Dark Energy Spectroscopic Instrument (DESI)

Making a 3D Map of the Universe at NERSC

Stephen Bailey

LBNL Physics

NERSC User Group Annual Meeting
2020-08-17

— What we do at NERSC
— Challenges
— Successes

DARK ENERGY
SPECTROSCOPIC
INSTRUMENT

U.S. Department of Energy Office of Science

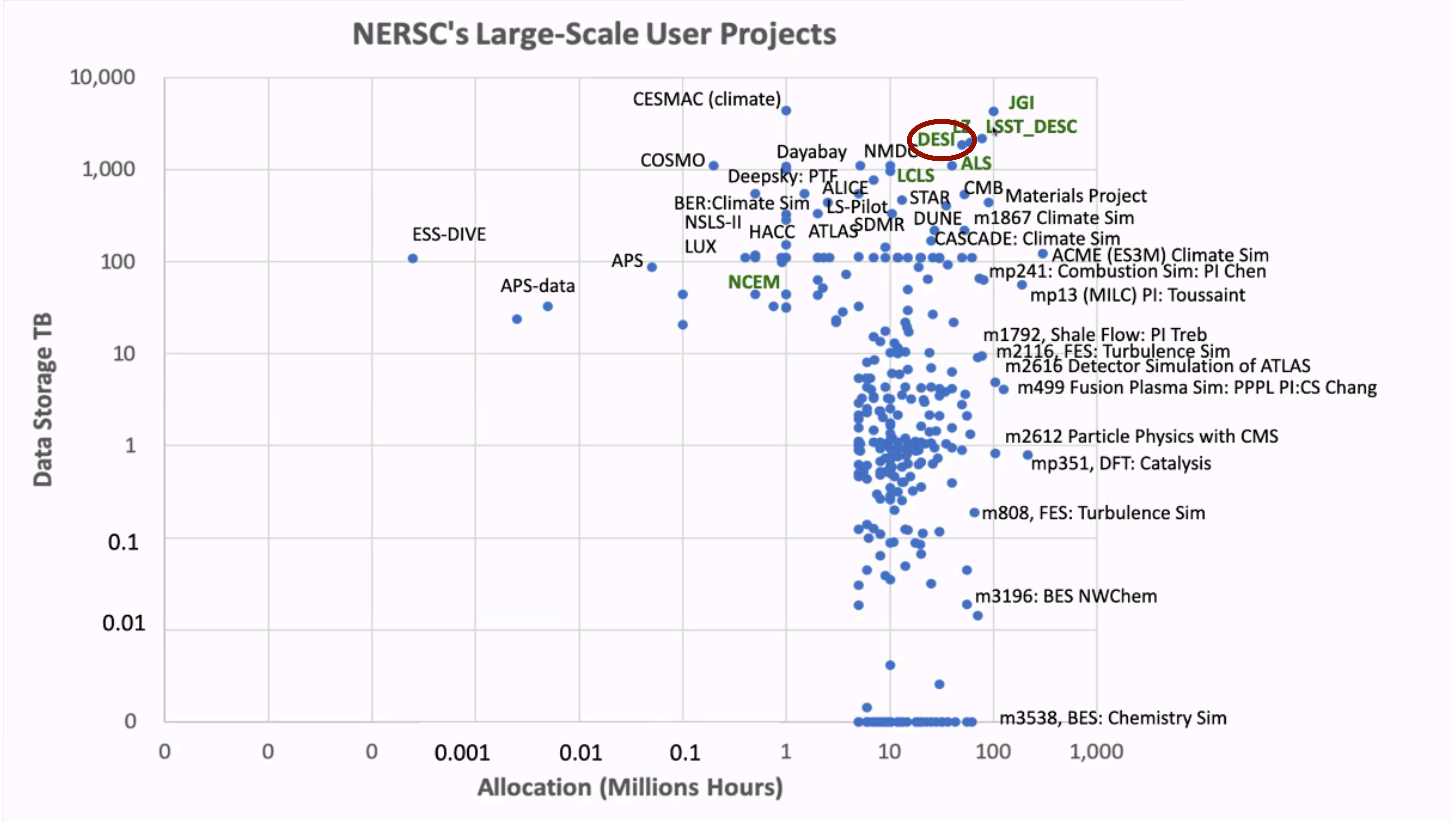
DESI@NERSC: Data Processing Basics

Nightly

- Every 10 minutes rsync new data from telescope at Kitt Peak, AZ –> NERSC
- Process data using 10 nodes of realtime queue
- Results ready by breakfast for analysis during the day, to inform the following night's observing plan
- Repeat nightly for 5 years to build 3D map of ~50M galaxies, quasars, stars
- 100s GB/night growing to ~10 PB and ~100M MPP hours/year in 5 years

Monthly / yearly

- Reprocessing runs with latest tagged code, starting from raw data
- Same code as nightly processing, but very different scaling needs
- This is the primary reason for DESI @ NERSC
- Also: one stop shopping for daily processing, big processing reruns, final science analyses


NERSC's Large-Scale User Projects
Data Storage TB
Allocation (Millions Hours)
10,000
1,000
100
10
1
0.1
0.01
0
0
0
0
0.001
0.01
0.1
1
10
100
1,000
CESMAC (climate)
JGI
LZ
LSST_DESC
DESI
COSMO
Dayabay
NMDC
ALS
Deepsky: PTF
LCLS
ALICE
CMB
Materials Project
BER:Climate Sim
LS-Pilot
STAR
NSLS-II
SDMR
DUNE
m1867 Climate Sim
ESS-DIVE
HACC
ATLAS
CASCADE: Climate Sim
LUX
APS
ACME (ES3M) Climate Sim
mp241: Combustion Sim: PI Chen
NCEM
APS-data
mp13 (MILC) PI: Toussaint
m1792, Shale Flow: PI Treb
m2116, FES: Turbulence Sim
m2616 Detector Simulation of ATLAS
m499 Fusion Plasma Sim: PPPL PI:CS Chang
m2612 Particle Physics with CMS
mp351, DFT: Catalysis
m808, FES: Turbulence Sim
m3196: BES NWChem
m3538, BES: Chemistry Sim

DESI uses the full NERSC ecosystem

Computing

- realtime, regular, interactive, debug queues

I/O

- CFS, scratch, HPSS
- Globus, rsync, data transfer nodes, portal.nersc.gov, spin container with nginx

Workflow management

- Workflow nodes, databases

Analysis & Development

- Jupyter
- Interactive & debug queues

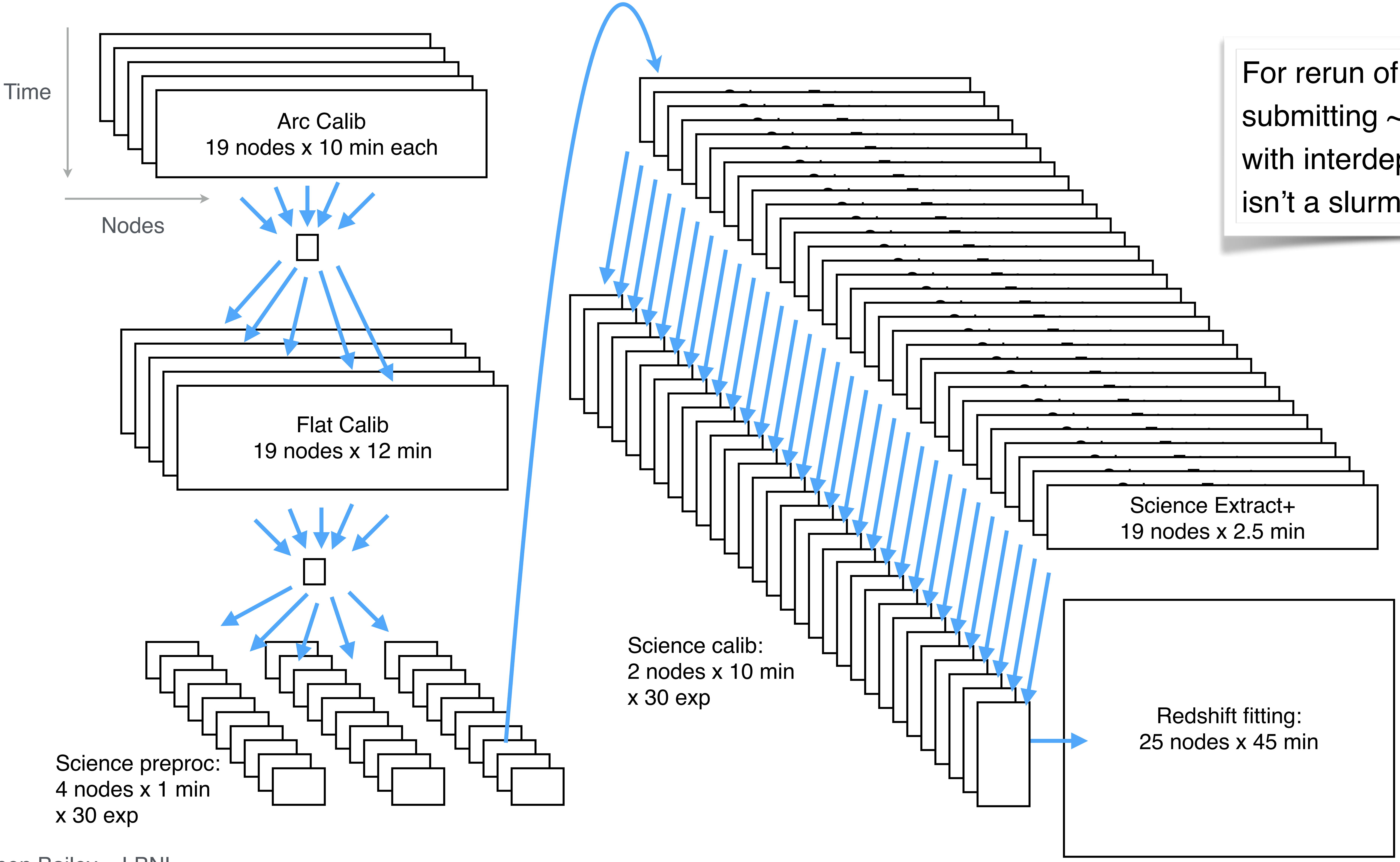
QA monitoring

- Spin, cron jobs

For DESI, it's much more than just FLOPS and I/O bandwidth

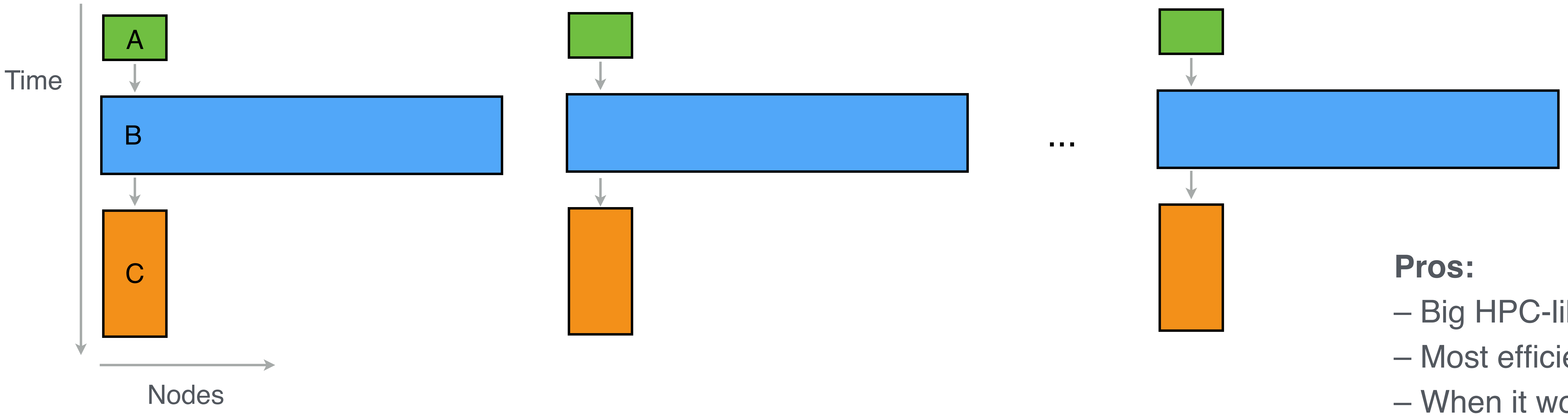
Challenge: queuing complex dependencies

One short night of data: O(100) units of work with wildly varying time and CPU needs



For rerun of 5 years of data, submitting ~110 x 365 x 5 = 200k jobs with interdependencies isn't a slurm queuing best practice

Attempt 1: Bundle each step x ~1 week of data



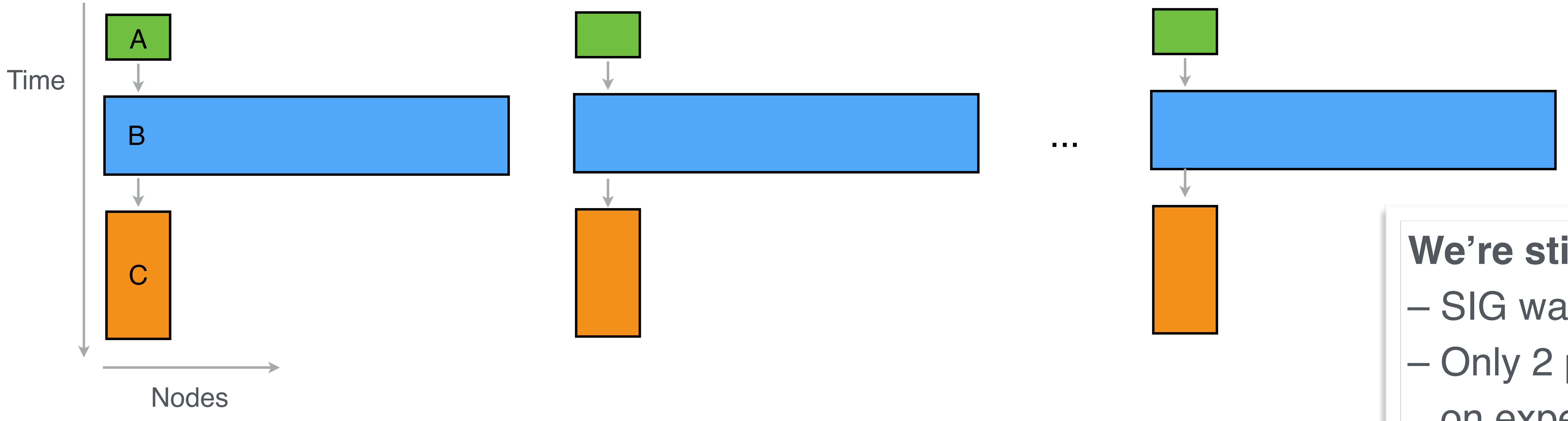
Pros:

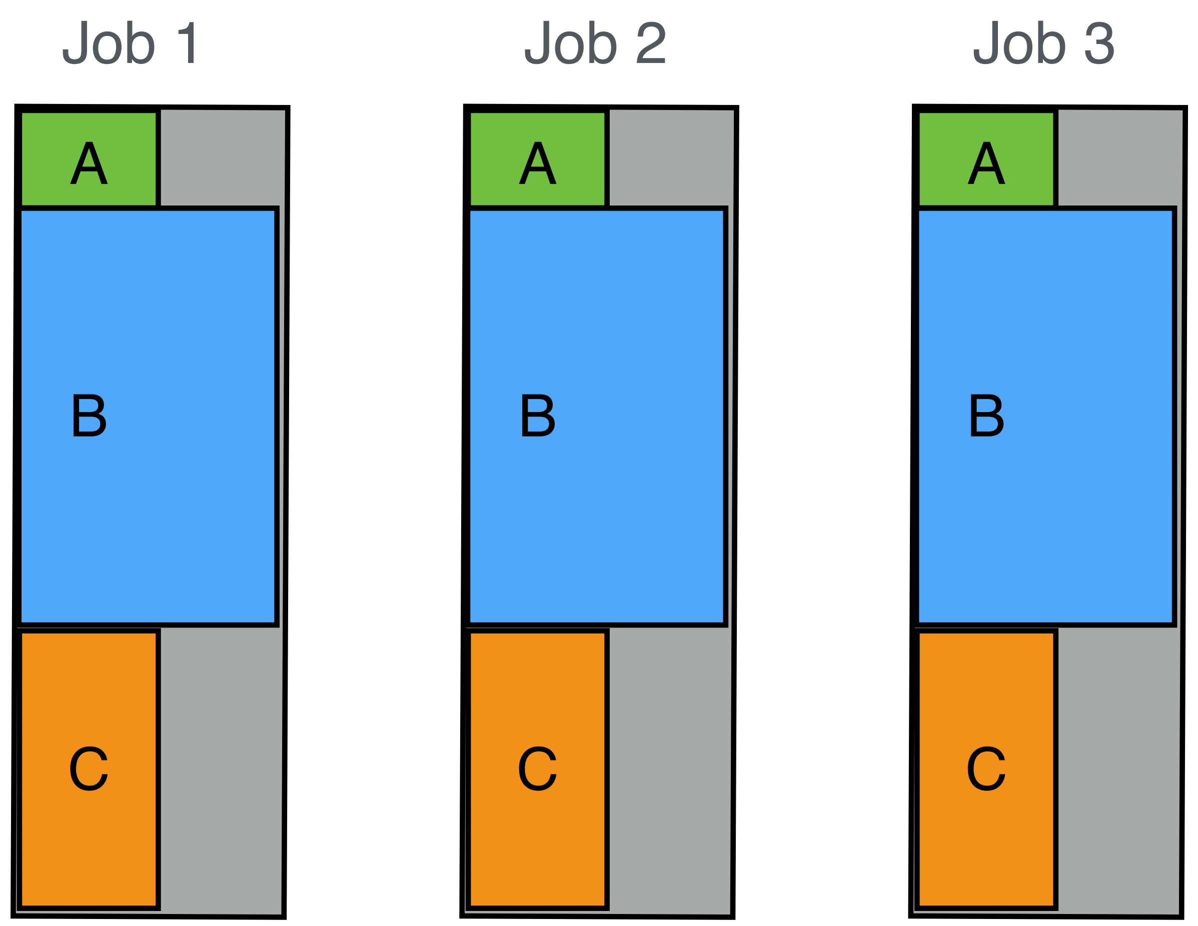
– Big HPC-like jobs
– Most efficient packing (in theory)
– When it works, it works great

Cons:

– Still requires hundreds of jobs, only 2 of which are priority scheduled, and remainder are bigger than ideal for backfill
– Job B doesn't start aging in queue until A finishes
– Couples otherwise completely independent tasks (fragile — one rank can take down all ranks)

Attempt 2: 1 exposure = 1 job, accept inefficiencies





Pros:
- Faster end-to-end for subset of data
- Decouples independent data
- Matches realtime job packing method

Cons:
- Many more jobs (~60k), won't scale to 5 years of data without job launch throttling (like Fireworks launcher)
- Wasted cores within a job during certain steps
- We've started thinking of ways that we would like to couple different exposures for algorithmic reasons

We're still working on this
- SIG was helpful for learning options
- Only 2 priority-scheduled jobs is big limit on experimental data processing
- Ironic that an HPC center has scaling problems with its scheduler: ***investment here could improve effective use of future systems***

Successes

Testing at NERSC

- Nightly cronjob — simple but effective
  - “git pull” all repos
  - run unit tests (confirms that it works at NERSC, not just travis-ci configuration)
  - run basic integration test
- Jupyter notebook orchestrates end-to-end integration test for quarterly software releases
- How will NERSC support continuous integration testing on GPUs?

Workflow resiliency [Work in Progress]

- Invest in easy *recovery from problems*, not just *avoiding problems* in the first place
- O(1%) transient job failures seems common across experimental projects
- Example: 0.1% failures on 1M images = 1000 failures
  = more than a human can easily handle if recovery is custom handwork

NESAP Optimizations

- One full-time postdoc + part-time senior consulting resulted in 10-25x speedups [Stephey, Margala, Thomas]

Summary

DESI is making a 3D map of the universe using NERSC as the primary computing center

- Yearly reprocessing drives the need for HPC
- Also benefit from one stop shopping for data processing + science analyses

Challenges

- Queueing N>>1 algorithmic steps with very different parallelism needs
- Not covered here
  - Downtimes, mean time-to-failure
  - I/O performance fluctuations, especially on metadata
  - Coordinating dataflow within NERSC (HPSS, scratch, CFS, BB, /tmp)

Successes

- Testing @ NERSC
- Resilient workflows [WIP]
- NESAP code optimizations

Thanks to

- You (NUG meeting attendees)
- NERSC (consulting, NESAP, liaisons)
- Our funding agencies
- David Lawrence for organizing NUG SIG for Experimental Facilities

DARK ENERGY SPECTROSCOPIC INSTRUMENT

U.S. Department of Energy Office of Science