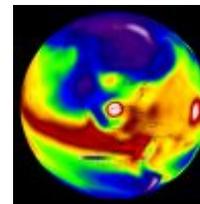
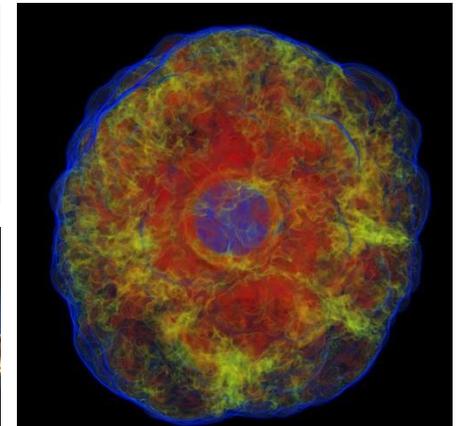
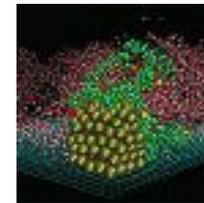
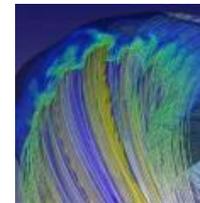
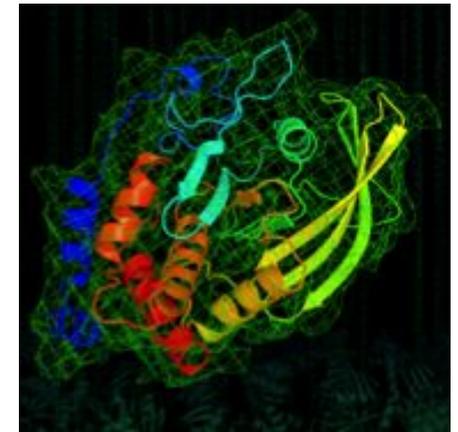
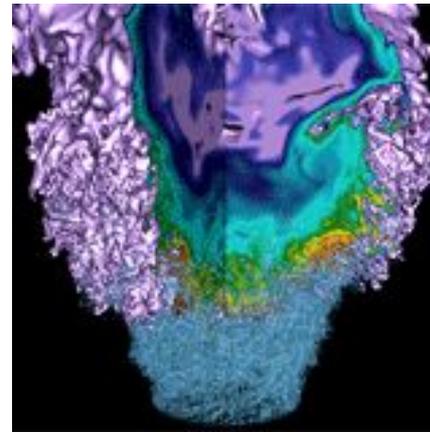


Interactive Distributed Computing with Jupyter and Friends

2019 Pangeo Community Meeting



Shreyas Cholia
scholia@lbl.gov

Lawrence Berkeley National Laboratory

Why Interactive Distributed Deep Learning?

- Using CNNs for Atlas LHC Image Data Classification
- Human in the loop Hyper Parameter Optimization and Distributed Training

[IPython Parallel \(ipyparallel\)](#)

- Hub and Controller communicate with a set of engines (ipython kernels)
 - E.g. with LoadBalancedView for 'destination agnostic' scheduling
 - Currently single controller bottleneck but only for notebook communication
 - use other MPI libraries (eg. Horovod, mpi4py) for bulk communication alongside

[QGrid](#)

- Quantopian widget for interactive tables

[BQPlot](#)

- Interactive Visualization of Results

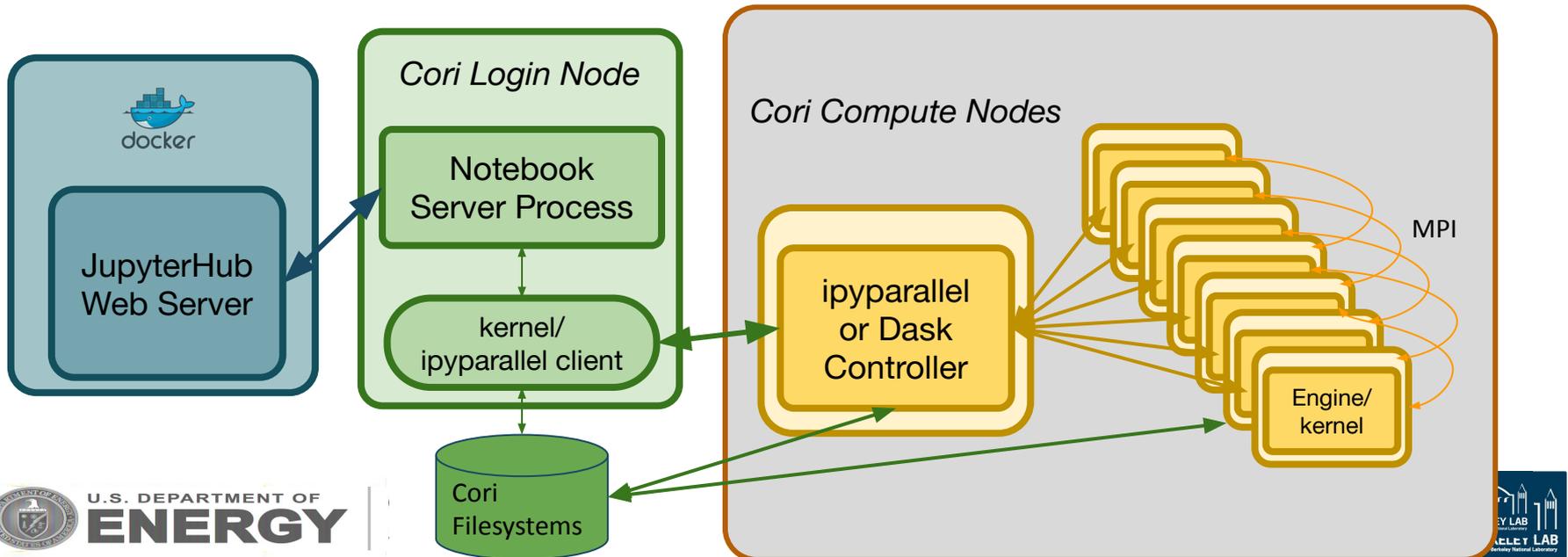
[Kale](#)

- Our manager and worker service that wrap backend task giving with fine-grain task control and node resource monitoring

NERSC 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 Plugin**

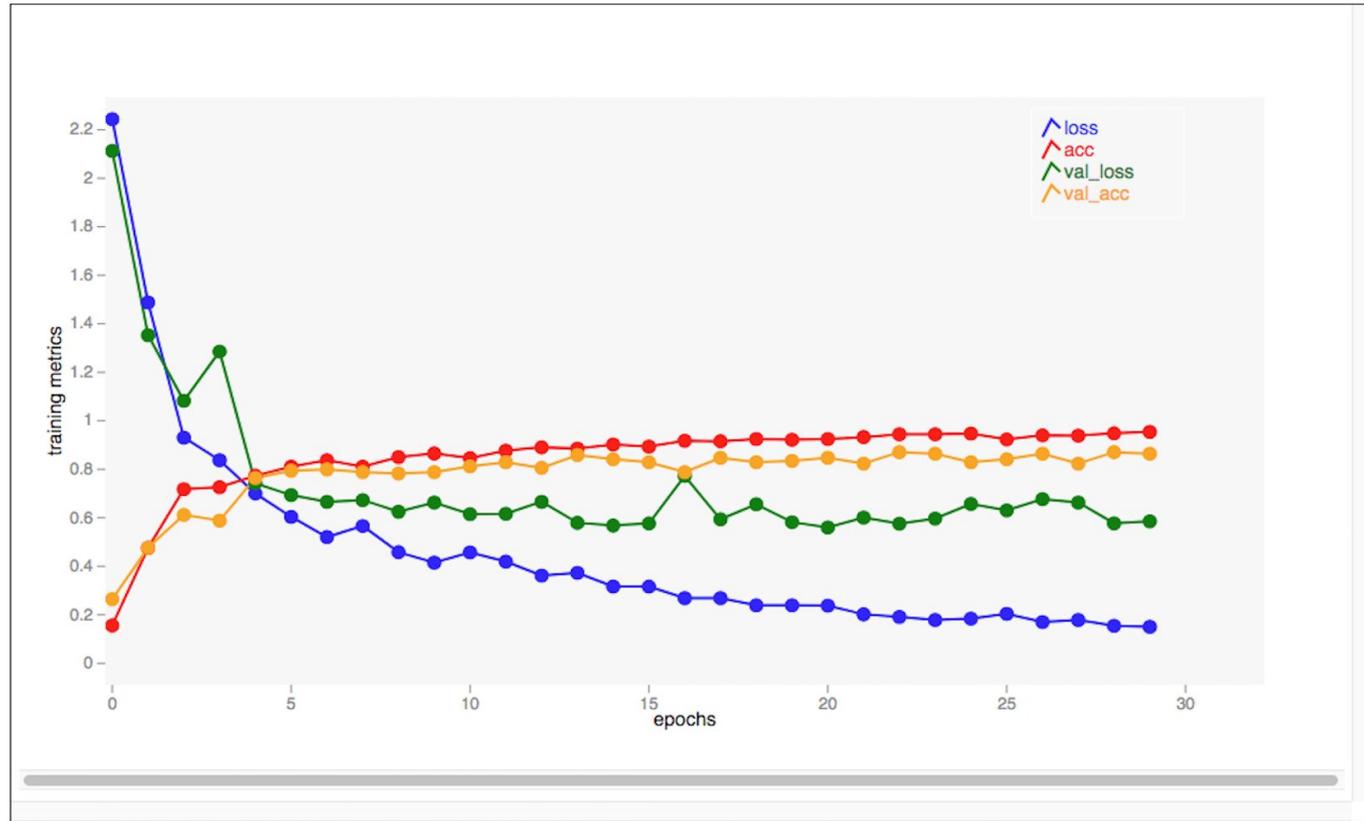


Plots update live
Table shows
different
configurations:

- Status
- Current loss and accuracy
- Sort

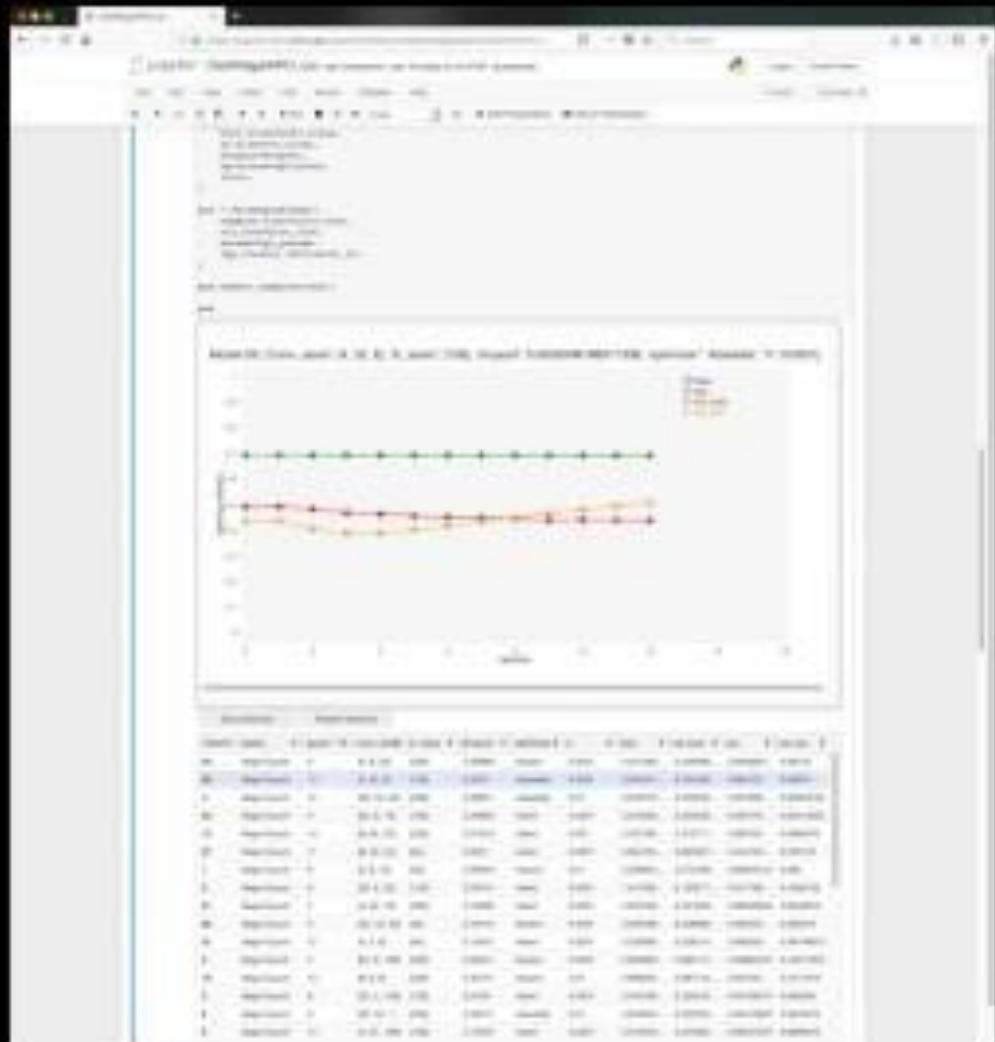
Can add further
quantities to plot
and interaction
buttons

<https://github.com/sparticle/steve/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	Adadelata	0.157987...	0.579903...	0.944578...	0.870588...
1	Ended Tra...	31	4	8	16	0.03825	Adadelata	0.151153...	0.585079...	0.954216...	0.864705...



Other Use Cases: NCEM



- Interactive exploration and analysis of electron microscope images via Jupyter
- Serial processing of 4D image arrays in numpy - looking to bump this up via Dask/IPP
- Achieved 20x speedup on NCEM Py4DSTEM Notebook running on HPC resources at NERSC

<https://github.com/py4dstem/py4DSTEM/blob>

```
All DPs
In [9]: # Get peaks
corrPower = 0.8
sigma = 2
edgeBoundary = 20
maxNumPeaks = 70
minPeakSpacing = 50
minRelativeIntensity = 0.001
verbose = True

peaks = find_Bragg_disks(dc, probe_kernel.data2D,
                        corrPower=corrPower,
                        sigma=sigma,
                        edgeBoundary=edgeBoundary,
                        minRelativeIntensity=minRelativeIntensity,
                        minPeakSpacing=minPeakSpacing,
                        maxNumPeaks=maxNumPeaks,
                        verbose=verbose)
```