The Wisconsin Analysis Facility

Ryan Simeon

9 September 2025

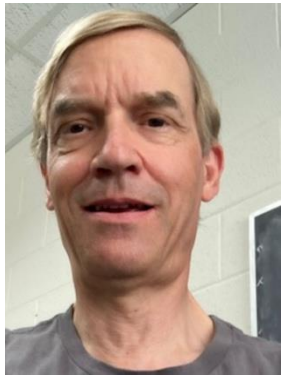
https://cms01.hep.wisc.edu

People Behind the Analysis Facility

- Professors
  - Tulika Bose
  - Sridhara Dasu
- Computing Operations & Support
  - Dan Bradley
  - Ajit Mohapatra
  - Chad Seys
  - Carl Vuosalo
- Grad Students
  - **Ryan Simeon** (via TAC-HEP traineeship)
- + our users!



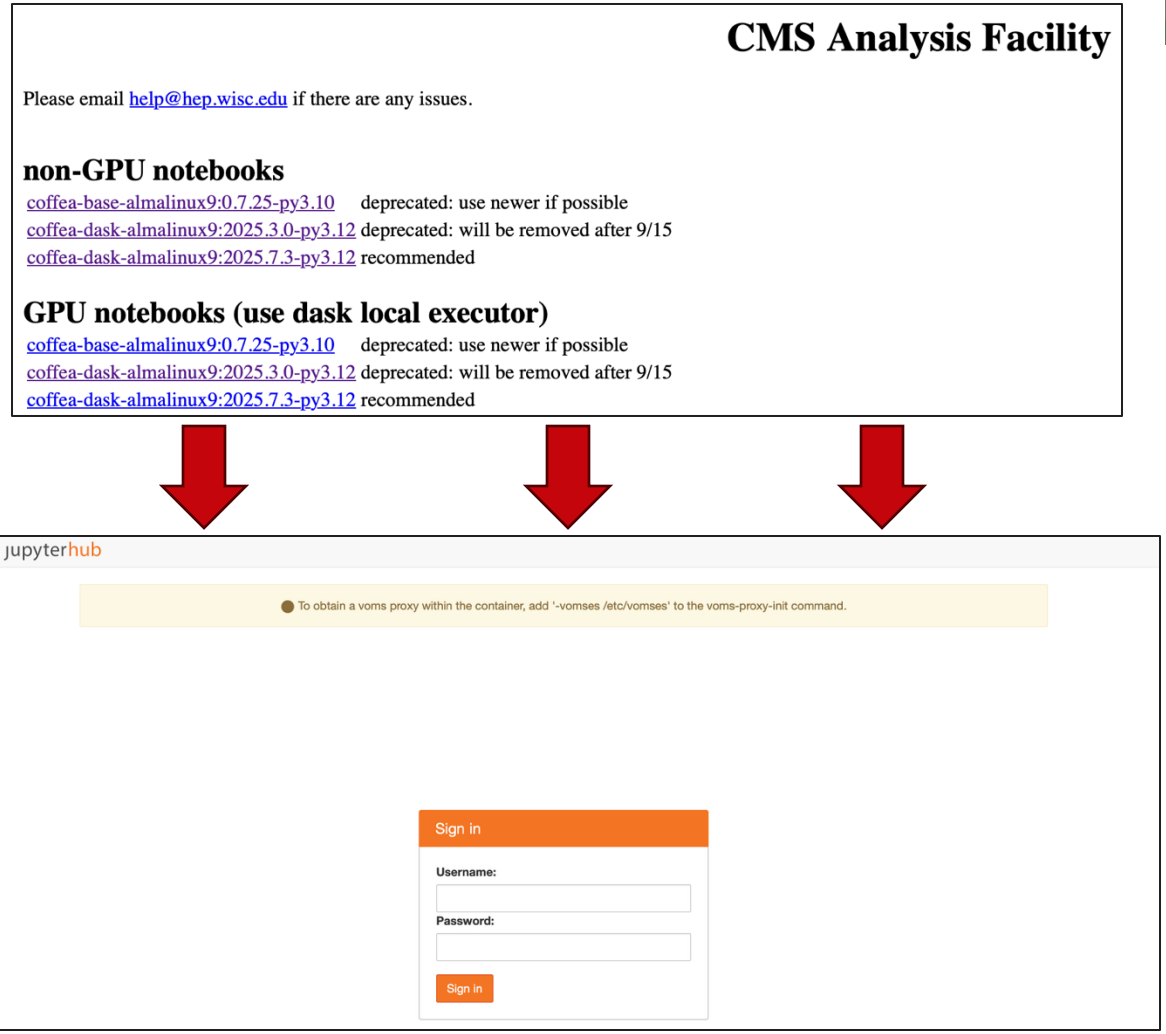


User Experience

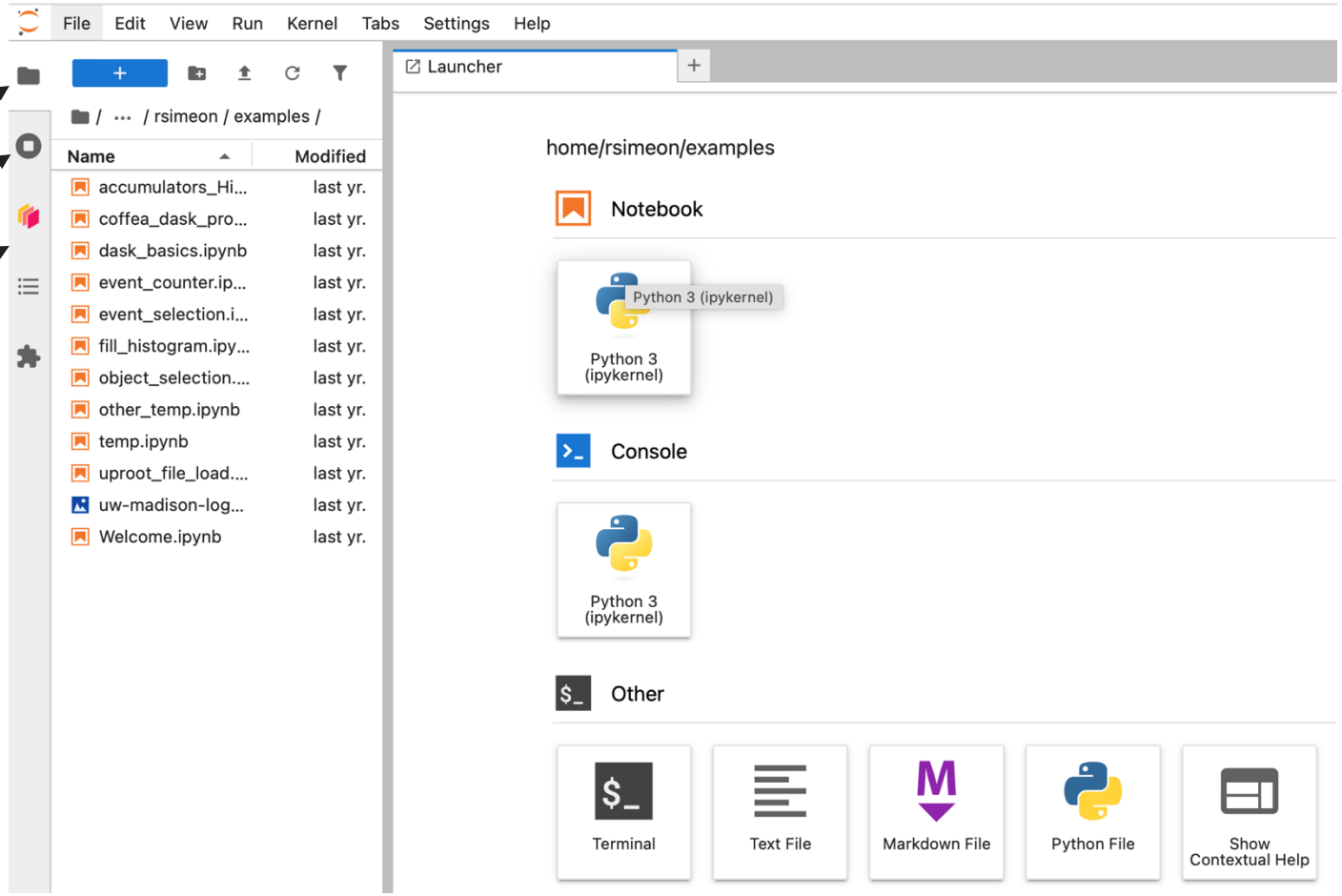
Logging In and Image Selection

- Analysis Facility accessible via web interface at https://cms01.hep.wisc.edu
- On entry, users can choose from a small collection of images, corresponding to different coffea versions
- Then, users log in
- Space here for news, announcements, and reminders
- Currently working on enabling CERN SSO login



Jupyterlab Interface

- Users presented with Jupyterlab interface
- Easy…
  - file navigation
  - kernel monitoring
  - Dask access
- Native notebook support
- Can interact via terminal, text editors as well
  - Useful for the experiment (with notebooks) → scale → reproduce (scripts/config files/utility files) workflow



Using Tier-2 Resources at Wisconsin

- Wisconsin AF provides access to a Condor cluster
  - Includes UW Tier-2 resources
  - More on this later
- Given a Dask client, very easy to move coffea code to run on a cluster
- cowtools package
  - One use is to make getting the Dask client easy – just one line
  - Allows configuration of cluster: max # workers, memory requirements, etc.
  - Installed by default in users' base environments
  - Custom-made for Wisconsin's computing setup, so users do not need to know about eg: where scratch directories should be, how to handle their x509 proxies, etc.
  - https://github.com/rpsimeon34/cowtools/

Moving to the cluster in just one line!

```
client = cowtools.GetCondorClient(max_workers=MAX_WORKERS)
```

```
[6]: client
```

[6]:

Client

Client-8c07d586-8cca-11f0-b3ad-3cecefda85c8

Connection method: Cluster object **Cluster type:** dask_jobqueue.HTCondorCluster

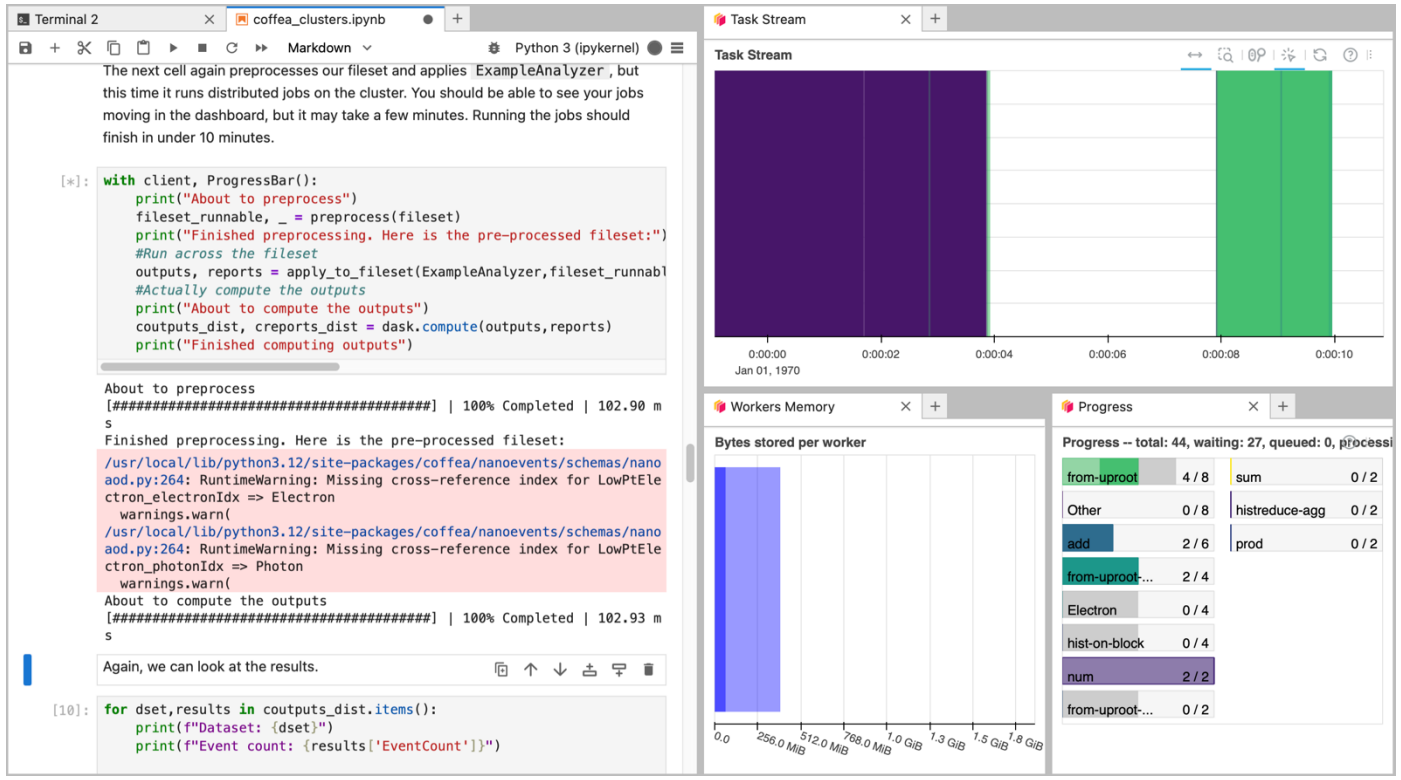
Dashboard: proxy/3083/status

Launch dashboard in JupyterLab

▸ Cluster Info

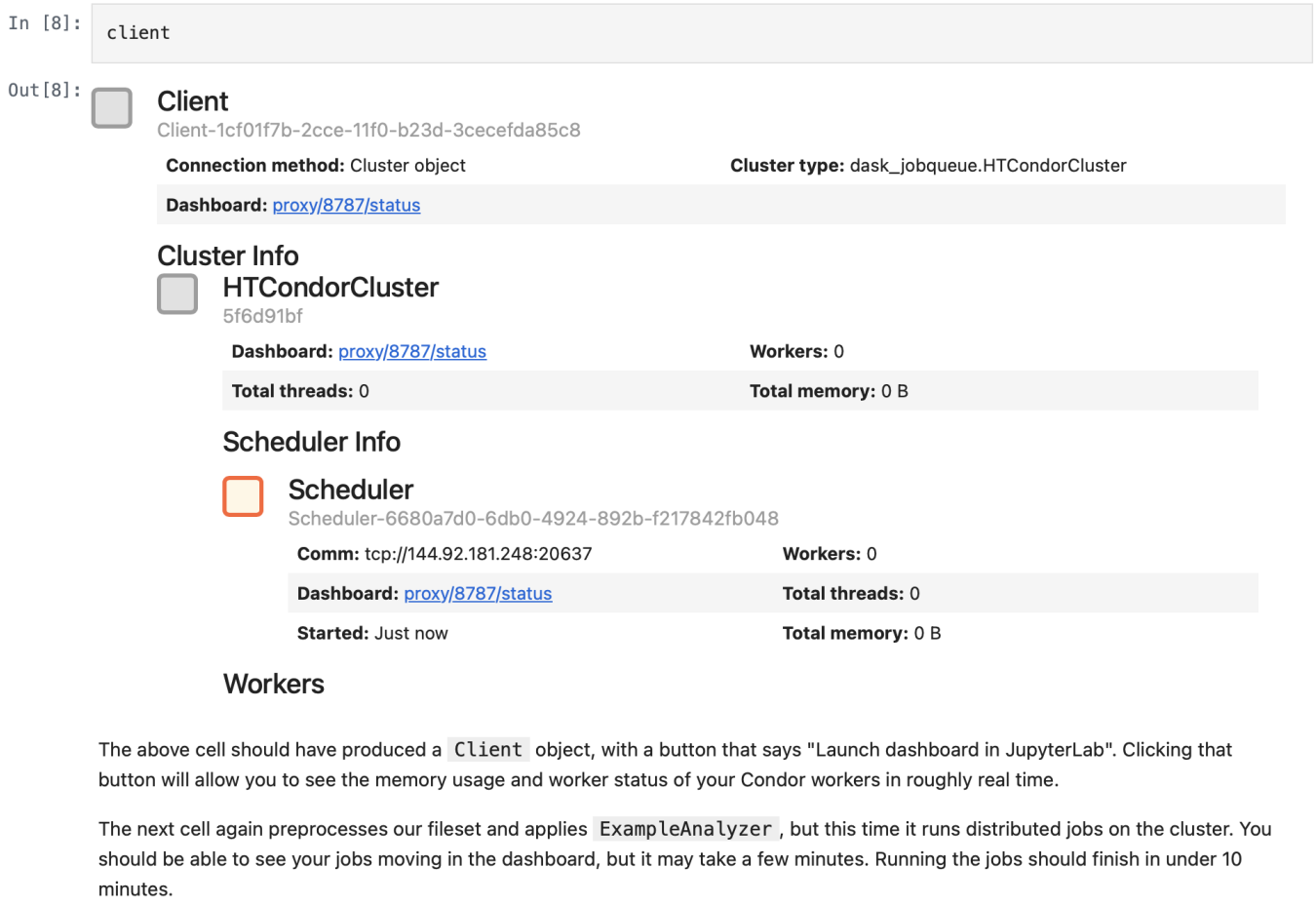
Using Tier-2 Resources at Wisconsin

- When running with notebooks, live Dask dashboard is available
- Also possible to save Dask report as html to examine later
  - Jupyterlab conveniently lets you examine and interact with saved html



Documentation

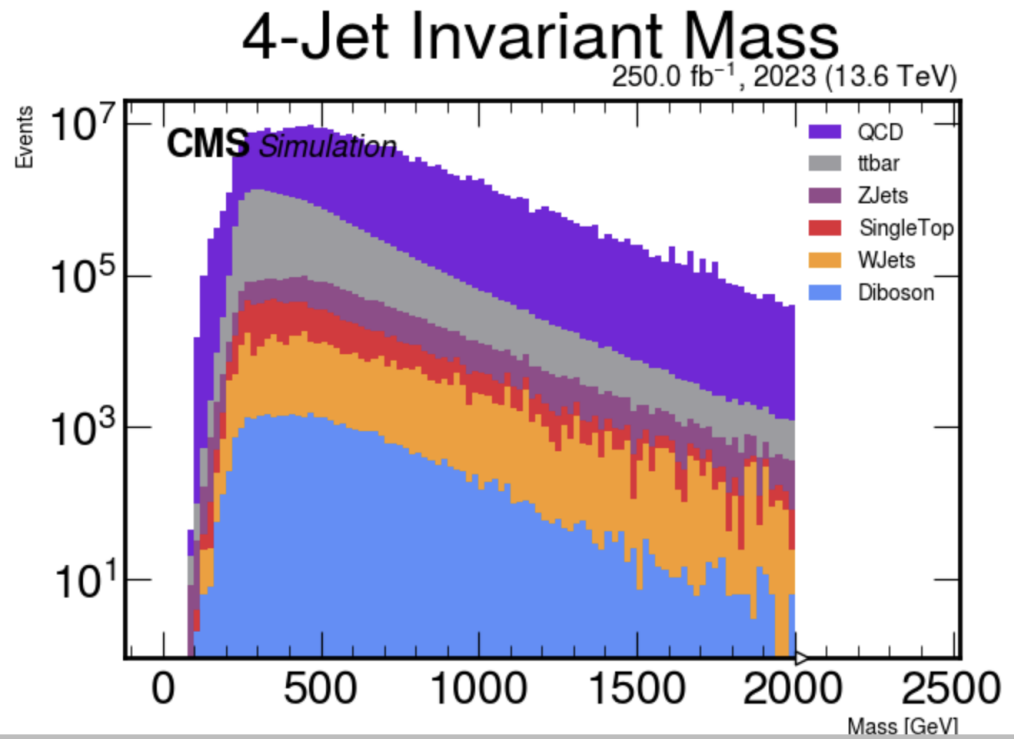
- Heavily annotated examples are available in a Github repository
  - https://github.com/rpsimeon34/wisc-af-examples/
- Can be read as rendered notebooks via Github, or of course cloned and interacted with
  - Let users experiment and gain confidence
- Shows users how to navigate the Juptyerlab interface, configure proxies, use cowtools, etc.
- Intended to take users from being completely new to the AF to being able to fill histograms with code that runs at scale on the cluster



Experience in Real Analysis Work

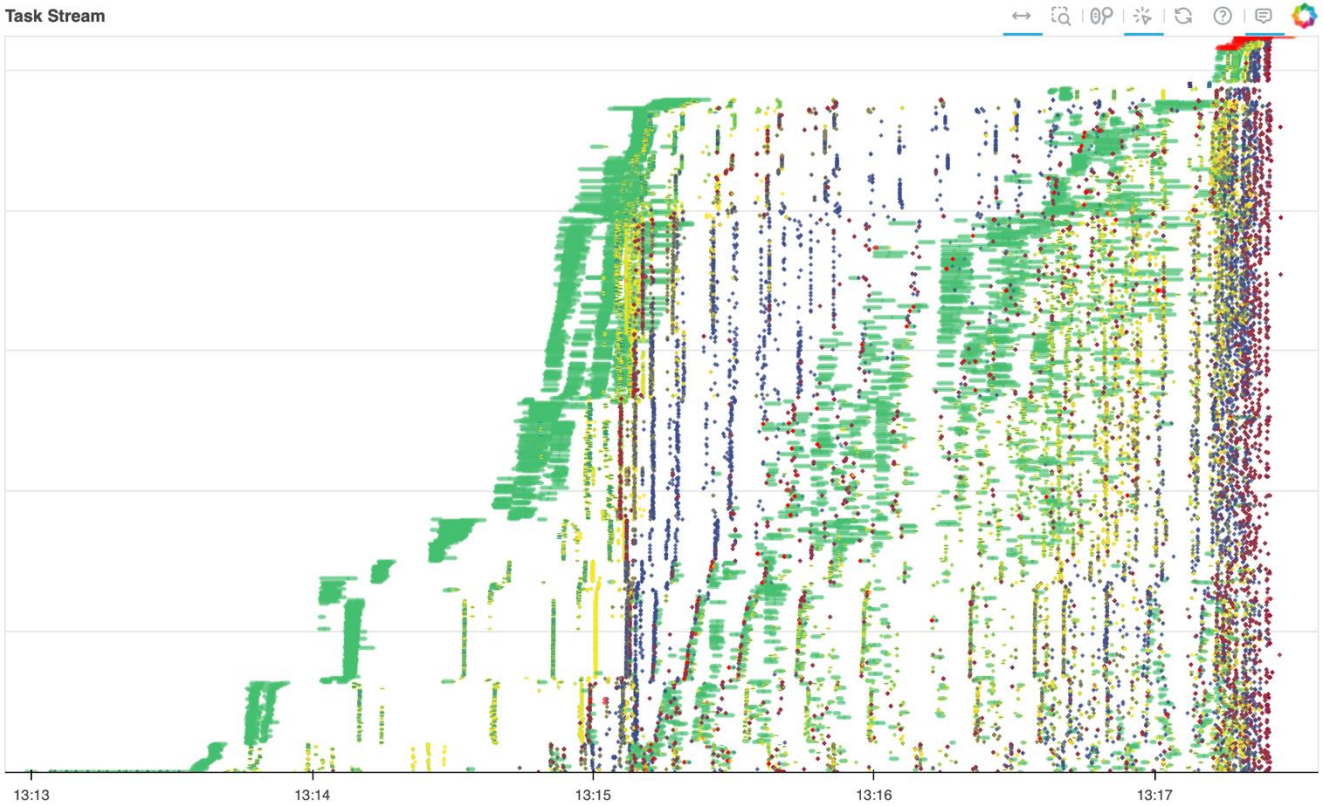
- I've been using the AF for my analysis work for ~2 years
- Been working primarily with samples totaling 756.8 GB
- AF covers many use-cases:
  - Quick, local work for development, experimentation, debugging
  - Medium-duration (coffee break) distributed computing
  - Hours-long tasks, like ML hyperparameter optimization
  - Plotting with quick turnaround (aesthetics and histogram slicing/rebinning)

```
[4]: plot_1d("4-Jet Invariant Mass",
        #sgl_hists=[mc_4jet_mass["Signal"]["4JetMass"]],
        #sgl_label=["Signal"],
        bkg_hists=[mc_4jet_mass[dset]["4JetMass"] for dset in mc_4jet_mass.keys() if dset != "Signal"],
        bkg_label=[dset for dset in mc_4jet_mass.keys() if dset != "Signal"],
        xlabel="Mass [GeV]",
        year=2023,
        lumi=int(LUMI),
        legend_xoffset=0.02,
        logy=True,
        signal_sf=10_000)
```



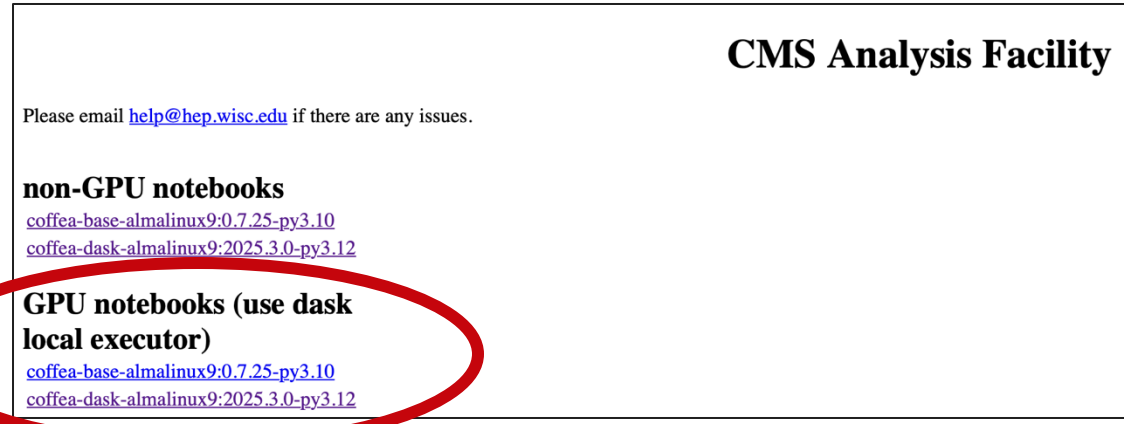
Experience in Real Analysis Work

- Very high availability of Condor workers
  - Regularly can get at least 1000 workers
  - Usually limited by the parallelizability of my code, rather than by resources
- Dashboards very helpful for monitoring memory usage, parallelizability, etc.



Using GPUs

- One of our servers hosts 2 GPUs
  - Interactive access to GPUs, good for developing GPU-accelerated code
- We also have GPUs available via Condor cluster
  - Integrates well with PyTorch + coffea, seen in personal use for analysis
  - More details later about GPUs in cluster
- One other UW user also already using these GPUs for DNNs – but we can support more!



CMS Analysis Facility

Please email help@hep.wisc.edu if there are any issues.

non-GPU notebooks

coffea-base-almalinux9:0.7.25-py3.10
coffea-dask-almalinux9:2025.3.0-py3.12

GPU notebooks (use dask local executor)

coffea-base-almalinux9:0.7.25-py3.10
coffea-dask-almalinux9:2025.3.0-py3.12

Technical Details/Under the Hood

Apptainer and Jupyterlab

- Environment provided with Apptainer images
  - coffeateam's images as base
  - Jupyterlab installed with mamba on top
  - cowtools installed by default in all Apptainer images we provide
- cowtools ensures that Condor workers operate in the same environment as local kernels

Analysis Facility Servers

- We have 2 servers: one with GPUs (cmsgpu01), and one without (cms01)
- cmsgpu01
  - 128 GB RAM
  - 1x AMD EPYC 7402P 24-Core Processor, 48 threads
  - **2x Tesla T4**
  - 2 TB NVMe SSD
- cms01
  - 256 GB RAM
  - 1x AMD ECPY 7713P 64-Core Processor, 128 threads
  - 4 TB NVMe SSD

UW Campus Resources

- Resource provisioning done with HTCondor
- Tier-2 core count is ~17,500
- In addition to Tier-2 cores, we make opportunistic use of campus computing resources
  - About 20,000 total cores
  - In the past month, CMS used an average of 865 opportunistic cores (CHTC), with bursts to thousands at a time

UW Campus Resources - GPUs

- 32 GPUs (Tesla T4) within our Tier-2 available for use
- In addition, more GPUs available for opportunistic use
  - Table at right
- Lots of opportunity for users!

Opportunistically Available Campus GPUs
(not including those in Tier-2)

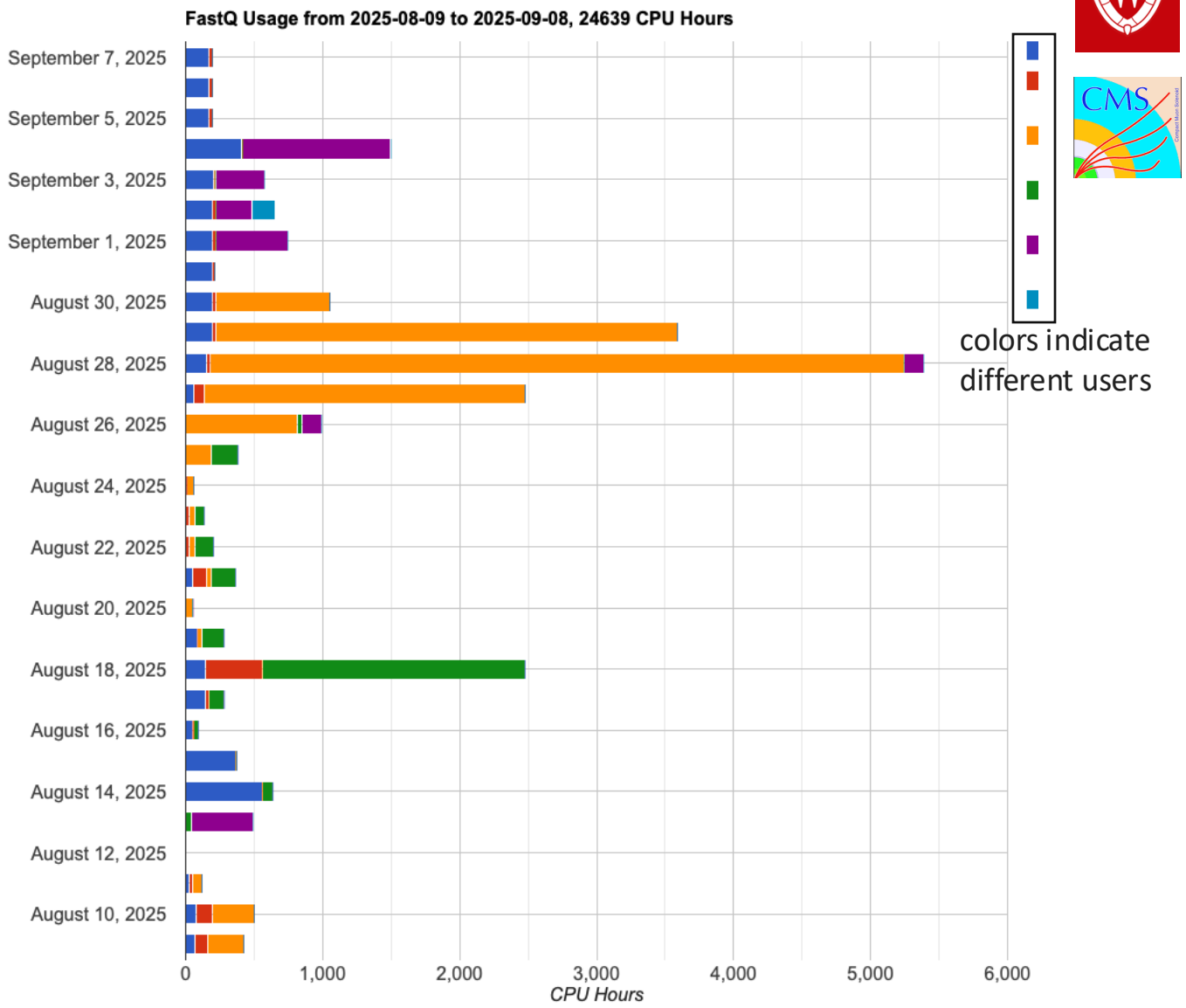
GPU Type	# Available
NVIDIA GeForce RTX 2080 Ti	14
NVIDIA A100-SXM4-80GB	14
NVIDIA L40	9
NVIDIA L40S	7
NVIDIA A100-SXM4-40GB	6
NVIDIA H200	5
Tesla P100-PCIE-16GB	4
NVIDIA H100 80GB HBM3	3
Quadro RTX 8000	1
NVIDIA GeForce GTX 1080 Ti	1
NVIDIA A40	1
NVIDIA A30	1

FastQ

- Tier-2 used for both analysis and production jobs
- To make interactive usage of Tier-2 via Condor possible, analysis jobs are put in FastQ and given higher priority
- **Jobs are not preempted**
  - Accomplished via overprovisioning
  - Ensures no jobs are killed to make room for analysis jobs

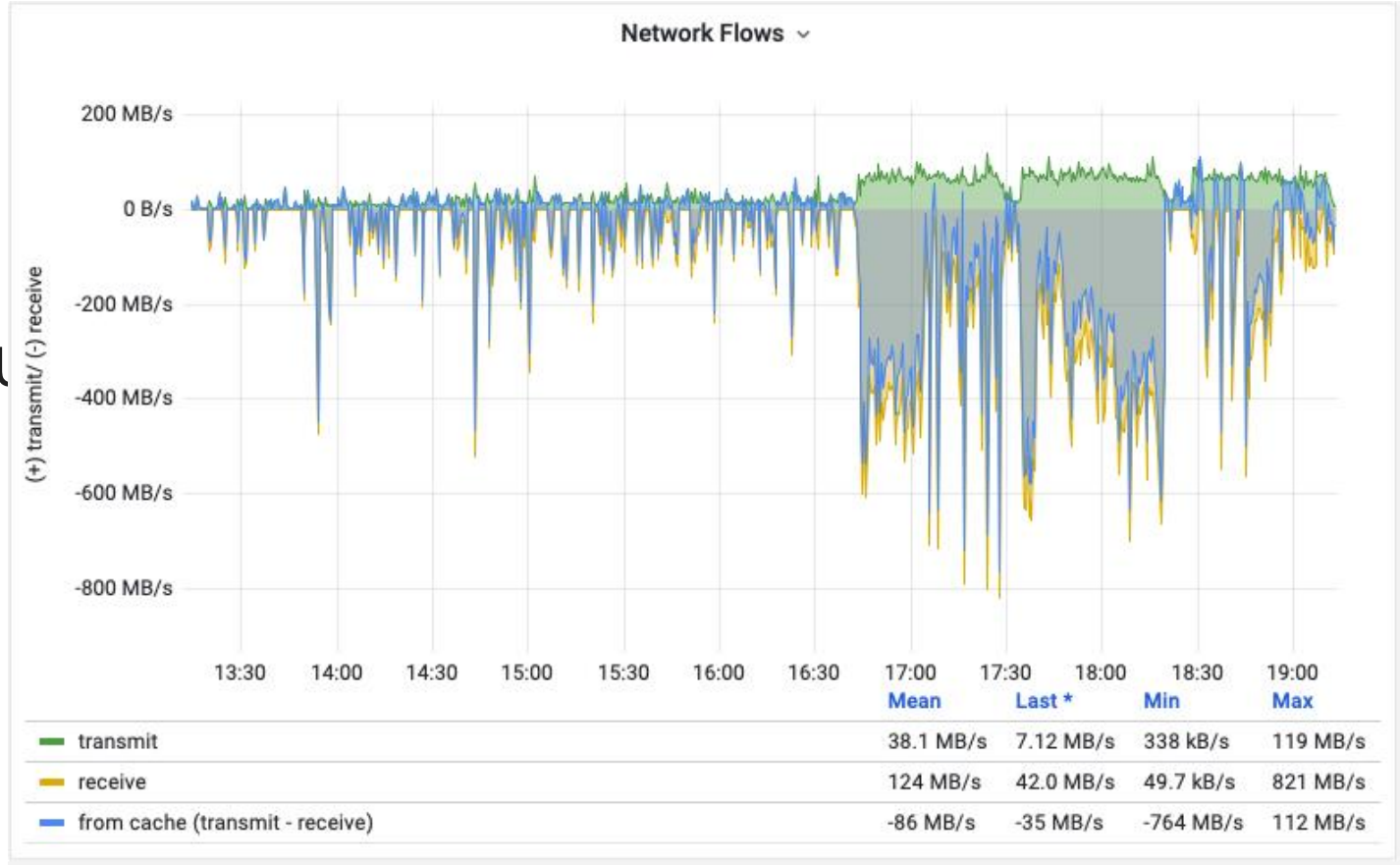
FastQ

- Tier-2 used for both analysis and production jobs
- To make interactive usage of Tier-2 via Condor possible, analysis jobs are put in FastQ and given higher priority
- Also a good measure of AF users' Condor worker consumption
- **Lots of capacity for more users**
  - Also note, high availability ⇒ better interactivity



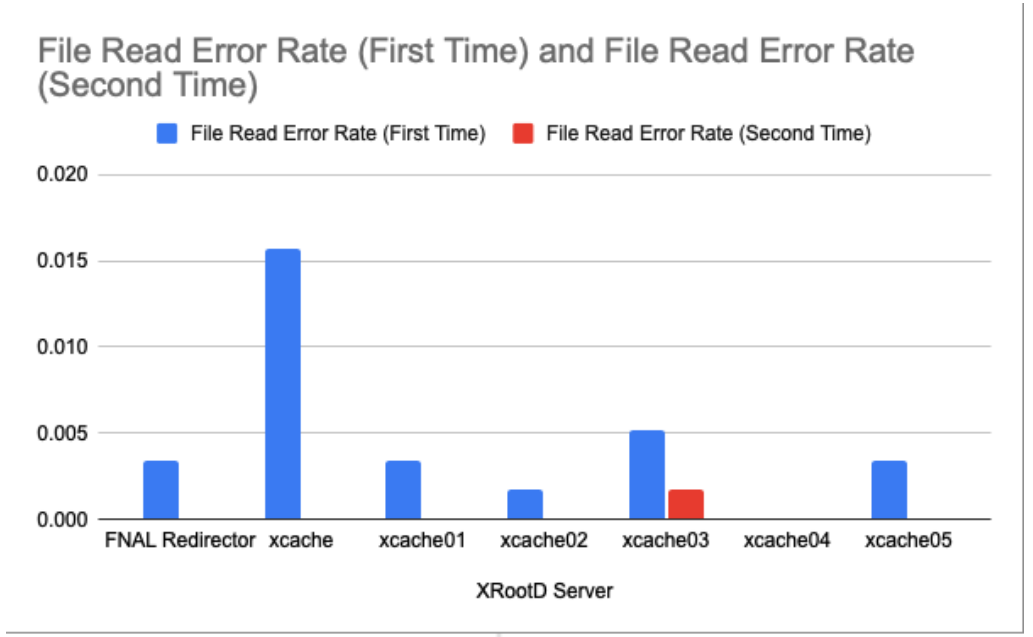
XCache

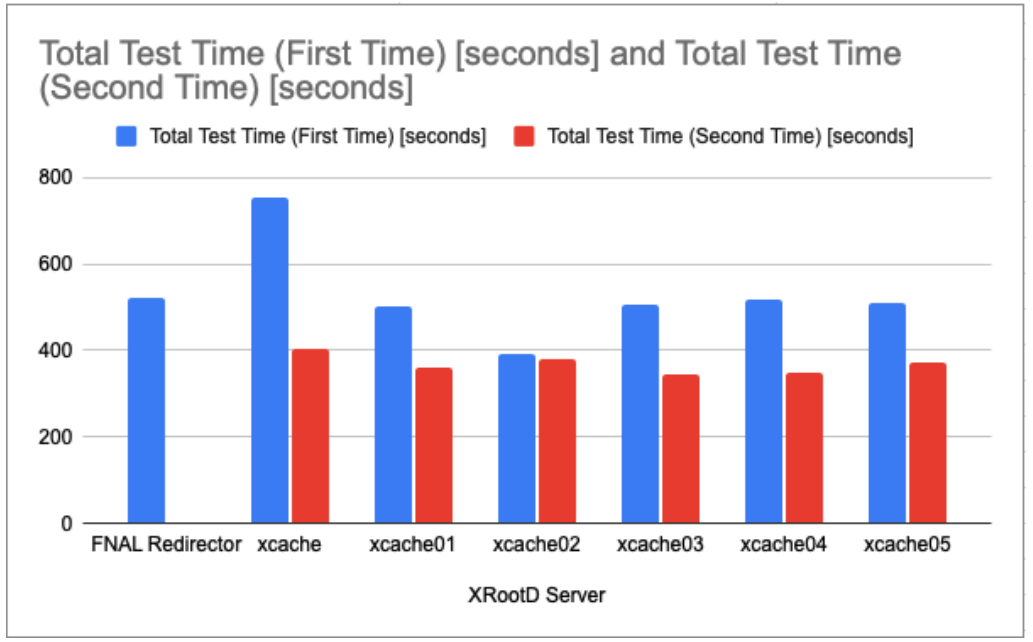
- Users access files through Wisconsin's XCache
- XCache details:
  - 5 servers, 44 TB capacity each
  - Each server has 2x Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz, 32 threads
- Usage shown at right over a few hours' period
  - "transmit": XCache serving information
  - "receive": XCache is caching information



XCache Testing

- Reads data and performs computations (histograms, floats)
- Total file size: 61 GB
- Actually reads 2.75 GB
- Total number of events: 41,402,863
- Total number of files: 575
- Uses code originally written for Analysis Grand Challenge, but substantially modified and repurposed





Thank you!

- Thank you to the Wisconsin CMS group for feedback, encouragement, and lots of collaboration
- Thank you to Nick Smith, Lindsey Gray, and Oksana Shadura for support and mentorship throughout this project
- Thank you to the DOE and TAC-HEP traineeship program managers for making this project possible
- And thank you for your attention!


WISCONSIN
UNIVERSITY OF WISCONSIN-MADISON