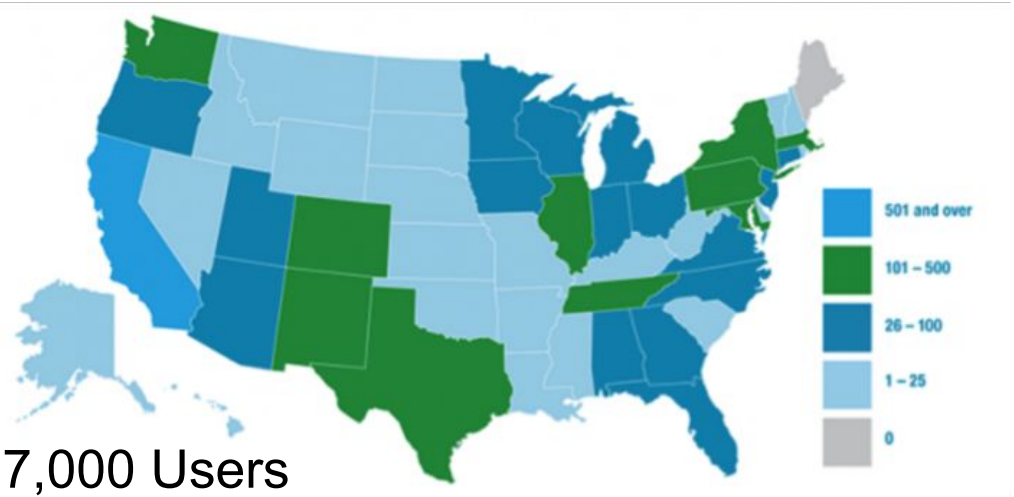
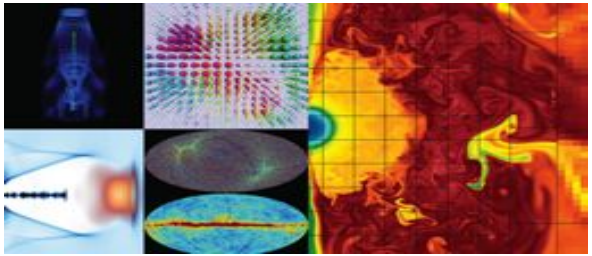
Special Interest Group (SIG) in NUG for Experimental Facility Users

NUGEX Chair: David Lawrence from Jefferson Lab

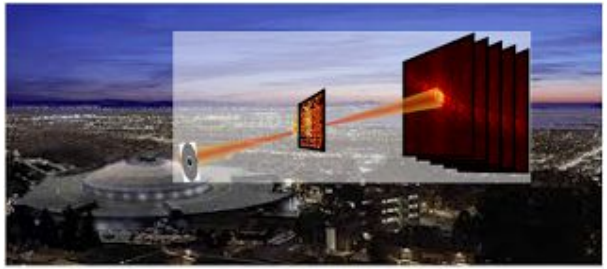
NERSC is the Production HPC & Data Facility for DOE Office of Science Research



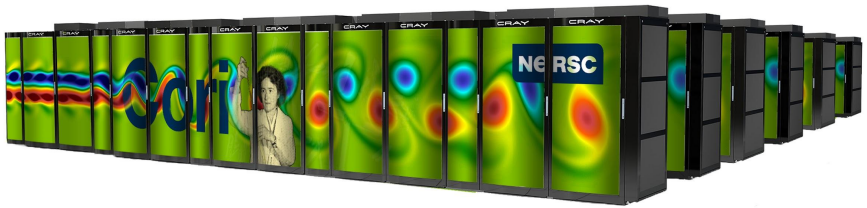
7,000 Users
800 Projects
700 Codes
2000 NERSC citations per year



Simulations at scale



Data analysis support for DOE's experimental and observational facilities
Photo Credit: CAMERA



U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

We've heard from the community through requirements gathering

- ***HPC is required*** to analyze data from experimental facilities - but changes are needed to ***both*** application workflows and HPC environments
- Scientists require support for analysis software and tools, many of which ***differ significantly*** from traditional simulation software.
- New approaches are needed for ***analyzing large datasets*** including ***advanced statistics*** and ***machine learning.***
- As science increasingly becomes a ***community*** effort, the need to ***share, transfer, search and access*** data becomes even more important.
- New strategies for ***resilient workflows*** are required
- Experimental facilities will require ***new modes of interacting*** with the systems including notebooks and faster queue turn-around.
- Changes in ***policies*** are ***as important*** to address as technical challenges

Superfacility Science Engagements

ALS

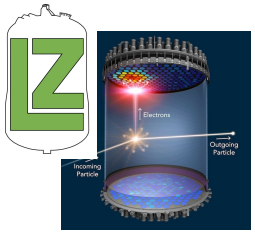
High-rate detectors use NERSC for real-time experimental feedback, data processing/management, and comparison to simulation

LSST DESC Dark Energy Science Collaboration

Processing streaming alerts (from NCSA) for detection of supernova and transient gravitational lensing events

LCLS

High-rate detectors use ESnet and NERSC for real-time experimental feedback and data processing



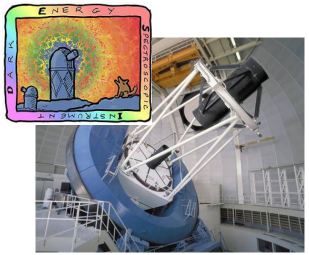
Next-generation dark matter detection, continuously sending data to NERSC and UK

FICUS Facilities Integrating Collaborations for User Science
JGI-EMSL Collaborative Science Call

Complex multi-stage workflow to analyse response of soil microbes to climate change

MOLECULAR FOUNDRY

4D STEM data streamed to NERSC, used to design ML algorithm for future deployment on FPGAs close to detector

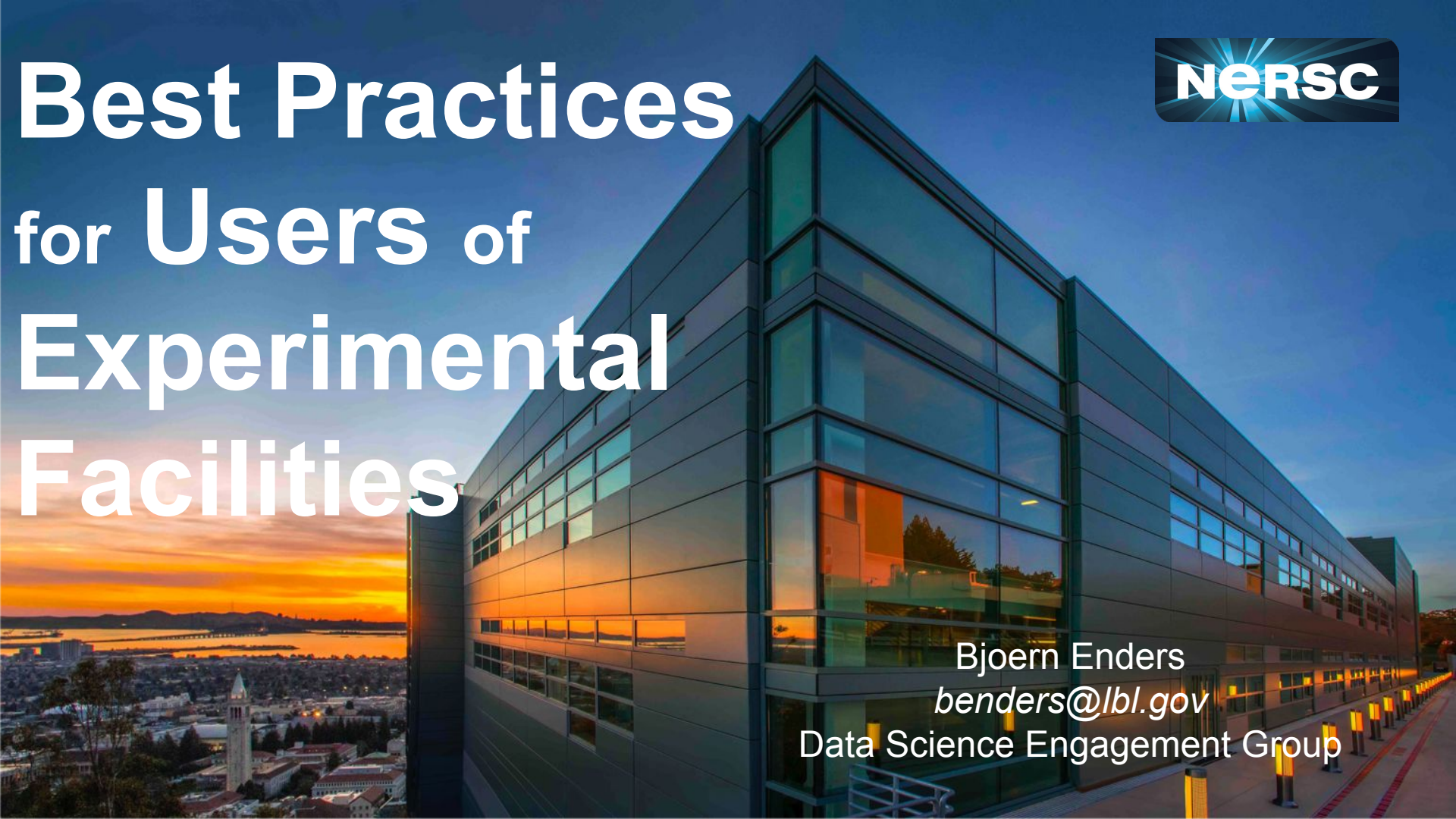


Nightly processing of galaxy spectra to inform next night's telescope targets

Experimental Facility SIG

- Build community, offer a forum to meet other users
- Learn about best practices from NERSC and and other users
- Learn about NERSC's plans to support this emerging community and provide feedback to NERSC
- Influence the direction of NERSC policies that affect this community

Best Practices for Users of Experimental Facilities



Bjoern Enders
benders@lbl.gov
Data Science Engagement Group

What makes experimental workloads different than traditional sim & modelling?

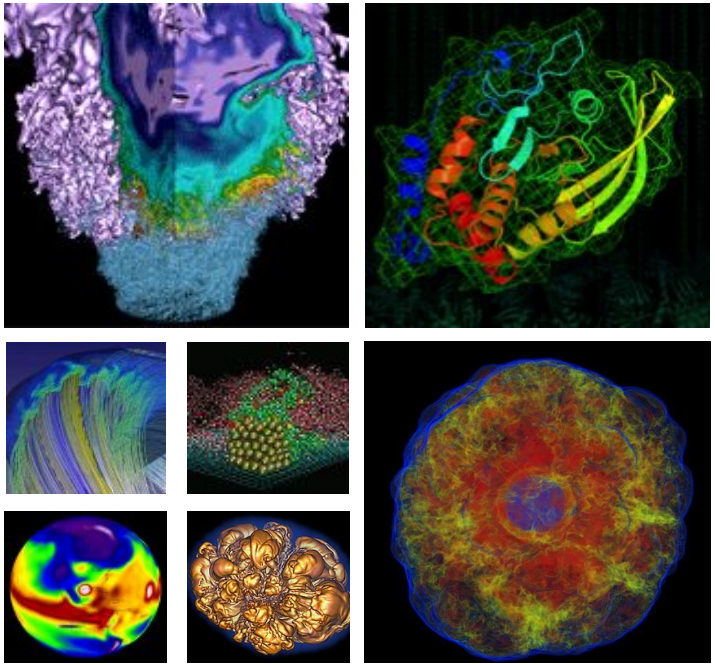
- **Data often not generated on-site but needs to be transferred in: sometimes to archive, sometimes to CFS, sometime direct to compute node for in-memory analysis.**
  - *How to get my data into NERSC?*
  - *Where should the data go?*
  - *How do I share my data?*
- **Productivity as important as performance.**
  - *How do I run Python?*
- **Needs real-time (or near) turnaround and interactive access for running experiments: Availability and throughput are crucial.**
  - *Which queues to use for quick access?*
  - *What if my job is single thread or serial?*
- **Needs an ecosystem of persistent edge services, including workflow managers, visualization, databases, web services…**
  - *How can I use containers at NERSC?*
- **Experimental facilities are user facilities themselves. (FedID, secondary users)**
  - *Where do I install my software?*
  - *What are communication best practices?*

OVERVIEW

Transfer and manage data

Running jobs

Container & Software

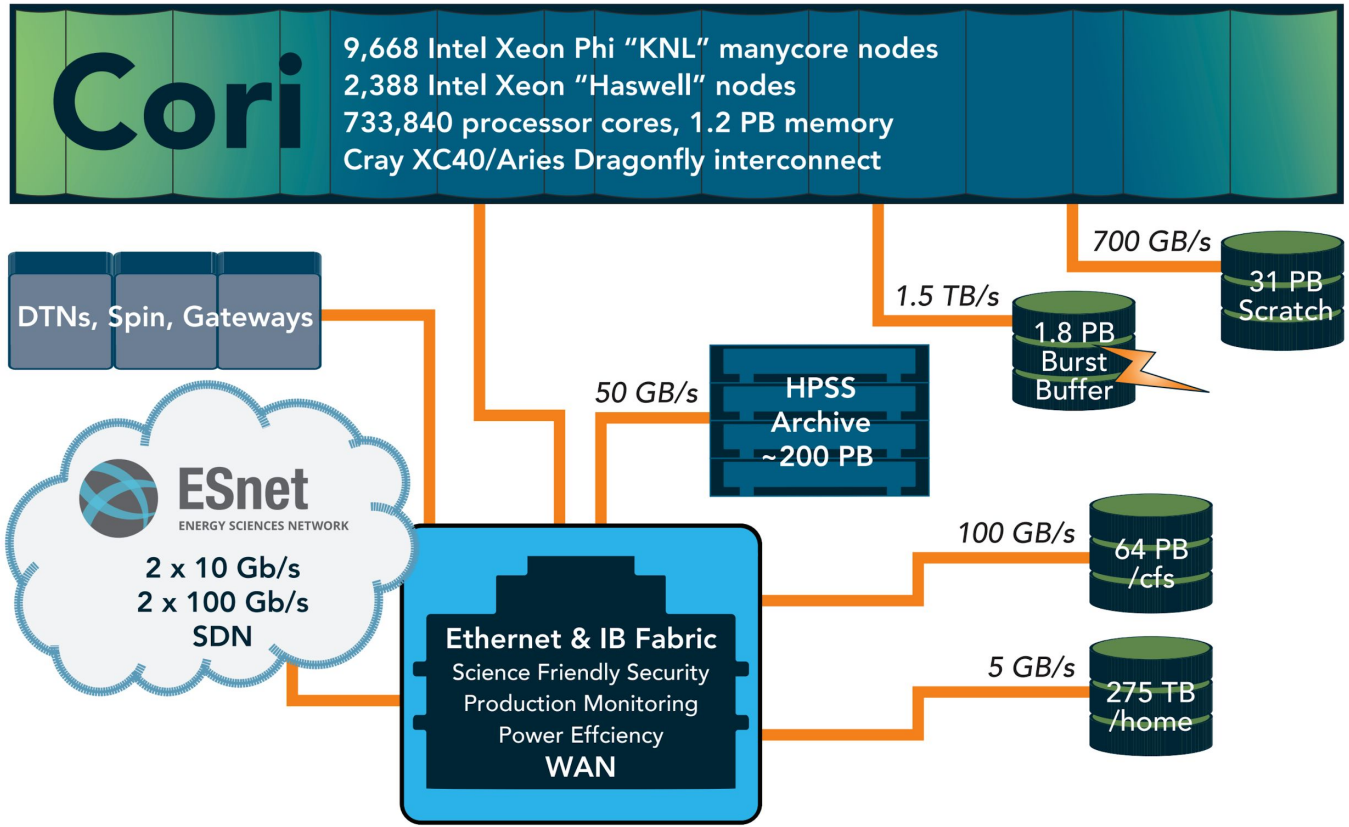


NERSC

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

NERSC systems today



How to get my data into NERSC?

Globus

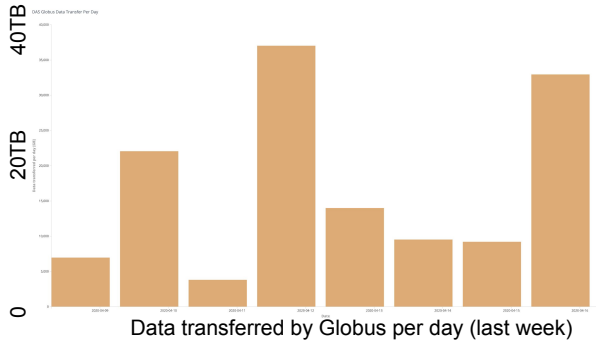
https://docs.nersc.gov/services/globus

- **The recommended tool for moving data in & out of NERSC**
  - Reliable & easy-to-use web-based service.
    - http://www.globus.org/ or http://globus.nersc.gov/
  - Accessible to all NERSC users.
  - NERSC managed endpoints for optimized data transfers.

- **Globus extensive documentation** https://docs.globus.org
  - Web based interaction with service.
  - REST/API for scripted interactions (in Bash, Python and other languages) with service.
  - Globus Connect Server & Personal for setting up additional remote endpoints such your personal laptop.

Did you know?

- **8,149,745.261GB were moved by Globus in/out of NERSC in 2019**
- **That is the same size as the /project filesystem!**



How to get my data into NERSC?

- **Data Transfer Nodes (DTN) are dedicated servers for moving data at NERSC. (dtnXX.nersc.gov)**
  - Servers include high-bandwidth network interfaces & are tuned for efficient data transfers
    - Monitored bandwidth capacity between NERSC & other major facilities such as ORNL, ANL, BNL, SLAC…
  - Direct access to global NERSC file systems & Cori cscratch1
  - Can be used (and *should* be used) to move data internally between NERSC systems &/or NERSC HPSS
  - *DO NOT USE* for non-transfer purposes

Where should the data go?

- Community File System
  - Quota depends on project allocation and is shared with other members of your project.
  - In globus use collection `NERSC DTN` and path `/global/cfs/cdirs/<project_name>`
  - Data never gets deleted and has 7 days of backup via snapshots.
  - Recommended as **first landing pad** for your data.
  - PI can partition storage allocation into custom folders via Iris.
- Cori Scratch
  - You have 20 TB for yourself.
  - In globus use collection `NERSC DTN` and path `/global/cscratch1/sd/<user_id>`
  - Data gets **purged** after 12 weeks. See dot files `.purged<date>` for list of purged files
  - Recommended if data is used for **imminent compute**.

Consult https://docs.nersc.gov/filesystems/ for more detailed info

Where should the data go?

- Tape Archive (HPSS)
  - Quota is determined by your project and can be adjusted by project PI.
  - In globus use collection `NERSC HPSS` and path
    - `/home/<u>/<user_id>` for your personal archive.
    - `/home/projects/<project_name>` for your project's archive.
  - Package your data in units of 100-500GB to avoid files being spread over many tapes.

  - If you're retrieving many files from HPSS, please use the Globus NERSC command line tools
  - Archive access comes with a serious latency and limited transfer speeds.
  - Recommended if data doesn't need to be touched **for months or years.**

Consult https://docs.nersc.gov/filesystems/ detailed info

How do I share my data?

- **For other project members**
  - In `/global/cfs/cdirs/<project>`, other project members have read permissions by default.
- **For other NERSC users**
  - You can use `give` and `take`.
  - Modify directory access permissions, e.g. `chmod o+r /path/to/sharing_dir`.
- **For external users, use Globus Sharing**
  - ***Tell us*** that you like to have Globus Sharing enabled for your project.
  - Place files in subdirectory of agreed upon sharing directory (gsharing).
  - In globus use endpoint `NERSC SHARE` and path `/global/cfs/cdirs/<project_name>/gsharing/<share_subdir>`
  - Use `NERSC SHARE` to create a globus share for the subdirectory.
  - Shares are read-only, but *any* Globus user can be added to a share.
  - Delete share if access is no longer needed, this will not delete the data.

Tips and Tricks

- **Use Globus Online for large, automated or monitored transfers**
  - Remember that every aspect of globus can be scripted using their CLI or (Python-)API.
- **scp is fine for smaller, one-time transfers (<100MB)**
  - But note that Globus is also fine for small transfers.
- **Plain "cp" can be used for transfers within file systems**
  - Can use Globus for convenience.
- **Staging data from HPSS for a compute job?**
  - Try not to use a login shell (you get kicked out)
  - Split your transfers in multiple jobs if you run out of time in your queues.
  - Use the transfer queue if you do lots of data movement
- **Don't use your `$HOME` directory**
  - Instead use `/global/cfs/cdirs/<project>`, `$SCRATCH` ...
  - unless, of course, for *very small* data (MBs)

What if transfers fail or are too slow?

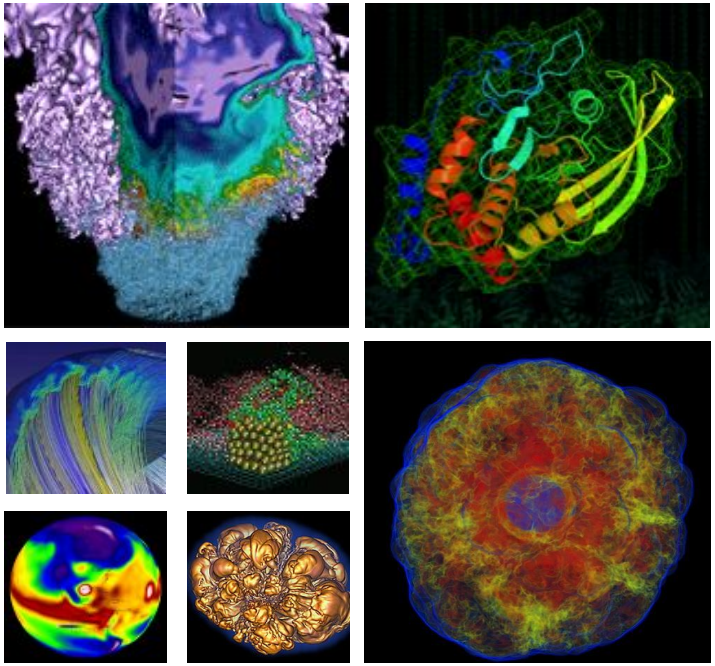
- **Performance is often limited by the remote endpoint**
  - Not tuned for WAN transfers or have limited network link
  - These can lower performance < 100 MB/sec.
- **File system contention may be an issue**
  - Try the transfer at a different time or on a different file system.
- **Use ESnet DTNs to test the link to NERSC and to your facility.**
  - This can be done in Globus as well.
  - The DTN's contains are read-only and contain datasets of varying sizes.
  - Initiate transfers from these sites to NERSC and to your endpoint. Globus logs the average transfers speed. All transfers are listed in the "Activity" tab of globus online.
- **Consult the ESnet perfsonar dashboard**

OVERVIEW

Transfer and manage data

Running jobs

Container & Software



NERSC

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Which queues to use for quick access?

Consult https://docs.nersc.gov/jobs/policy/ for detailed info about available queues

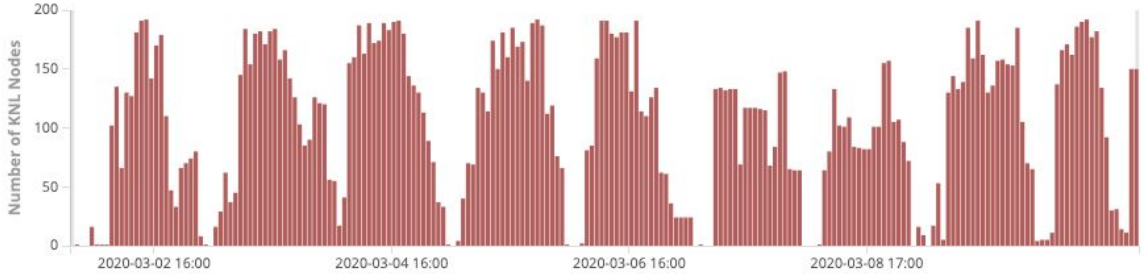
- **Debug (batch mode or interactive)**
  - Limits: 512 nodes, walltime 30 mins , run 2, submit 5
  - Depending on job size, it can take a while for your interactive session to be granted.
  - `% salloc -N 20 -q debug -C haswell -t 30:00`
- **Interactive**
  - **THE** recommended queue for interactive processing.
  - Instant allocation (in 5 min or reject), run limit 2, submit limit 2.
  - Is less busy during off-hours (PT).
  - Max nodes is 64 *per repository* and max walltime is 4 hours
  - Batch submission is *disabled.*
  - `% salloc -N 2 -q interactive -C knl,quad,cache -t 2:00:00`

Which queues to use for quick access?

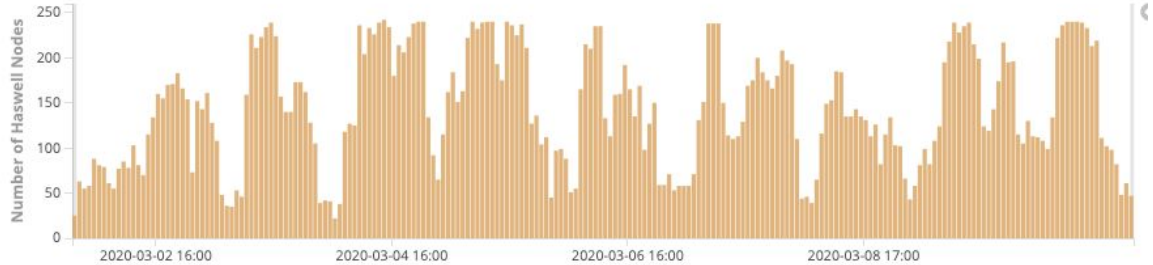
Consult https://docs.nersc.gov/jobs/policy/ for detailed info about available queues

- **Realtime (batch mode or interactive)**
  - Queue is not shared with users from other projects.
  - Only available via special request --
  - Intended for groups that rely on immediate computing turnaround to operate experiment now or in future.
    - Not simply for impatient users.
  - Use for **rapid access** to resources on Cori.
  - Some projects have a realtime queue
    - Check on Iris if you are have the qos enabled.
  - Only Cori Haswell - No realtime queue possible for KNL.
  - Realtime queues are limited to a couple of nodes (usually 10).
  - `% salloc/sbatch -q realtime -A <repo>`
- *If you have other realtime needs please let us know, we want to work with you*

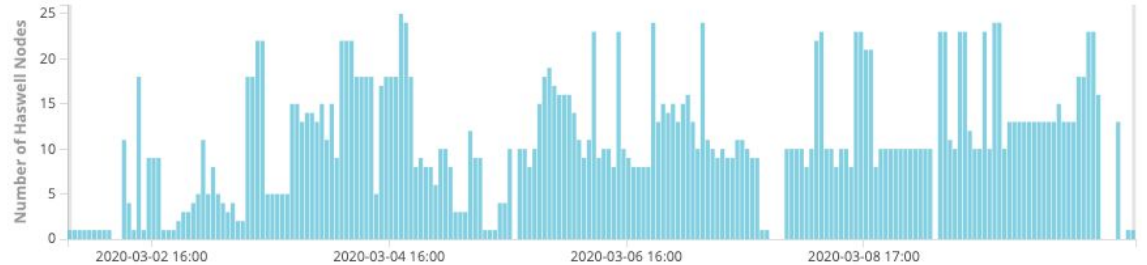
Interactive Queue Utilization



Cori KNL



Cori Haswell



Realtime

Why can't I just use my login shell?

(for everything)

- Most resources at NERSC are **shared.** This includes (among others):
  - Bandwidth on the network.
  - Bandwidth accessing (I/O) for global filesystems, like scratch, project, hpss, etc…
  - Human support :-).
  - .. and the login nodes.
- When on a Cori login node (by `ssh cori.nersc.gov` for example)
  - Be mindful that this resource is shared with other users.
  - SSH connection are not all as reliable and might get interrupted.
    - Consider NoMachine (NX) for a longer session with graphics
    - Consider Jupyter for Python scripts.
  - Use the login node **primarily** to:
    - Edit files, compile codes, submit batch jobs, access nodes etc,
    - Run short, serial utilities and applications.

What if my job doesn't need a full node?

- **Does your job require only a few cores/threads?**
  - Use the shared queue (https://docs.nersc.gov/jobs/examples/#shared).
  - Can schedule as fine as a single core (+HT).
- **Do you need to run a large number of small jobs?**
  - Pack the jobs with a workflow tools (https://docs.nersc.gov/jobs/workflow-tools/)
    - e.g. Taskfarmer (non-MPI jobs)
    - NERSC is currently evaluating workflow tools
    - What is your preferred workflow manager? -> Discussion
  - Avoid using job arrays (only 2 jobs at a time in the array will be considered by Slurm for scheduling)
  - Avoid using `srun` repeated in large for-loops (Slurm will seize up trying to execute them all)

How can I talk to my running job?

Software Defined Networking

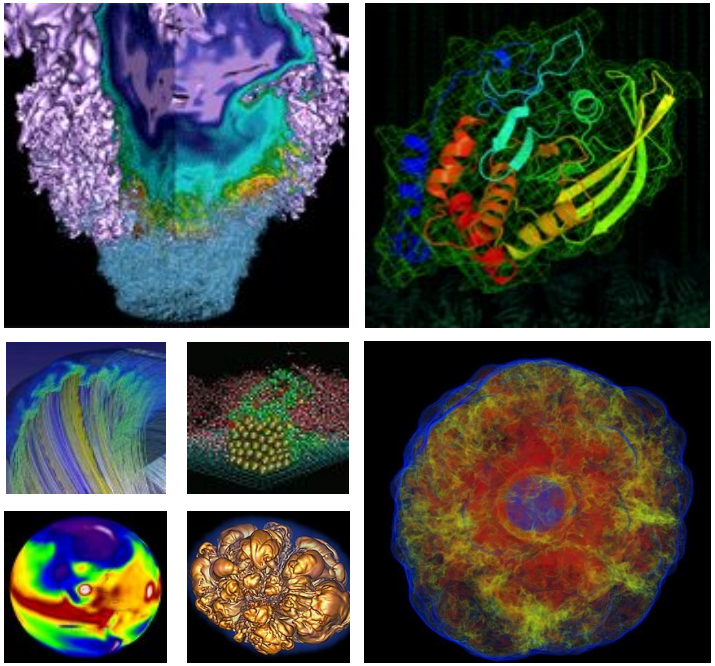
- Compute nodes in Cori do not have IP addresses and cannot be reached from the outside.
- NERSC deployed a software translation layer on Cori bridge nodes to direct IP traffic to the head node of a job.
- Usage example for an interactive session:
  - `user@cori10:~> salloc -C haswell -q interactive --sdn`
  - `salloc: Granted job allocation 29234281`
  - `user@@nid00025:~> echo $SDN_IP_ADDR`
  - `128.55.224.202`
- You can reach the head node now from the outside under the ip address `128.55.224.202` or `job29234281.cori.services.nersc.gov`
- **CAUTION**: This address is directly exposed to the internet, make sure to run secure services.

OVERVIEW

Transfer and manage data

Submit jobs

Container & Software



NERSC

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Where do I install my software?

3 places recommended to install your software at NERSC, depends on scaling and complexity

- **$HOME/<your software>**
  - The default, use for testing and small scale applications.
  - Can lead to IO congestion for very large jobs.
- **/global/common/software/<your project>/<your software>**
  - The *best* place for entire project/community and *created by default*.
  - Requires you to be on a project and "take ownership" of the deployed software.
  - Good for larger scale applications, software dir is **cached** on the compute nodes.
- **Shifter**
  - Best for largest scale application, software is shipped to compute nodes on launch.
  - Investment in effectively packaging up your code in a Shifter container.
- **Preferably, *do not use:***
  - /global/cfs/cdirs -> It is meant for data.
  - $SCRATCH -> Same as above, plus **purging**

How do I run Python?

Consult https://docs.nersc.gov/programming/high-level-environments/python/ for detailed information as well as the new user training video and slides

- Python is a popular language with domain scientist due to ease of use and large module ecosystem.
- NERSC delivers Python through Anaconda and does not use the system Python.
  - `user@cori10:~> module load python; which python`
  - `/usr/common/software/python/3.7-anaconda-2019.10/bin/python`
- Gives you the full “Anaconda” experience.
- Has an acceptable launch time performance, as it is cached on the compute nodes.
- mpi4py module in the *base* environment is mapped to the appropriate Cray hardware and libraries of Cori.

That's great, but I want *my* Python!

The **recommended** way for this scenario is:

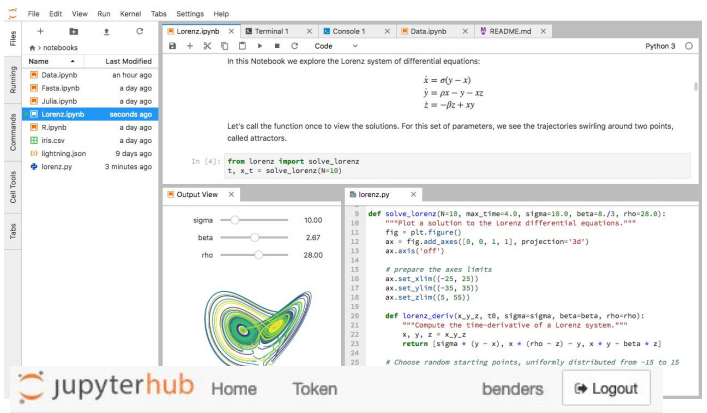
- Create a new empty env either from our conda, or better from a fresh miniconda install.
- For mpi4py, load the necessary module (i.e. cray-mpich) to be able to talk to Cray MPI hardware.
- `pip install`, `setup.py install` or `conda install`
- **Caution**: Rigorous dependencies in conda channels might destroy your previous work.
- Dispose of the env is something goes wrong and start again.
  - Better: make a script to build your env
- Deploy env on `/global/common/software/<your project>` for all your project members. This also makes loading faster for large jobs.
- Package your conda env in docker and put it in Shifter for best performance.
- ***Consult with your liaison at NERSC or with a ticket for guidance along this process.***
- On upgrading the system this process might need to get repeated as with all other software.

Jupyter

Consult https://docs.nersc.gov/connect/jupyter/ for detailed info.

- Perform exploratory data analytics and visualization of data stored at NERSC.
- Enjoy a machine learning ecosystem.
- Manage workflows through the Cori batch queue.

- Go to https://jupyter.nersc.gov to use Jupyter at NERSC.
- You can use *your* conda environment in your notebook. Make kernel spec with
  - `$ source activate myenv`
  - `$ python -m ipykernel install --user --name myenv --display-name MyEnv`
- You can customize kernelspec files in many ways (docs)
- ***We work on making Jupyter work for you***



https://jupyter.nersc.gov/

	Shared CPU Node	Shared GPU Node
Gerty	start	
Cori	start	start
Spin	start	
Resources	Use a node shared with other users' notebooks but outside the batch queues.	
Use Cases	Visualization and analytics that are not memory intensive and can run on just a few cores.	

How do we coordinate a large computing campaign?

Consult https://docs.nersc.gov/accounts/collaboration_accounts/ for more information

- Collaboration Accounts (CA) are designed to facilitate collaborative computing by allowing multiple users to use the same pseudo-account, eg to coordinate a simulation campaign.
  - ***Users still need their own NERSC accounts! You cannot share your regular account.***
- Use the CA for shared access to batch jobs or data.
- A CA can be requested by PI or PI Proxy of a project with a ticket, future feature in Iris.
- PI or PI Proxy can add project member to a CA on Iris.
- Internally (after login), you can assume the identity of the CA with `collabsu`:
  - `user@cori10:~> collabsu <collaboration account name>`
  - `<enter nersc password at the prompt>`
  - `<collabuser>@cori10:~>`
- For external use, you can use `sshproxy` to generate an ssh key for the account:
  - `user@cori10:~> sshproxy.sh -c <collaboration account name>`

Note: Users come and go. **The individual actions of the CA can be traced back to the user via logs**, but since the data has no specific person associated with it, it is much harder to decipher later on whether or not the data needs to be kept or can be deleted.

How can I use Containers? -- Shifter & Spin

Why we cannot just run Docker:

- **Security**: Docker currently uses an all or nothing security model. Users would effectively have system privileges
- **System Architecture:** Docker assumes local disk
- **Integration:** Docker doesn't play nice with batch systems.
- **System Requirements**: Docker typically requires a very modern kernel
- **Complexity**: Running real Docker would add new layers of complexity

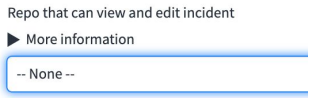
Still, we like containerized workloads so NERSC provides two solutions based on Docker:

- Spin runs on dedicated hardware with stock Docker and Rancher for services that need to run "indefinitely" or need to be externally visible (https://docs.nersc.gov/services/spin/) .
- Shifter runs processes as the user on our HPC systems for simulation and analysis that need to run at scale (https://docs.nersc.gov/programming/shifter/how-to-use/).
- NERSC supports a private docker registry (registry.services.nersc.gov) for container images with sensitive or proprietary components

What are communication best practices?

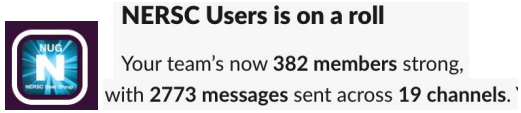
How can I see other tickets submitted by members of my team?

- Make sure you enable your repo to see your ticket when you submit it, using this drop-down list:
- We are working on enhancements to ServiceNow to make this automatic when submitting all tickets.



How can I chat with other users informally about NERSC topics?

- We've launched the NUG NERSC Users slack channel (login required)! https://www.nersc.gov/users/NUG/nersc-users-slack/ .
- Please note that this is not an official NERSC staff-supported platform. While NERSC staff may sometimes join the NERSC Users Slack, the best way to reach NERSC is still through the online help desk at https://help.nersc.gov.



How should we coordinate NERSC support for a large collaboration?

- It is best practice for experimental teams to have a 'NERSC liaison' or a NERSC point person to coordinate issues and advocate for your team's needs.

Discussion Time

- Possible topics for future Experimental Facility SIGs
  - Resilience strategies
  - Deep dives on one of the topics addressed today
  - Perlmutter architecture and readiness
  - Best workflow managers to use

Thank You