interactive Distributed Deep Learning Using Jupyter!

Distributed Learning 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 Plugin**

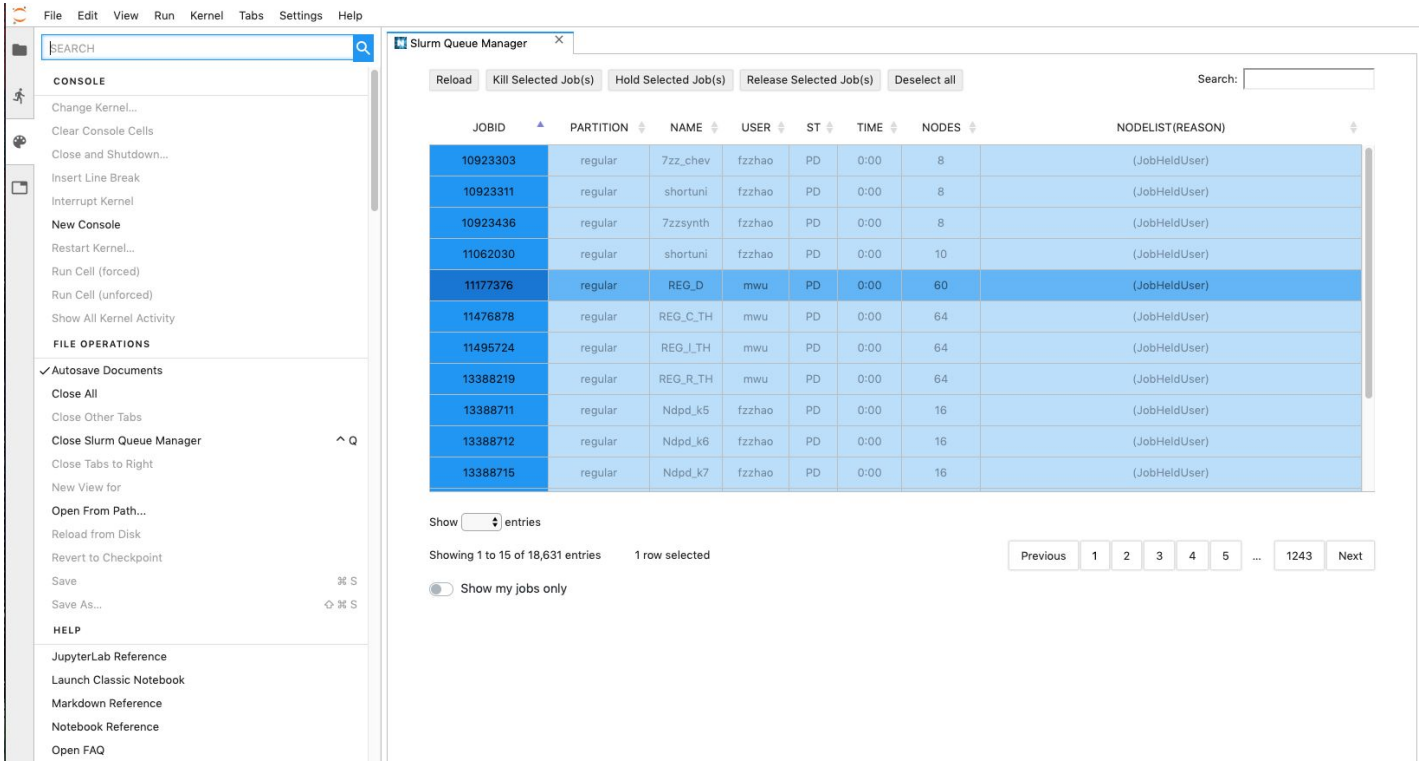




JupyterLab SLURM

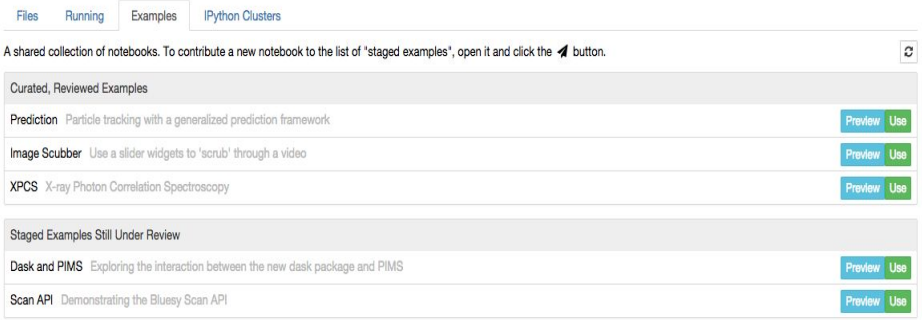
A JupyterLab extension that interfaces with the Slurm Workload Manager, providing simple and intuitive controls for viewing and managing jobs on the queue

https://github.com/NERSC/jupyterlab-slurm



Future Work

- Generalize work done in ATLAS DL case
- JupyterLab Tools and IPyWidgets for users to manage interactive compute tasks and viz through Jupyter
- More Better Parallel Computing (Dask, IPyParallel ...)
- JupyterLab extensions for SLURM
- Superfacility Integration - manage workflows distributed across facilities through Jupyter
- Project curated examples Notebooks that can be cloned and modified. Extend to cloned data/environments



Future Work

Delivery in late 2020

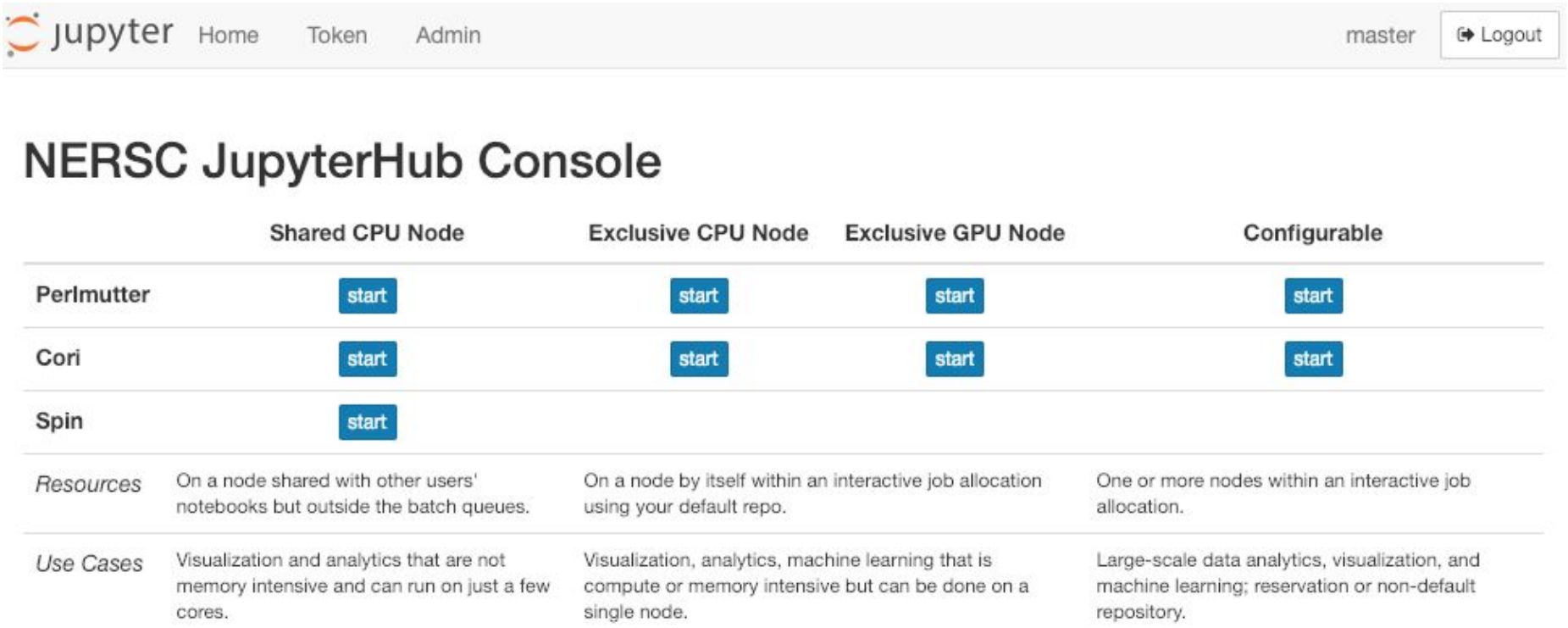
Cray Shasta System providing 3-4x capability of Cori system

First NERSC system designed to meet needs of both large scale simulation and data analysis from experimental facilities

- Includes both NVIDIA GPU-accelerated and AMD CPU-only nodes
- Cray Slingshot high-performance network will support Terabit rate connections to system
- Optimized data software stack enabling analytics and ML at scale
- All-Flash filesystem for I/O acceleration

Robust readiness program for simulation, data and learning applications and complex workflows

Future Work


jupyter
Home
Token
Admin
master
Logout
NERSC JupyterHub Console
Shared CPU Node
Exclusive CPU Node
Exclusive GPU Node
Configurable
Perlmutter
start
start
start
start
Cori
start
start
start
start
Spin
start
Resources
On a node shared with other users' notebooks but outside the batch queues.
On a node by itself within an interactive job allocation using your default repo.
One or more nodes within an interactive job allocation.
Use Cases
Visualization and analytics that are not memory intensive and can run on just a few cores.
Visualization, analytics, machine learning that is compute or memory intensive but can be done on a single node.
Large-scale data analytics, visualization, and machine learning; reservation or non-default repository.

Future Work

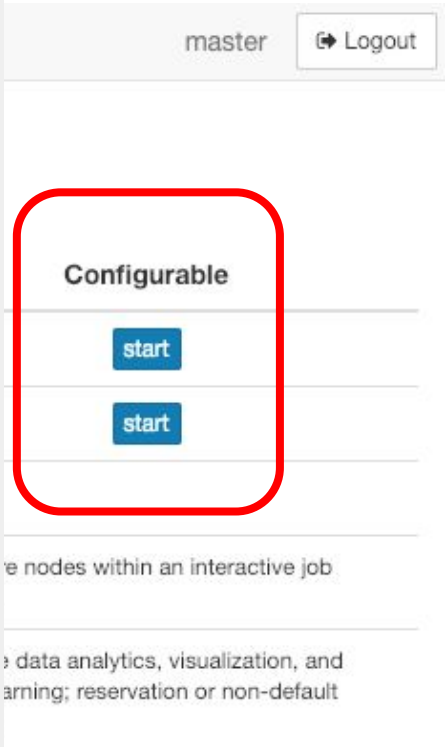
JupyterHub needs to know and validate:

- What accounts can this user charge to?
- What queues can they submit to?
- How many nodes can they use?
- How long can they run jobs?
- What containers can they run on each machine?
- Do they have any reservations?
- What about persistent burst buffer reservations?
- … and other kinds of resources we have or think of?

How? Need REST API(s) for:

- Center resources, accessibility, & even availability.
- User's resources (my images, my reservations, etc).
- *(… what if each center had standard APIs for this?)*

Many parts are there, they just don't talk to each other yet.



Acknowledgements

Big Thanks to the Community!

- MSI
- TACC
- SDSC
- Jupyter Team