Dark Energy Spectroscopic Instrument (DESI)

Making a 3D Map of the Universe

Stephen Bailey

DESI Data Management Lead, LBL Physics

Snowmass Workshop on Software and Computing for Small HEP Experiments

2021-11-15

Outline

— What we do

— How we do it

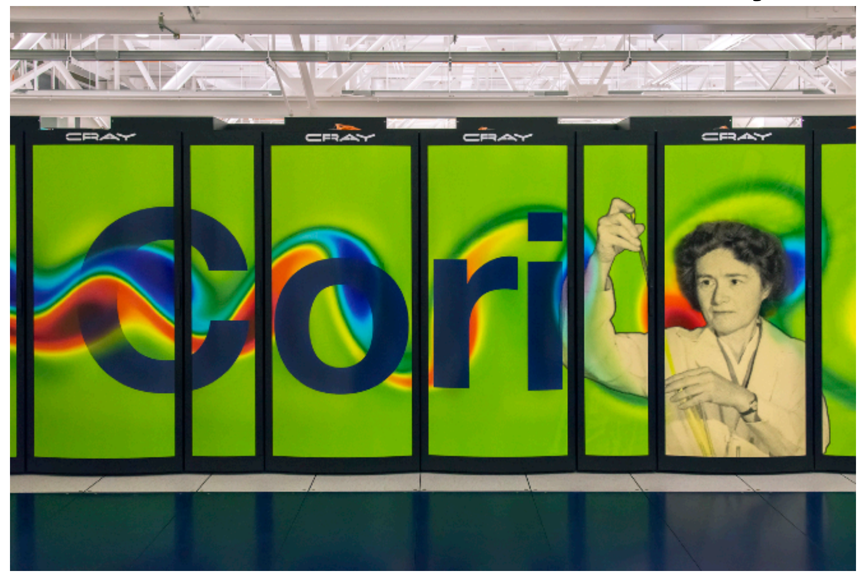
— What we need in the future

DESI is making a 3D map of the universe

Telescope at Kitt Peak near Tucson

Semi-realtime data transfer

NERSC in Berkeley



>10k cores
regular queue

320-cores
realtime
queue

Feedback
— semi-realtime QA
— survey ops next night

Nightly Processing

Quarterly/Yearly Data Assemblies

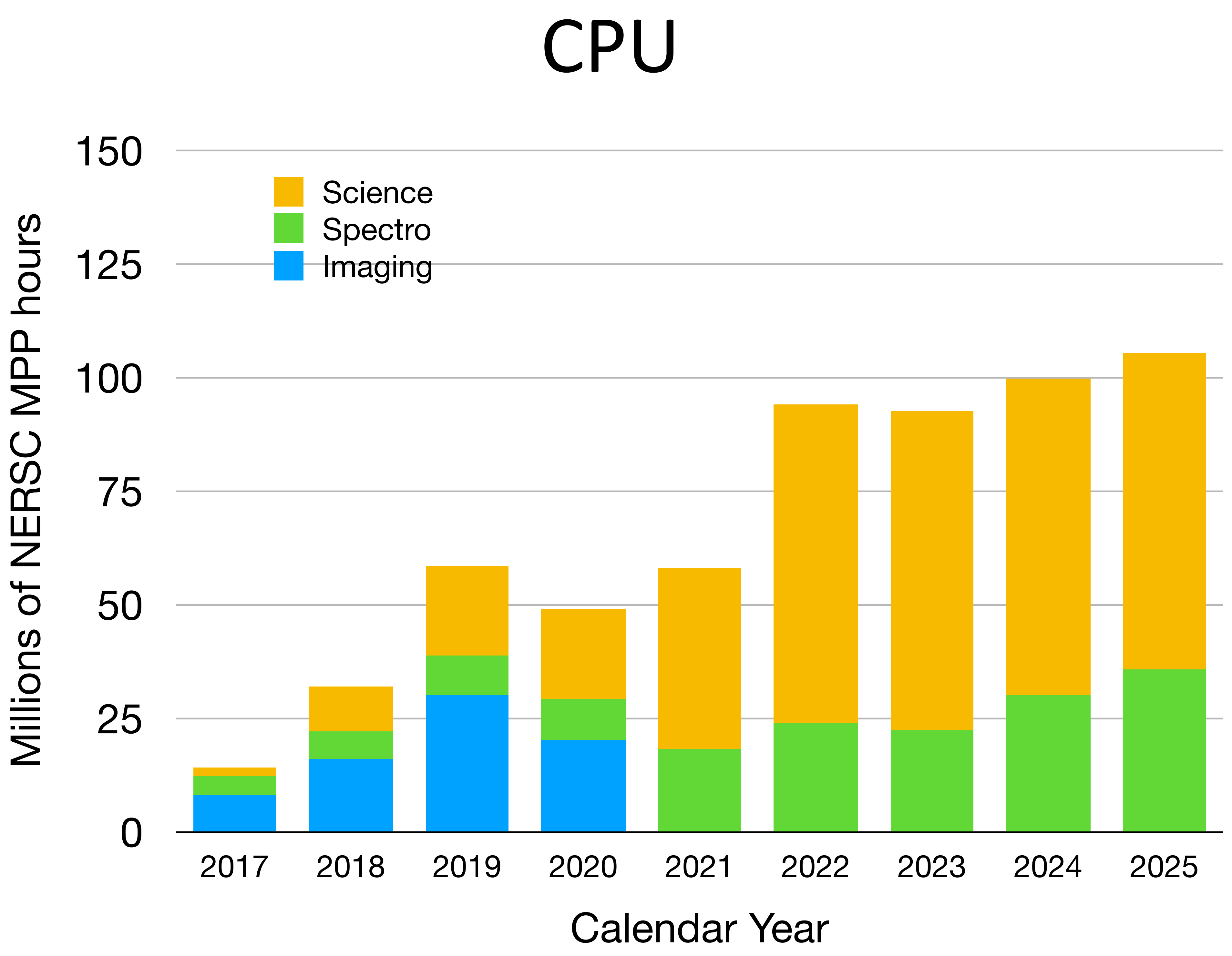
this is primary motivation for using NERSC / HPC center

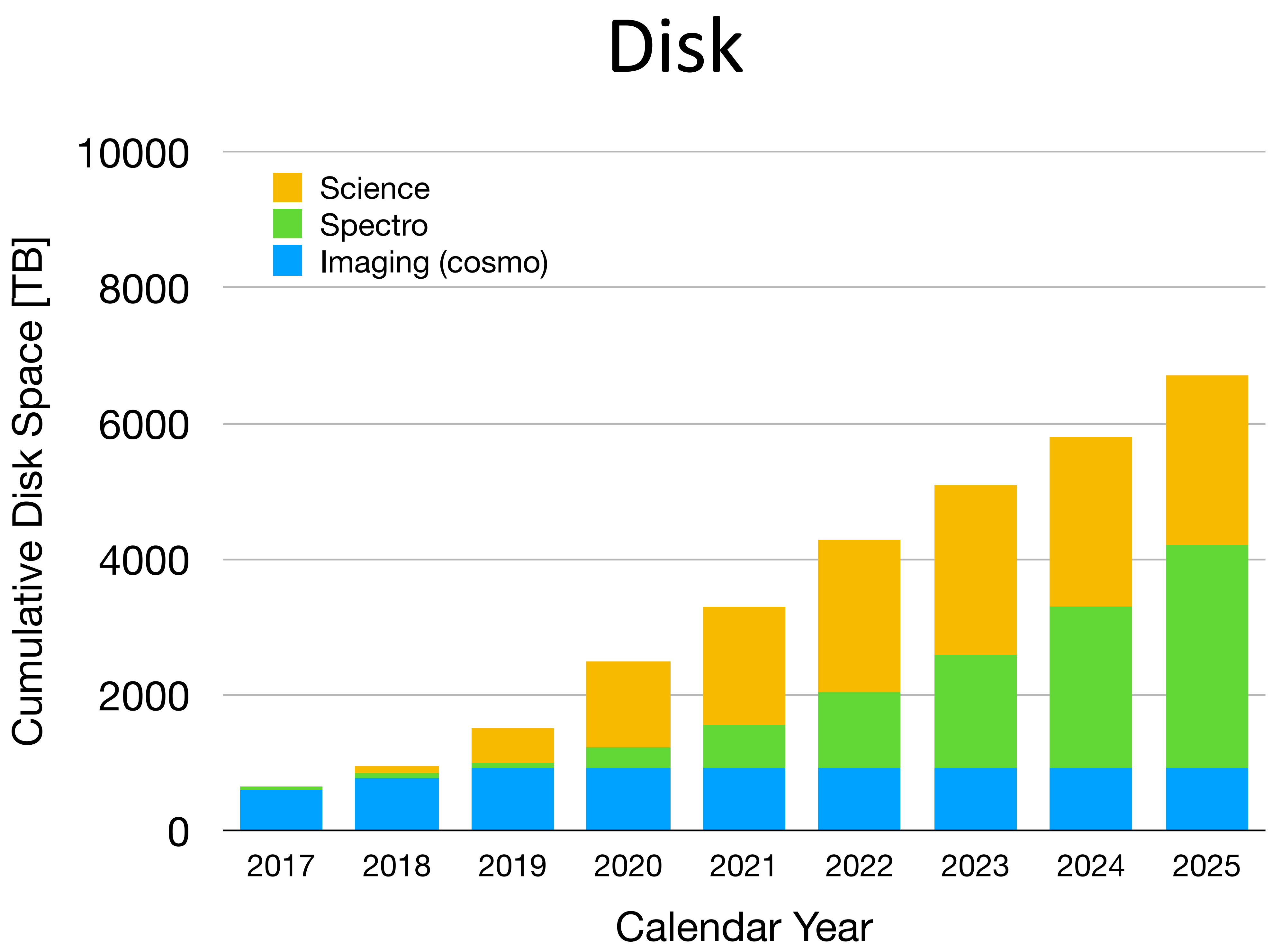
DESI Collaboration (hundreds of scientists, worldwide)

DESI Software

- All open source at https://github.com/desihub
  - relatively modern development methodology (unit tests, pull requests, code review)
    - somewhat new for cosmology/astronomy projects
  - separated DESI-specific packages from experiment-agnostic algorithm toolkits for reuse
    - initially some management pushback about who should pay for work to benefit others
    - significantly benefited DESI when a predecessor experiment used our toolkit for their project [eBOSS+redrock]
      - external contributions vetting DESI code on real data made DESI startup smoother
      - great PR at reviews (we can trust our software because others are already using it...)
- Mostly Python for developer efficiency
  - leverages numpy, scipy, numba, mpi4py, cupy for heavy-lifting
  - this model has worked very well for us to leverage non-expert coder contributions from collaboration
- Homegrown workflow / job management
  - not ideal, but didn't find alternative that was sufficiently simple but also met requirements
    - in particular, complex DAG of dependencies with very different parallelism needs

DESI Computing Requirements: CPU-hours and Petabytes





"Small Experiment" is a relative term

— Part of our success at NERSC is that we are big enough to have extra personnel resources

— I'm here to listen & help advocate, not to tell small experiments how to do their work on HPC

This is big, but fits within what NERSC can already provide

— our pragmatic requirements are driven by things other than raw FLOPS, petabytes, and bandwidth

DESI went “all-in” on NERSC / HPC

HPC = High Performance Computing
Traditional supercomputers with large highly interconnected parallelism, usually MPI

Python for developer-efficiency, but written for parallelism at HPC center from the start.

6-paid scientist-computing staff, some dedicated to HPC-specific challenges; ~20 active part-time contributors

Benefits	Challenges
Scale to tens of thousands of cores when needed	Stability
One stop shopping for daily operations, big yearly reprocessing, science analyses	Queue wait time
Account management for hundreds of collaborators	Lack of control over configuration, upgrades, policies
Extra services: jupyter, docker, realtime queue, interactive queue, globus data sharing, databases, collaboration accounts for productions	Sharing resources with thousands of users on unrelated projects (esp. when those users are breaking the system)

Messages I want Snowmass to communicate to DOE

- Experiments have computing requirements beyond FLOPS, bytes, & bandwidth
  - Mostly issues that significantly impact human efficiency but not project success/failure “cliff”
    - robustness / I/O stability / mean-time-to-failure (MTTF)
    - queue wait time (and availability of interactive and realtime queues)
    - authentication hassles (federated identities, collab accounts for productions, especially cross-site)
    - edge services: jupyter; continuous integration; cloud-like user-controlled database, webapps, etc.

- Using an HPC center isn’t “free” for experiments (or that center)
  - Not having a dedicated cluster = more personnel needed to adapt and maintain
  - Impedance mismatch: data-centric high-throughput complex-workflows vs. traditional HPC

- HPC centers need dedicated support for non-HPC workloads
  - and matching evaluation metrics, e.g. minimizing MTTF and queue-wait time instead of raw compute power and utilization percent

Opportunities

- Fund programs like NESAP: postdocs dedicated to improving computing efficiency
  - Important that they are dedicated to this task and their performance/career is evaluated for that
    - not domain science publications, not improving data/algorithm quality, not delivering last night's data or production run
  - DESI experience: fulltime postdoc saved DOE millions of CPU hours and made turnaround fast enough that it created new possibilities for how DESI could perform its survey
- Coordinate small clusters and larger HPC centers
  - e.g. small cluster under experiment control with high robustness, but easily migrate to any of several HPC centers when more capacity is needed for larger reruns
  - tier 1 / tier 2 model? Standardize minimum features/interfaces available at every center?
    - slurm + jupyter + cross-site login (federated identity) + web access to disks + ...
  - "just use Docker then you can run anywhere" isn't sufficient
- Invest in robustness, stability, data-centric & multi-site workflows, not just exascale computing
  - "99.999% Uptime Project" to compliment the "Exascale Compute Project"