NERSC Superfacility API

NERSC

Credits to:

LBNL Superfacility Team:
D. Bard, C. Snavely, **G. Torok**, R. Thomas, A. Greiner, etc.

NCEM: P. Ercius, C. Harris (Kitware)

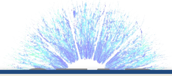
Bjoern Enders
Data Science Workflows Architect
NERSC/LBNL

NERSC Data Day, Oct 26, 2022

NERSC supports a large number of users and projects from DOE SC's experimental and observational facilities

Palomar Transient Factory Supernova

Planck Satelli
Cosmic Micro
Background
Radiation

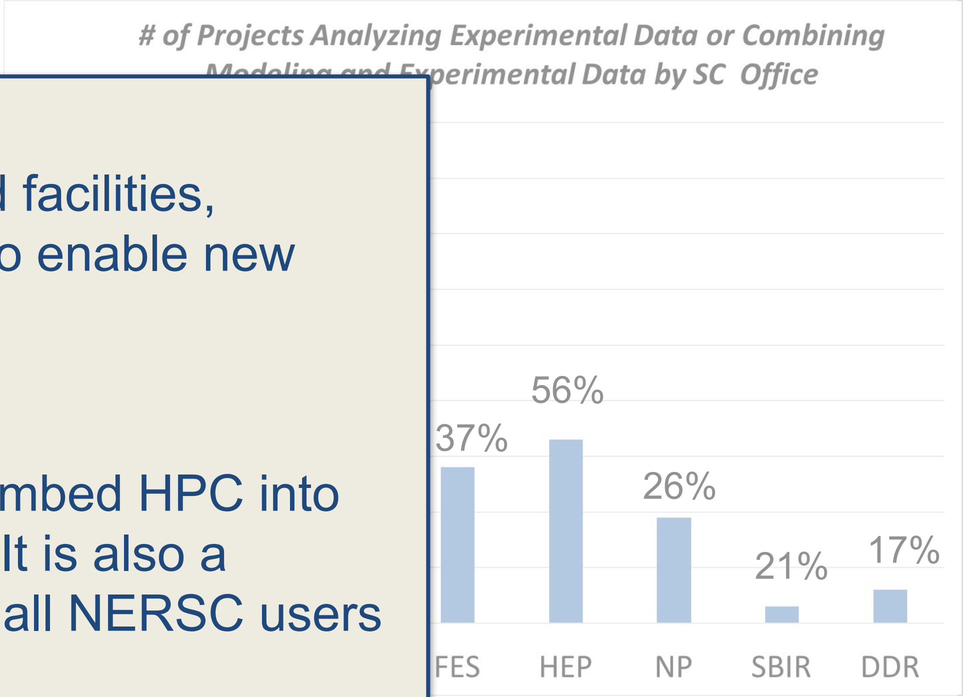


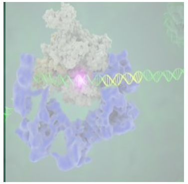
Dayabay Neutrinos

ALS Light Source

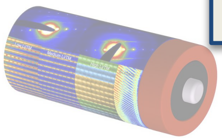
Superfacility:
Ecosystem of connected facilities, software and expertise to enable new modes of discovery

Superfacility API:
An API into NERSC to embed HPC into cross-facility workflows. It is also a general purpose API for all NERSC users and projects.





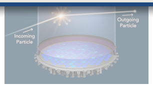
Cryo-EM



NCEM

DESI

LSST-DESC



LZ

~35% (235) of NERSC projects self identified as confirming the primary role of the project is to 1) analyze experimental data or; 2) create tools for experimental data analysis or; 3) combine experimental data with simulations and modeling

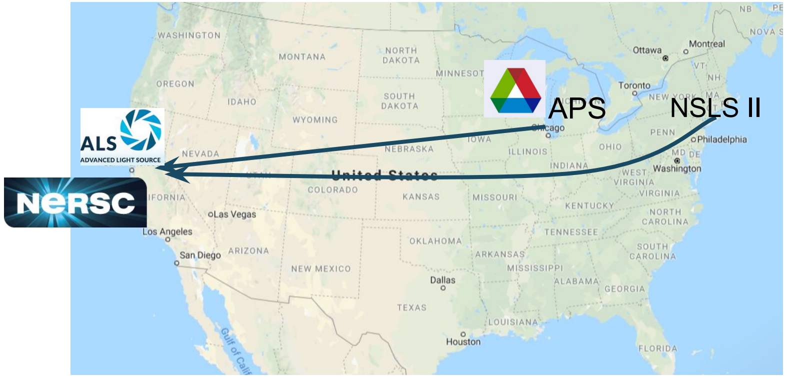
Model case

Experiments at ext. facilities use high frame rate 2D detectors for their science.

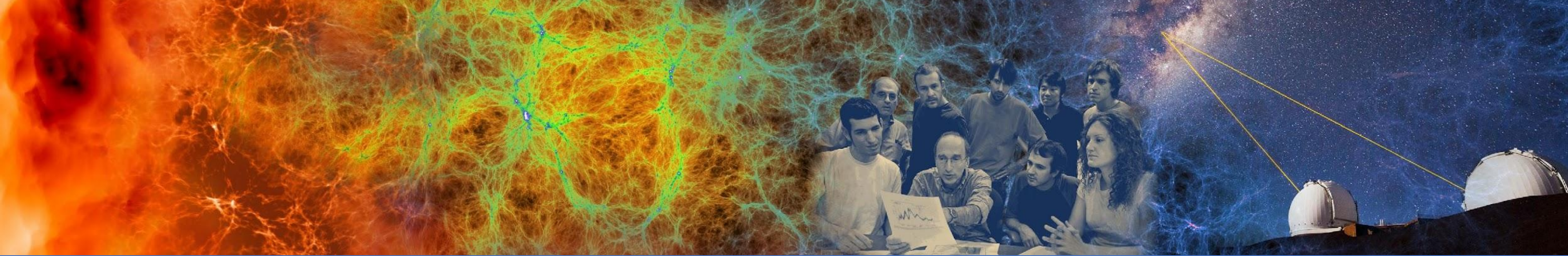
Hosting data & compute on site has become increasingly demanding.

Requirements

- Planning (HPC as reliable partner)
  - machine-readable status
- Resiliency (needs failover)
  - compatible interfaces
- Realtime (can't wait in queue)
  - workflow endpoint
- Services (portals, data, db)



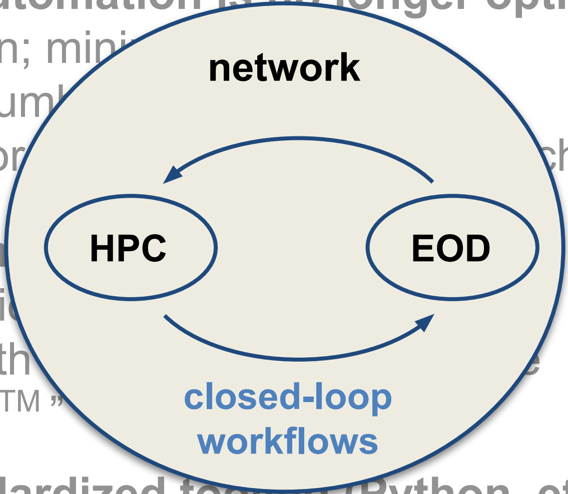
1. Plan / Check availability of NERSC resource for experiment.
   - check status / accounts
2. Get raw data to NERSC, when experiment is live.
   - move data
3. Start analysis job quasi synchronous with data
   - submit job / monitor job
4. Gather feedback, ideally immediate.
   - download / execute command
5. Move data and results to archive after analysis.
   - move data



The API

Why an API?

- **Meets a critical need; automation is no longer optional**
  - Unattended operation; mini
  - Track/submit large num
  - Interface with collabor chines
- **NERSC becomes "mach**
  - Enables easier creati
  - Allows integration with e
    - " NERSC inside™ "



- **Less DIY: simpler, standardized tooling (Python, etc)**
  - Stable refactor target for established projects or easier on-ramp for new ones
  - Contribute to HPC interface standards for portability
  - Authentication and security models

Drivers:

- **Complex workflows**
- **Data-driven projects**
- **Real-time compute and streaming data from instruments**
- **Automation**

What *specifically* can the API do?

Vision: all NERSC interactions are callable;
backend tools assist large or complex operations.

Endpoints prototyped or in prep:

`/status`	query the status of NERSC component system health
`/account`	data about the user's projects, roles, groups and usage information.
`/compute`	run batch jobs, query job and queue statuses on compute resources.
`/storage`	move data with Globus or between NERSC storage tiers
`/tasks`	get info about asynchronous tasks (eg. from `/compute` or `/storage`).
`/utilities`	traverse the filesystem, upload and download small files, and execute commands on NERSC systems
`/reservations`	submit and manage future compute reservations (coming soon)

Action	Manual steps	With SuperFacility API
Check status	Test SSH or ping specific services for status	Query the `/status` API endpoint if resources are active.
Submit job	SSH in and submit jobs with `sbatch ...`	Create jobs using POST calls from a script or Spin service to the `/compute` endpoint.
Monitor job	SSH in (again) and do `squeue | grep | sort | ...`	Consult the `/compute` and `/tasks` endpoints.
Plan ahead	Read the NERSC MOTD to see if any down time is planned	Query the `/status/outages/planned` API endpoint for planned outages
Move data	SSH in and run file transfer tools to move data	POST to the `/storage` API endpoint.
Check account	Log into "Iris" (our accounting web app) and check allocation account balance.	Query the `/account` API to get the same information.

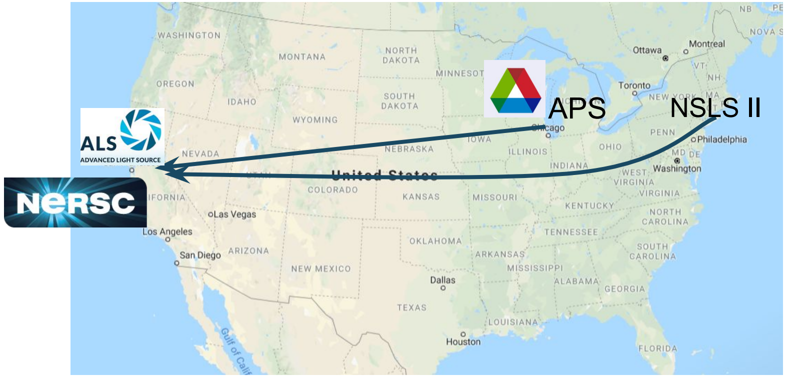
Model use

Experiments at ext. facilities use high frame rate 2D detectors for their science.

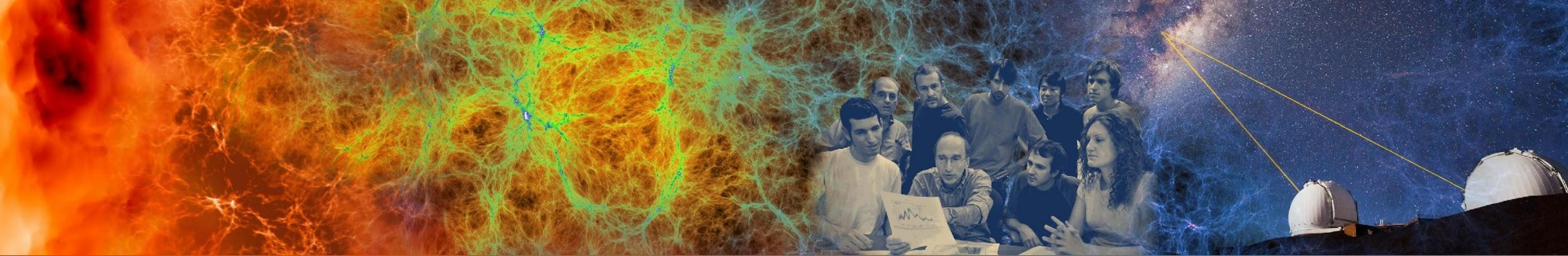
Hosting data & compute on site has become increasingly demanding.

Requirements

- Planning (HPC as reliable partner)
  - machine-readable status
- Resiliency (needs failover)
  - compatible interfaces
- Realtime (can't wait in queue)
  - workflow endpoint
- Services (portals, data, db)



1. Plan / Check availability of NERSC resource for experiment.
   - `/status (/reservations)`
2. Get data to NERSC, when experiment is live.
   - `/storage`
3. Start analysis job quasi synchronous with data
   - `/compute /tasks`
4. Gather feedback, ideally immediate.
   - `/utilities /storage`
5. Move data and results to archive after analysis.
   - `/storage`



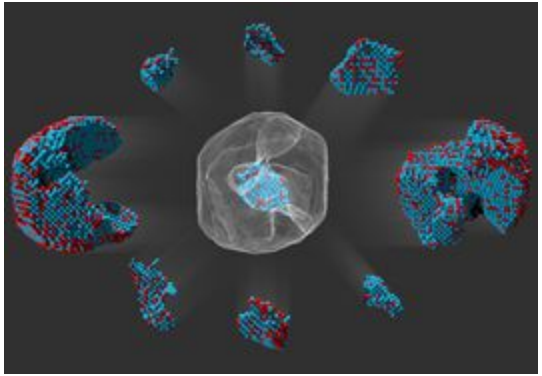
A science example

National Center for Electron Microscopy

- The MF provides state-of-the-art expertise, methods, and instrumentation in nanoscale science in a safe environment free of charge
- NCEM is one of 7 facilities in the MF (about ~1/3 of total proposals and staff)
- Staff Scientists work in a 50/50 model: 50% of their time is spent on user research and 50% of their time is spent on internal research. User research is often highly collaborative.

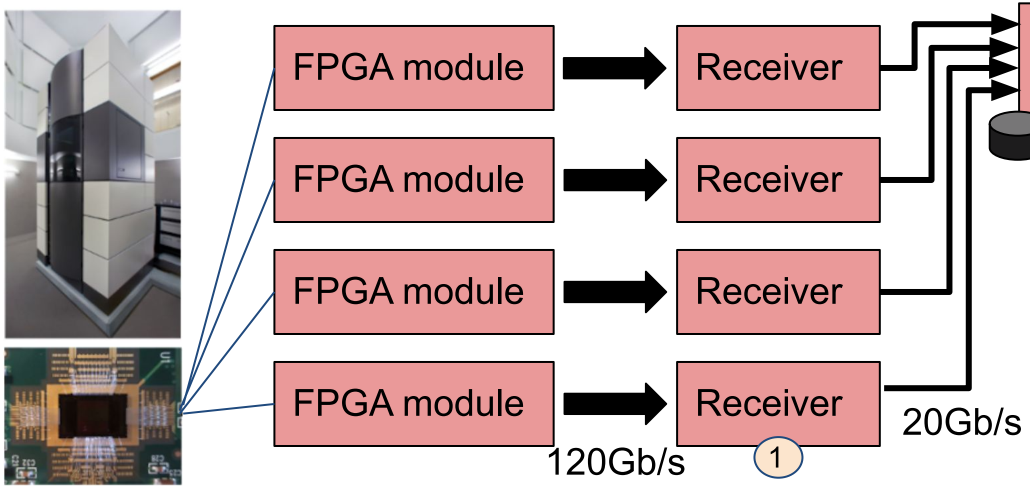
<u>We are leaders in:</u>

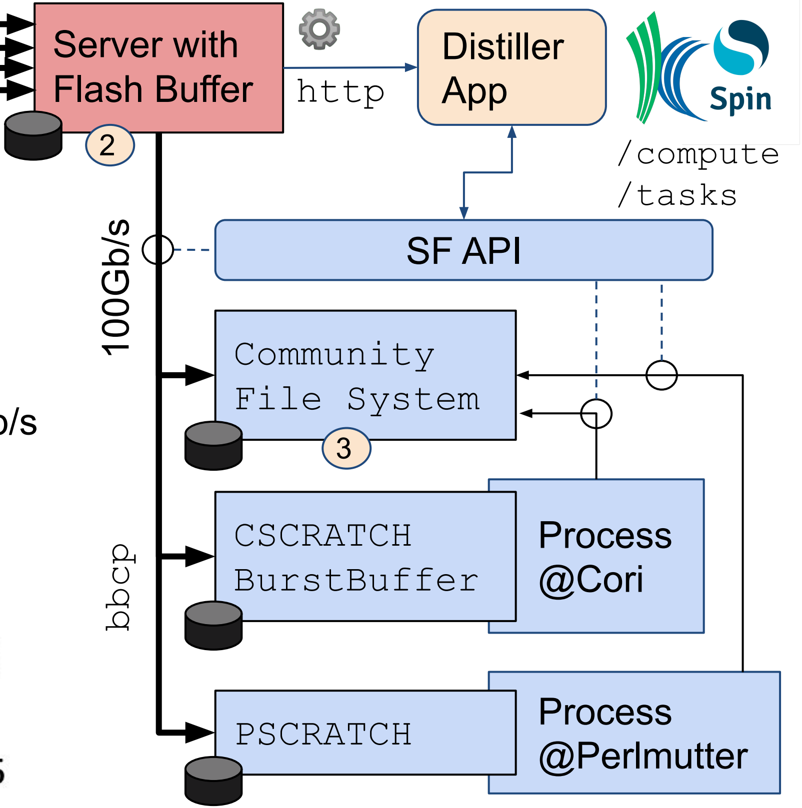
- high resolution
- tomography
- in situ
- soft materials
- 4DSTEM
- image simulation
- electron detector technology



Y. Yang, C Ophus, M. Scott, P. Ercius, J. Miao, et al., *Nature* (2017)

National Center for Electron Microscopy ...





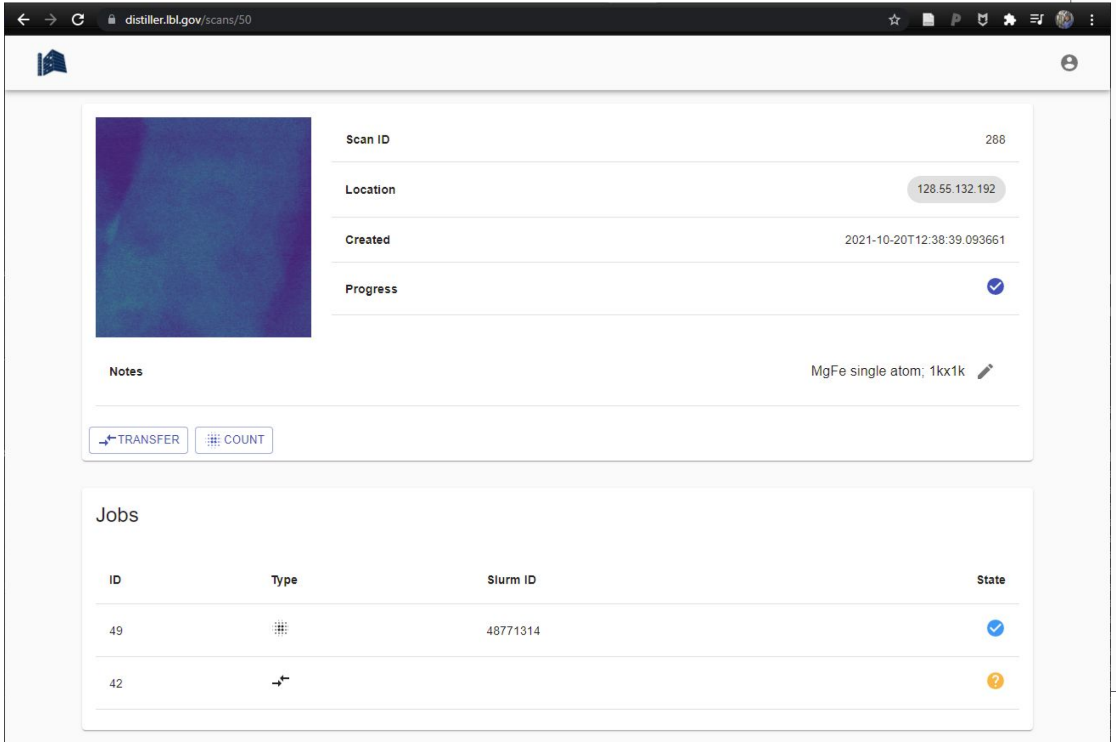
… uses NERSC to process large data sets live during experiments

- 87,000 Hz (480 Gbit/s) readout (typical STEM scanning rates)
- 1kx1k scan is 650 GB captured in 15 seconds
- Data pipeline: FPGA → RAM → Flash storage → Sparse HDF5

15 sec 1 140 sec 2 5 min 3

NCEM - Distiller app

credits: Chris Harris @ Kitware, Peter Ercius @ NCEM



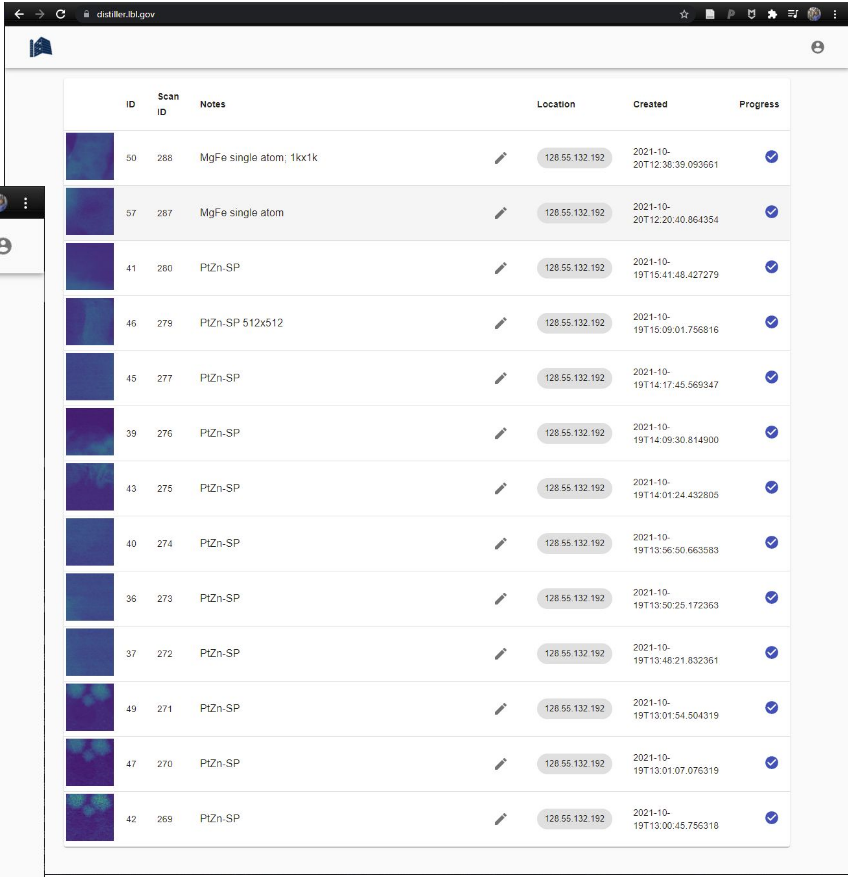
distiller.lbl.gov/scans/50

Scan ID	288
Location	128.55.132.192
Created	2021-10-20T12:38:39.093661
Progress	
Notes	MgFe single atom; 1kx1k

TRANSFER COUNT

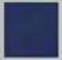
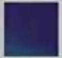
Jobs

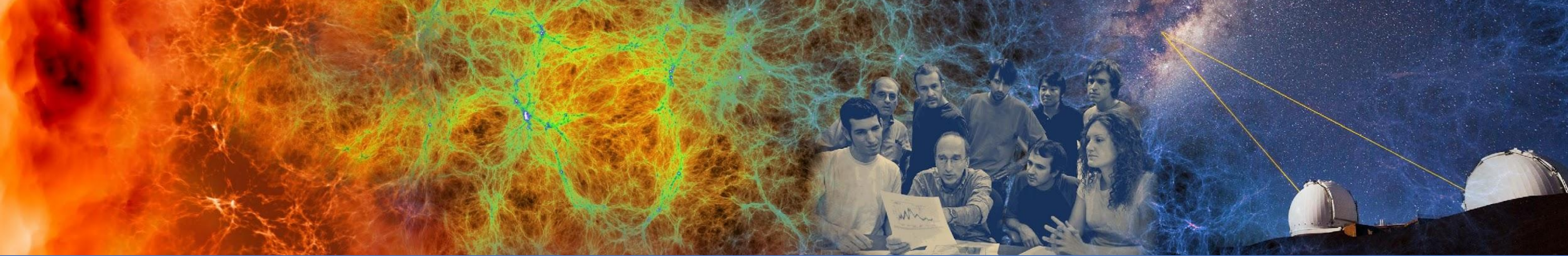
ID	Type	Slurm ID	State
49		48771314	
42			



distiller.lbl.gov

ID	Scan ID	Notes	Location	Created	Progress
50	288	MgFe single atom; 1kx1k	128.55.132.192	2021-10-20T12:38:39.093661	
57	287	MgFe single atom	128.55.132.192	2021-10-20T12:20:40.864354	
41	280	PtZn-SP	128.55.132.192	2021-10-19T15:41:48.427279	
46	279	PtZn-SP 512x512	128.55.132.192	2021-10-19T15:09:01.756816	
45	277	PtZn-SP	128.55.132.192	2021-10-19T14:17:45.569347	
39	276	PtZn-SP	128.55.132.192	2021-10-19T14:09:30.814900	
43	275	PtZn-SP	128.55.132.192	2021-10-19T14:01:24.432805	
40	274	PtZn-SP	128.55.132.192	2021-10-19T13:56:50.663583	
36	273	PtZn-SP	128.55.132.192	2021-10-19T13:50:25.172363	
37	272	PtZn-SP	128.55.132.192	2021-10-19T13:48:21.832361	
49	271	PtZn-SP	128.55.132.192	2021-10-19T13:01:54.504319	
47	270	PtZn-SP	128.55.132.192	2021-10-19T13:01:07.076319	
42	269	PtZn-SP	128.55.132.192	2021-10-19T13:00:45.756318	

	ID	Scan ID	Notes	Location	Created	Progress
	81	14	STEMH some lattice	128.55.132.192	2021-11-04T17:13:04.090986	
	80	13	STEMH better alignment	128.55.132.192	2021-11-04T17:10:50.558984	
	79	12	STEMH 512x512 100p Au on edge	128.55.132.192	2021-11-04T16:54:53.814969	
	78	11	512x512 8mm CL	128.55.132.192	2021-11-04T16:06:10.726925	
	77	10	512x512 Au	128.55.132.192	2021-11-04T15:54:21.766914	
	76	9	512x512x4 Au	128.55.132.192	2021-11-03T16:19:13.325631	
	75	8	512x512x4 Au	128.55.132.192	2021-11-03T15:51:44.517606	
	74	7	128x128 Au	128.55.132.192	2021-11-03T15:41:57.041597	
	73	5		128.55.132.192	2021-11-01T13:05:29.114843	
	72	308		128.55.132.192	2021-10-22T14:03:03.121835	
	71	307		128.55.132.192	2021-10-22T13:23:56.489800	
	70	306		128.55.132.192	2021-10-22T13:19:35.265796	
	50	288	MgFe single atom; 1kx1k	cori 128.55.132.192	2021-10-20T12:38:39.093661	
	57	287	MgFe single atom	128.55.132.192	2021-10-20T12:20:40.864354	
	41	280	PtZn-SP	cori 128.55.132.192	2021-10-19T15:41:48.427279	



How to use the API

Superfacility API Basics

https://api.nersc.gov/api/v1.2

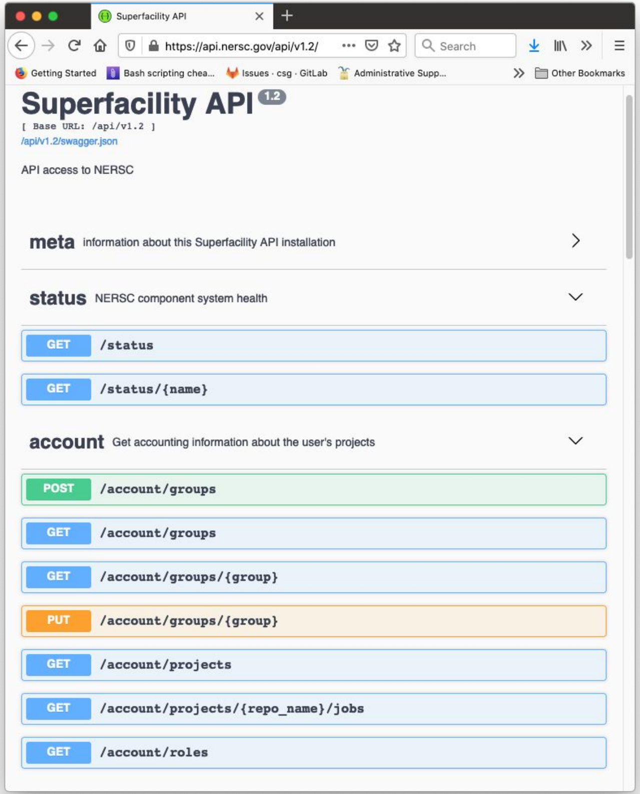
- A unified programmatic approach to accessing NERSC
- REST API with json input/output
- Standards-based authentication
- End user docs and examples: https://docs.nersc.gov/services/sfapi/

~ 4M logged requests since May,

= one request every 3 sec

Swagger documentation:

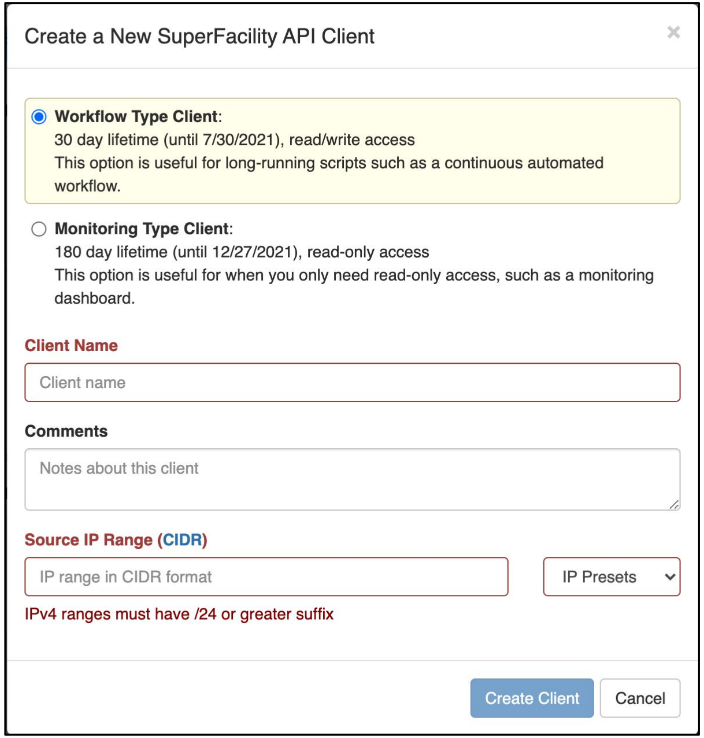
- Interactive, up-to-date and self-documenting
- See endpoints, payloads, example code
- Works with any dev environment



How to get your hands on the API

As a user:

- The /status endpoints are all public.
- Every NERSC user can get API access to non-public endpoints via Iris https//iris.nersc.gov
    - Profile -> scroll down to "Superfacility API Clients" tab
    - R/W clients require filling out a form
- Getting started documentation available at https://docs.nersc.gov/services/sfapi/


Create a New SuperFacility API Client
×
Workflow Type Client:
30 day lifetime (until 7/30/2021), read/write access
This option is useful for long-running scripts such as a continuous automated workflow.
Monitoring Type Client:
180 day lifetime (until 12/27/2021), read-only access
This option is useful for when you only need read-only access, such as a monitoring dashboard.
Client Name
Client name
Comments
Notes about this client
Source IP Range (CIDR)
IP range in CIDR format
IP Presets
IPv4 ranges must have /24 or greater suffix
Create Client
Cancel

As an HPC facility:

- Please get in touch with us if you have question on how to build an API at your facility.
    - benders@lbl.gov (Bjoern Enders)
    - djbard@lbl.gov (Debbie Bard)

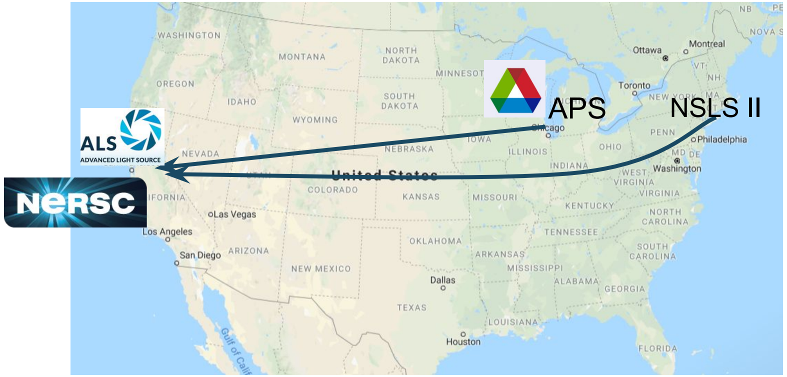
Example use

Experiments at ext. facilities use high frame rate 2D detectors for their science.

Hosting data & compute on site has become increasingly demanding.

Requirements

- Planning (HPC as reliable partner)
  - machine-readable status
- Resiliency (needs failover)
  - compatible interfaces
- Realtime (can't wait in queue)
  - workflow endpoint
- Services (portals, data, db)



1. Plan / Check availability of NERSC resource for experiment.
   - `/status (/reservations)`
2. Get data to NERSC, when experiment is live.
   - `/storage`
3. Start analysis job quasi synchronous with data
   - `/compute /tasks`
4. Gather feedback, ideally immediate.
   - `/utilities /storage`
5. Move data and results to archive after analysis.
   - `/storage`

File Edit View Run Kernel Git Tabs Settings Help

isc21_demo.ipynb

Code git PtyPy

FAVORITES

$HOME

$SCRATCH

FILE BROWSER

Filter files by name

/ ··· / benders / apidemo /

Name	Last Modified
isc21_dem...	a minute ago
job.sh	a minute ago
minimal2.py	6 minutes ago
slurm-437...	seconds ago
test.tif	2 minutes ago

{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'new', 'result': None}
{'id': '703', 'status': 'completed', 'result': '{"status": "ok", "jobid": "43778070", "error": null}'}

Check queue status, wait for job to complete.

[211]:
```
print(api("compute/jobs/cori/"+jobid+"?sacct=true"))
```


{'status': 'ok', 'output': {'workdir': '/global/homes/b/benders/apidemo', 'wckeyid': '0', 'wckey': '', 'usercpu': '00:00:00', 'user': 'benders', 'uid': '74824', 'tresusageouttot': '', 'tresusageoutmintask': '', 'tresusageoutminnode': '', 'tresusageoutmin': '', 'tresusageoutmaxtask': '', 'tresusageoutmaxnode': '', 'tresusageoutmax': '', 'tresusageoutave': '', 'tresusageintot': '', 'tresusageinmintask': '', 'tresusageinminnode': '', 'tresusageinmin': '', 'tresusageinmaxtask': '', 'tresusageinmaxnode': '', 'tresusageinmax': '', 'tresusageinave': '', 'totalcpu': '00:00:00', 'timelimitraw': '10', 'timelimit': '00:10:00', 'systemcomment': '', 'systemcpu': '00:00:00', 'suspended': '00:00:00', 'submit': '2021-07-01T23:47:41', 'state': 'RUNNING', 'start': '2021-07-01T23:47:46', 'resvcpuraw': '320', 'resvcpu': '00:05:20', 'reserved': '00:00:05', 'reservationid': '', 'reservation': '', 'reqtres': 'billing=64,cpu=64,node=2', 'reqnodes': '2', 'reqmem': '118Gn', 'reqgres': 'PER_NODE:craynetwork:1', 'reqcpus': '64', 'reqcpufreqgov': 'Unknown', 'reqcpufreqmax': 'Unknown', 'reqcpufreqmin': 'Unknown', 'reqcpufreq': 'Unknown'}, 'error': None}

Read from the slurm output file

[ ]:
```
slurmfile = home+"/apidemo/slurm-"+jobid+".out"
response = api("utilities/command/cori", { "executable": "tail "+slurmfile }, as_form=True)
if isinstance(response, Task):
    print(response.wait_for_result()['output'].strip())
```

Extract the file

We could just copy paste from above or us this one-liner to search the output for the saved file.

[ ]:
```
filename = [l for l in open(slurmfile, 'r').readlines() if l.startswith("Saving")][0].strip().split()[-1]
```

Simple 0 1 PtyPy | Idle Mem: 310.88 MB Saving completed Mode: Command Ln 1, Col 1 isc21_demo.ipynb sc21_doe_demo_perlmutter

Markdown

sfapi

```
Poll 04:{'id': '2236', 'status': 'completed', 'result': '{"status": "ok", "jobid": "484275", "error": null}'}
{"status": "ok", "jobid": "484275", "error": null}
```

Check queue status, wait for job to complete.

[338]:
```
#print(json.dumps(api("compute/jobs/perlmutter/"+jobid+"?sacct=true"), indent=2))
print(api("compute/jobs/perlmutter/"+jobid+"?sacct=true")['output']['state'])
```

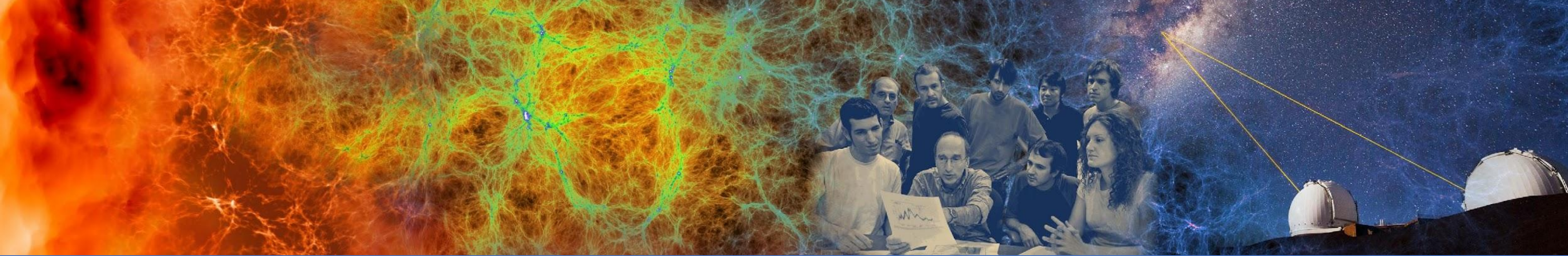
```
RUNNING
```

Read from the slurm output file

[334]:
```
slurmfile = home+"/apidemo/slurm-"+jobid+".out"
response = api("utilities/command/perlmutter", { "executable": "tail -n 20 "+slurmfile })
if isinstance(response, Task):
    print(response.wait_for_result()['output'].strip())
```


```
Poll 03:{'id': '2237', 'status': 'completed', 'result': '{"status": "ok", "output": "  * subpix               : linear\\n  * update_object_first  : True\\n  * overlap_converge_... : 0.05\\n  * overlap_max_itera... : 10\\n  * probe_inertia        : 1e-09\\n  * object_inertia       : 0.0001\\n  * fourier_power_bound  : None\\n  * fourier_relax_factor : 0.05\\n  * obj_smooth_std       : None\\n  * clip_object          : None\\n  * probe_center_tol     : None\\n  * compute_log_likel... : True\\n  * probe_update_cuda... : False\\n  * object_update_cud... : True\\n  * fft_lib              : reikna\\n  * alpha                : 1.0\\n  * name                 : DM_pycuda\\n================================================================================\\nIteration #10 of DM_pycuda :: Time 1.249\\nErrors :: Fourier 5.94e+01, Photons 4.73e+01, Exit 3.51e+01\\n", "error": null}'}
{"status": "ok", "output": "  * subpix               : linear\n  * update_object_first  : True\n  * overlap_converge_... : 0.05\n  * overlap_max_itera... : 10\n  * probe_inertia        : 1e-09\n  * object_inertia       : 0.0001\n  * fourier_power_bound  : None\n  * fourier_relax_factor : 0.05\n  * obj_smooth_std       : None\n  * clip_object          : None\n  * probe_center_tol     : None\n  * compute_log_likel... : True\n  * probe_update_cuda... : False\n  * object_update_cud... : True\n  * fft_lib              : reikna\n  * alpha                : 1.0\n  * name                 : DM_pycuda\n================================================================================\nIteration #10 of DM_pycuda :: Time 1.249\nErrors :: Fourier 5.94e+01, Photons 4.73e+01, Exit 3.51e+01\n", "error": null}
```

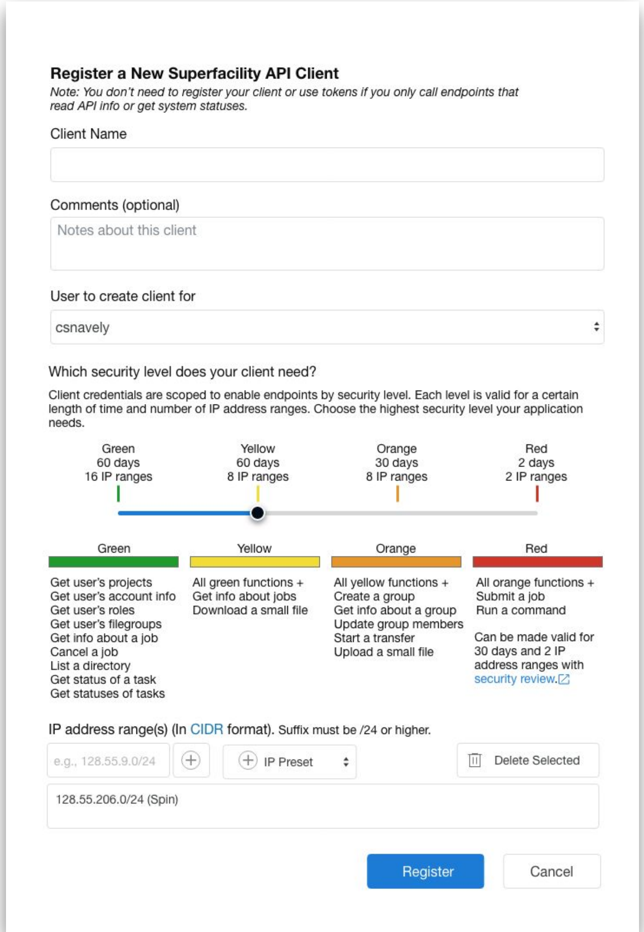

```
* subpix               : linear
  * update_object_first  : True
  * overlap_converge_... : 0.05
  * overlap_max_itera... : 10
  * probe_inertia        : 1e-09
  * object_inertia       : 0.0001
  * fourier_power_bound  : None
  * fourier_relax_factor : 0.05
  * obj_smooth_std       : None
  * clip_object          : None
  * probe_center_tol     : None
  * compute_log_likel... : True
```

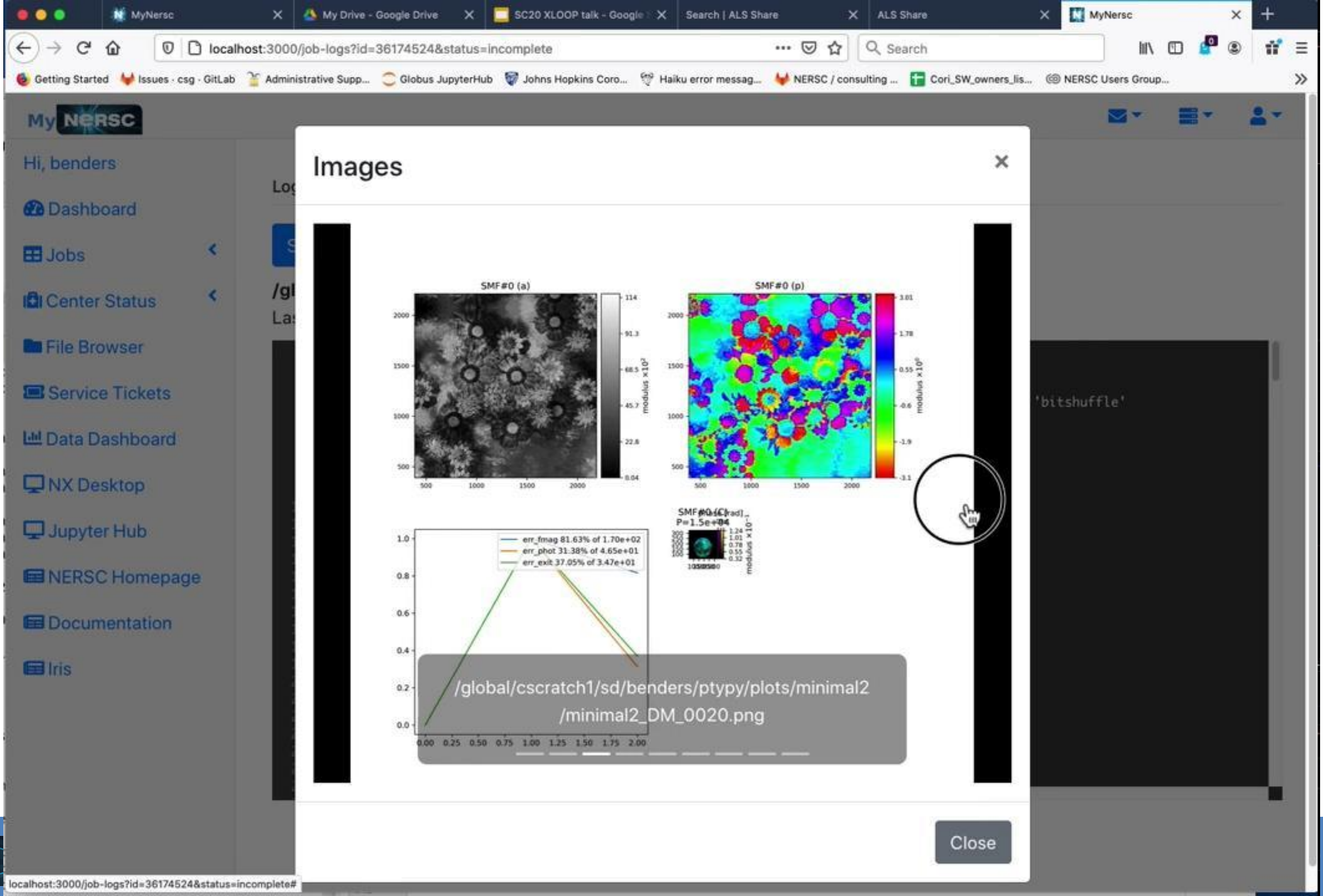


Roadmap

Roadmap

- Clients and Tokens with more granular scope (~ weeks)
  - new client interface (draft see right)
  - more source IP ranges per client
  - short-lived full featured clients without manual review
- SF API to retire NEWT (~ months)
  - login-based route to get tokens for mynersc, science gateway apps or other web apps.
- Common API interface (~ year)
  - a set of endpoints/methods that work with many facilities
  - we're talking with CSCS (firecrest API), HPCS@LBL, S3DF@SLAC, OLCF





Soren Bear
Gabor Torok
Bjoern Enders

Outreach

- High-level overviews of the API have been given at workshops and meetings oriented toward software development, such as the DOE Workflow Workshop and Hack-a-thon and NERSC GPUs for Science Day, both in 2019.
- A proof-of-concept demonstrations with Jupyter notebooks were shown at the DOE exhibition booth at SC'19 and SC'21 (the latter already with Perlmutter) (https://scdoe.info/demonstrations/)
- A detailed presentation of the API architecture and usage coupled with a Jupyter-based demonstration was given at the Superfacility Project Demo Series in 2020. https://www.youtube.com/watch?v=dmbBJmMUErU&list=PL20S5EeApOSsv6RVG6m0I6tx2wMp2T4PP&index=3
- A paper was published with the ISC’21 SuperCompCloud workshop and accompanied by a presentation “Automation for Data-Driven Research with the NERSC Superfacility API”
- Science examples of earlier adopters were presented at a SC’21 BoF about HPC APIs.
    - Building an HPC API community.
- We're in touch with OLCF to adapt a similar API for their facility