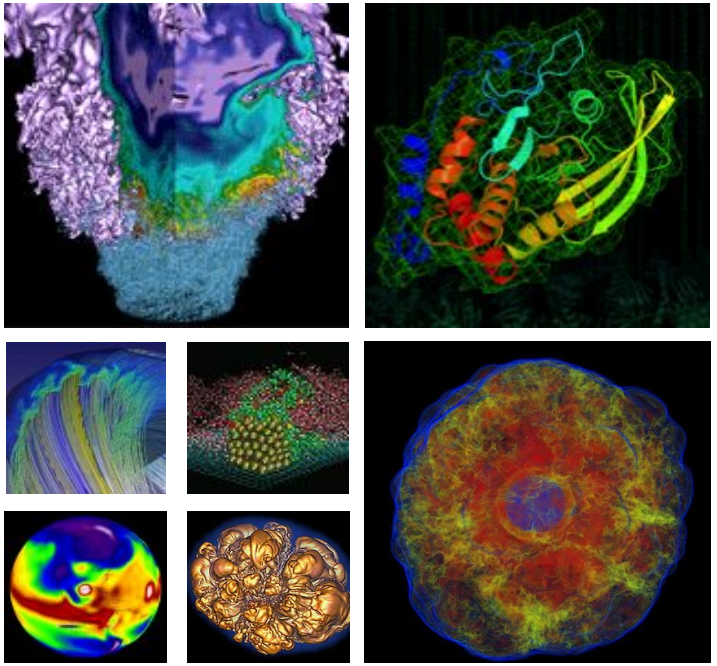
NERSC-10 Workload Analysis (Data from 2018)



Brian Austin et al.

April 1, 2020

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Contributors

Wahid Bhimji	Richard Gerber	Mustafa Mustafa
Chris Daley	Helen He	Rollin Thomas
Tom Davis	Kadidia Konate	Carey Whitney
Steve Farrell	Glenn Lockwood	Nick Wright
NERSC Computational Systems Group		Zhengji Zhao
NERSC Infrastructure Systems Group		

Workload analysis is key to procuring productive, high performing systems for science.

Workload analysis asks: "How do how users exercise the available computational resources?" (mine every log you can find)

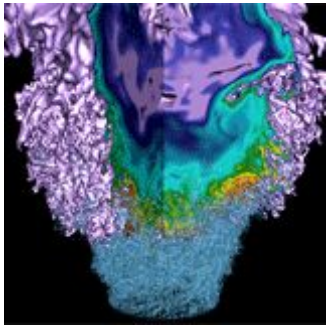
- What types of algorithms need to be supported?
- What efforts are needed to transition to future architectures?
- Which resources are underutilized? oversubscribed?

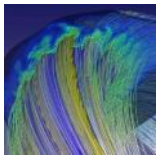
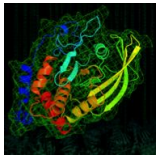
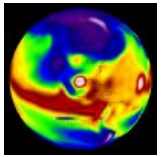
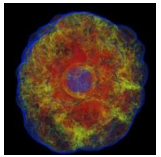
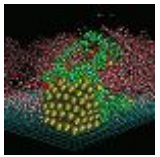
Other activities complement workload analysis:

- Requirements reviews - interview users about future needs and goals.
- Work*flow* analysis - operational and data dependencies.
- Benchmark analysis - performance characteristics of individual codes.
- System monitoring - LDMS, OMNI, TOKIO

Requirements for future procurements combine all these sources of information.

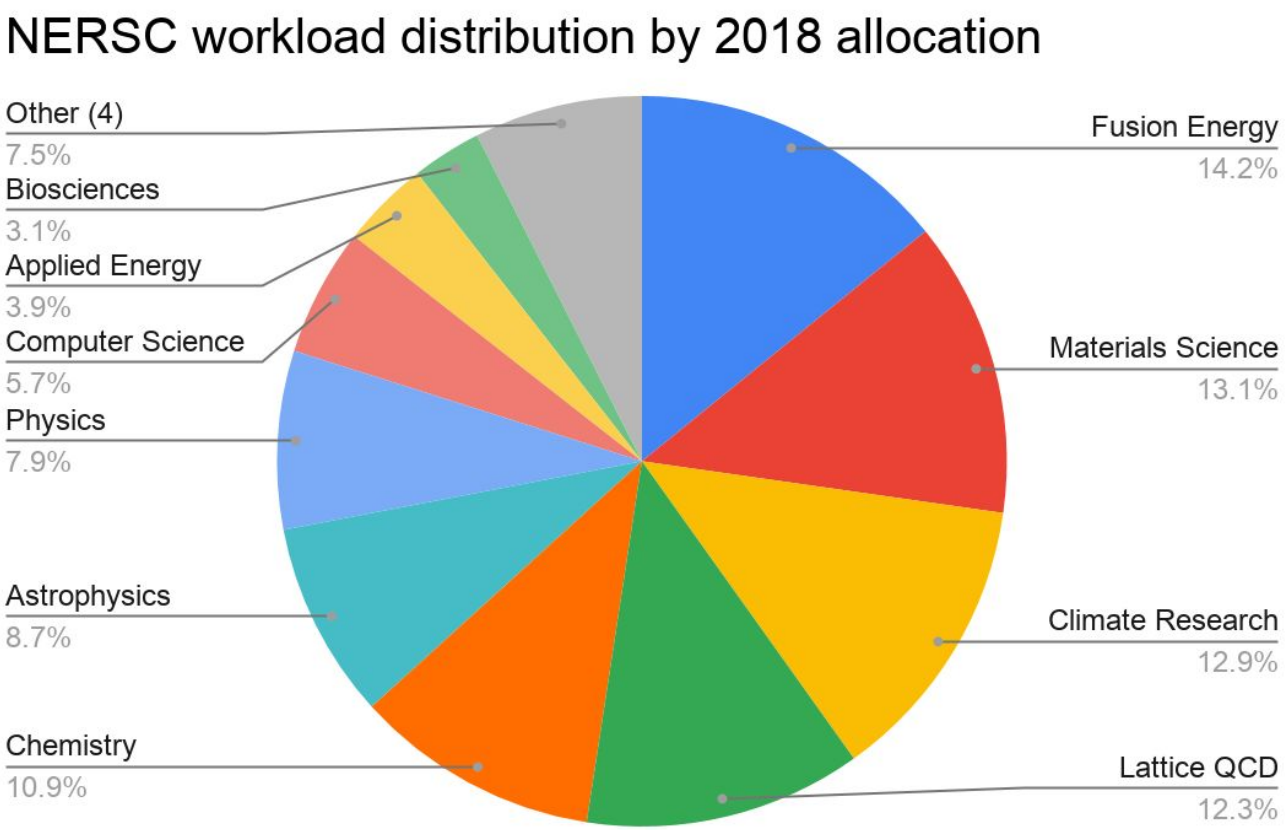
Application Demographics



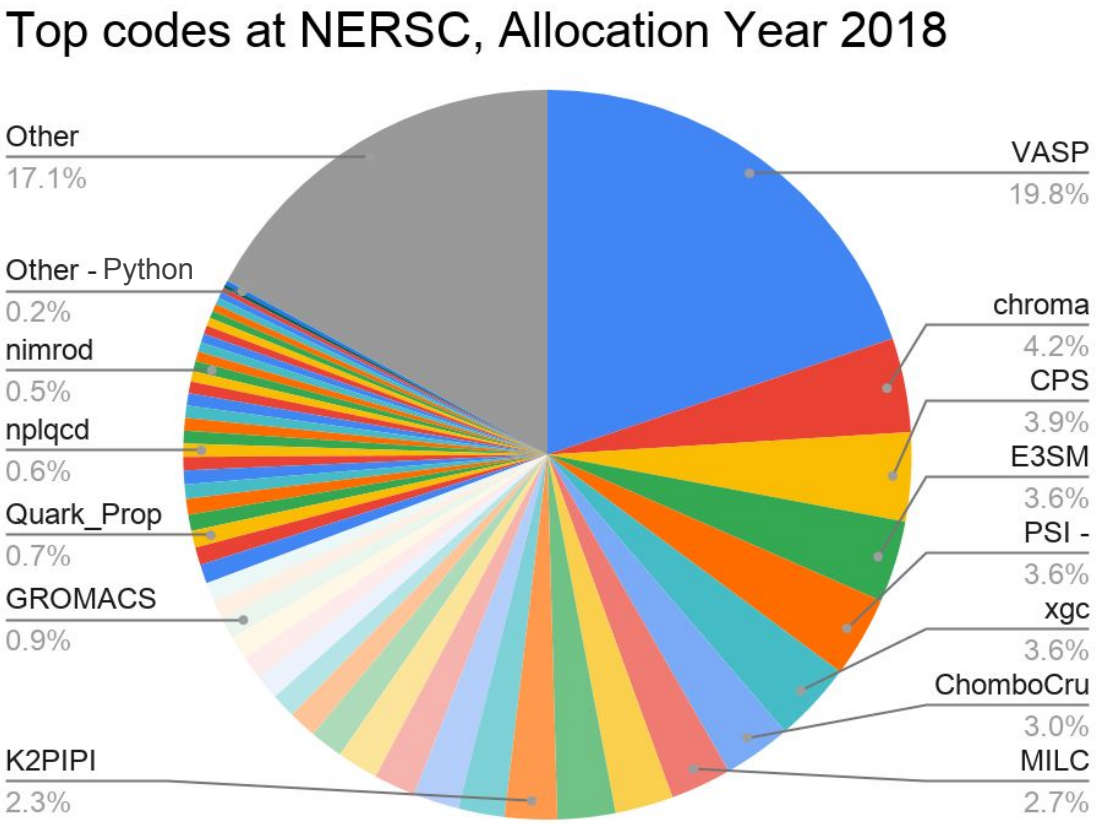
NERSC

NERSC serves a broad range of science disciplines for the DOE office of Science.



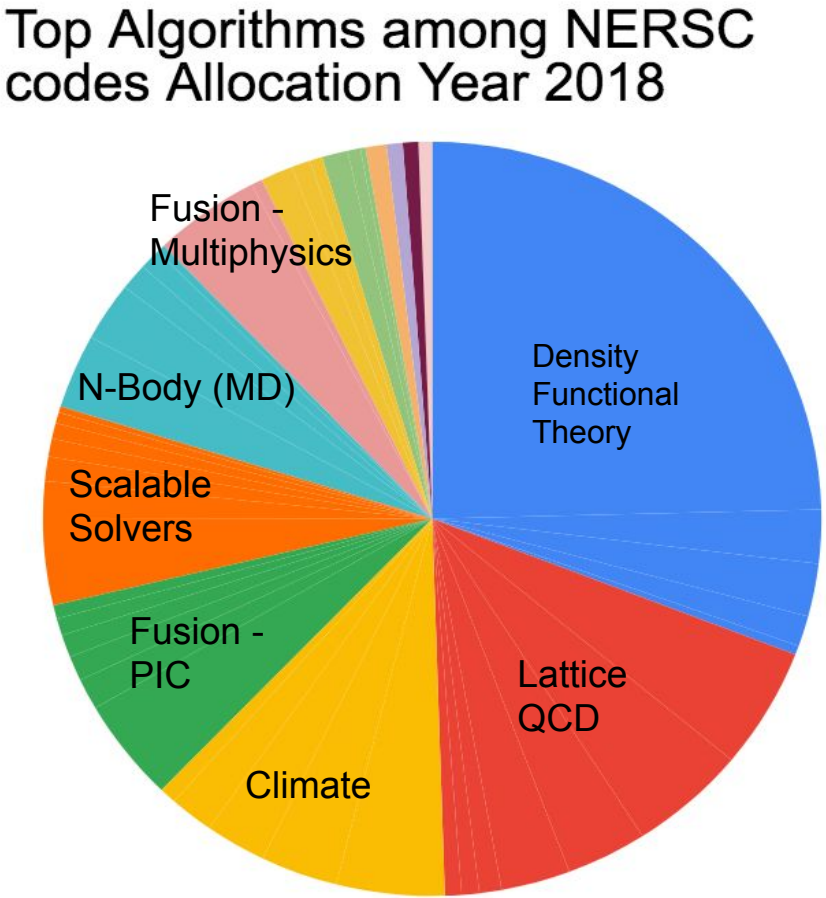
- **Nearly 4300 active users**
- **Over 850 projects**

NERSC workload is extremely diverse, but not evenly divided.



- **10 codes make up 50% of workload.**
- **20 codes make up 66% of workload.**
- **50 codes make up 84% of workload.**
- **Remaining codes (over 600) make up 16% of workload.**

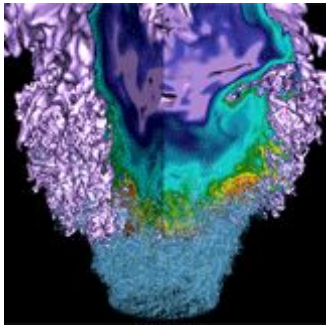
Many codes implement similar algorithms.

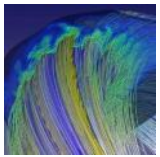
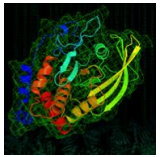
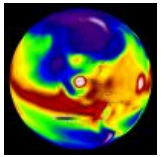
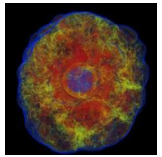
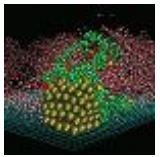


Regrouped top 50 codes by similar algorithms.

A small number of benchmarks can represent a large fraction of the workload.

Concurrency & Job Size

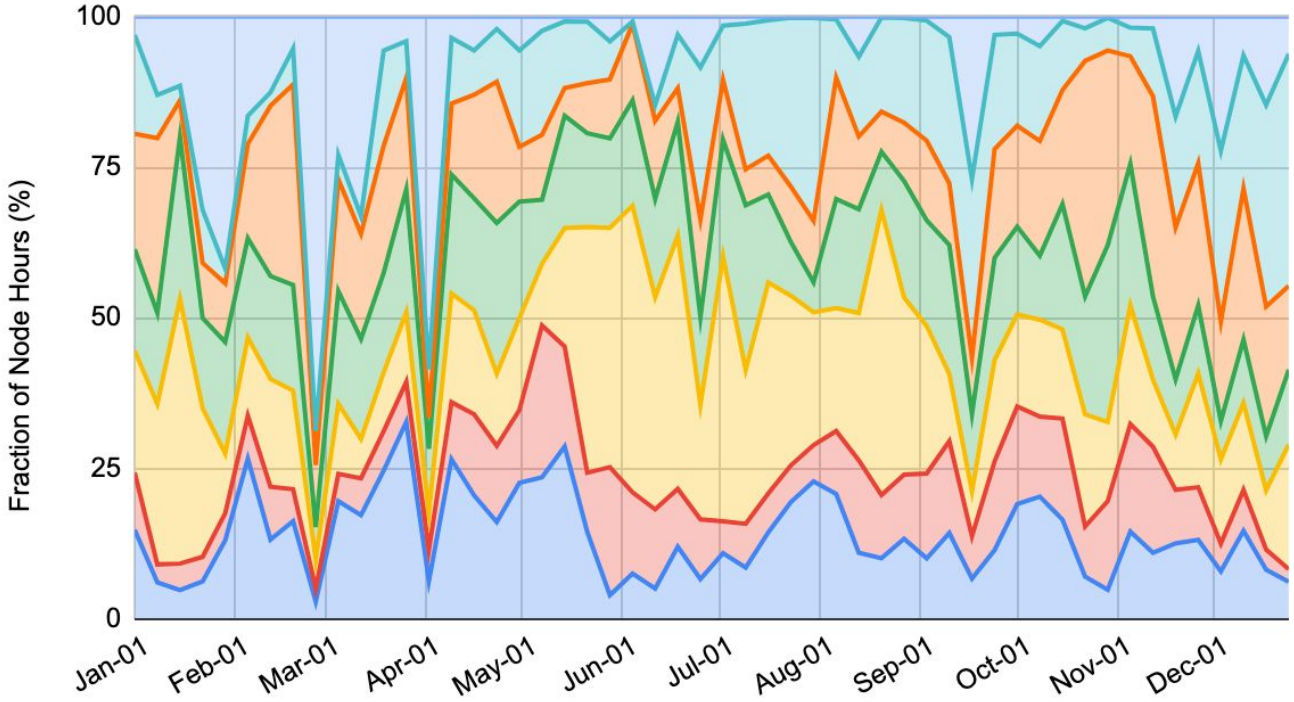


NERSC

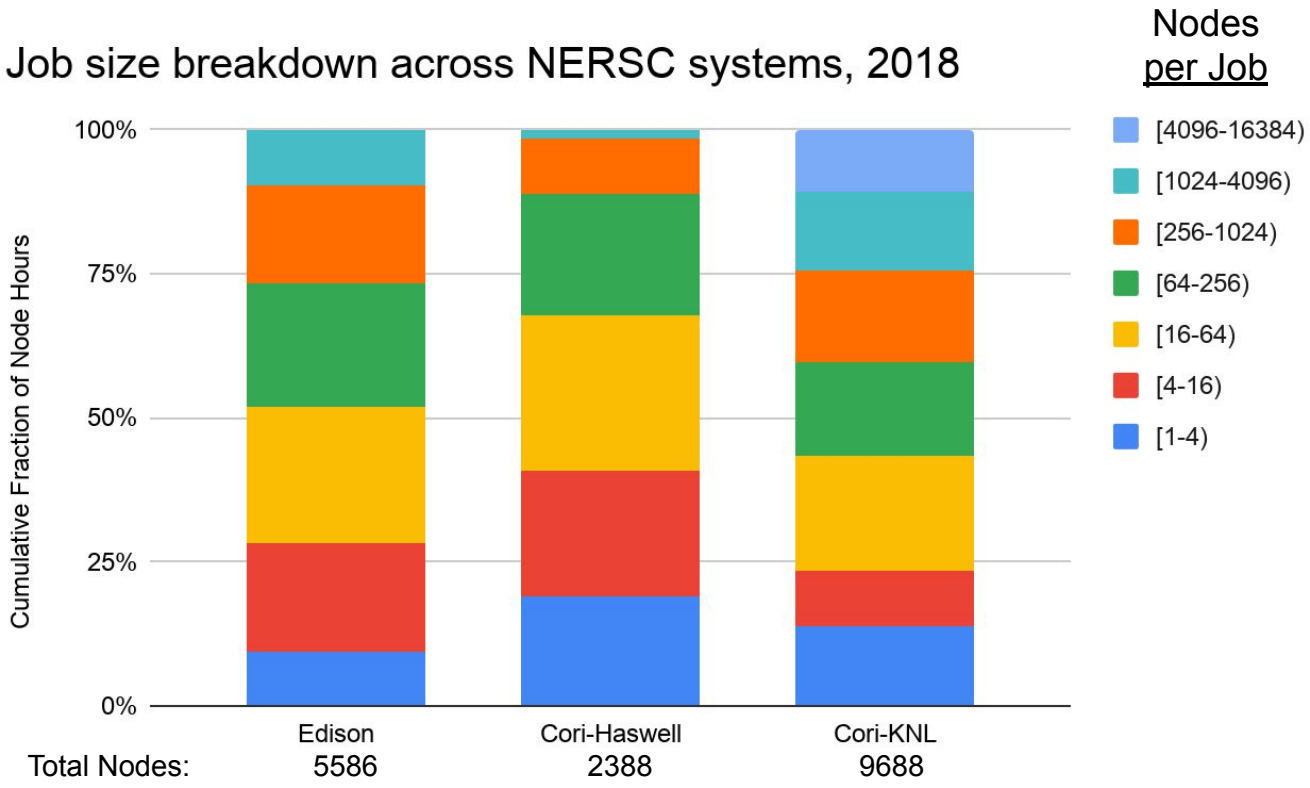
High concurrency jobs are an important component of the NERSC workload.

Job Size Breakdown on Cori KNL, 2018

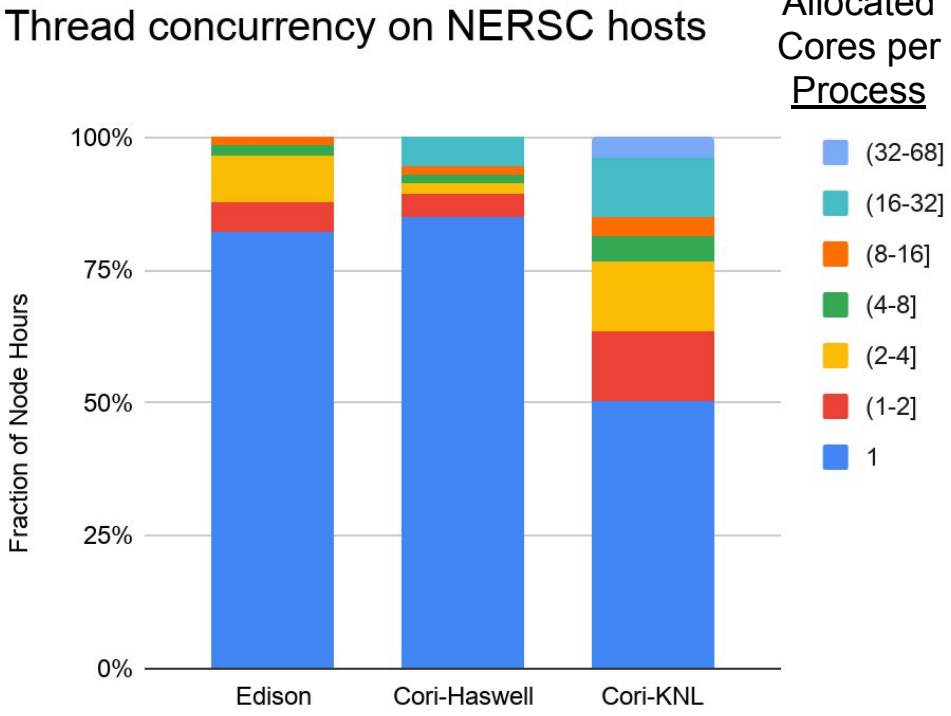


Nodes per Job	Node Hours
[4096 - 9688]	11%
[1024 - 4096)	14%
[256 - 1024)	16%
[64 - 256)	16%
[16 - 64)	20%
[4 - 16)	10%
[1 - 4)	14%

NERSC's largest jobs run on KNL



KNL jobs use more threads.

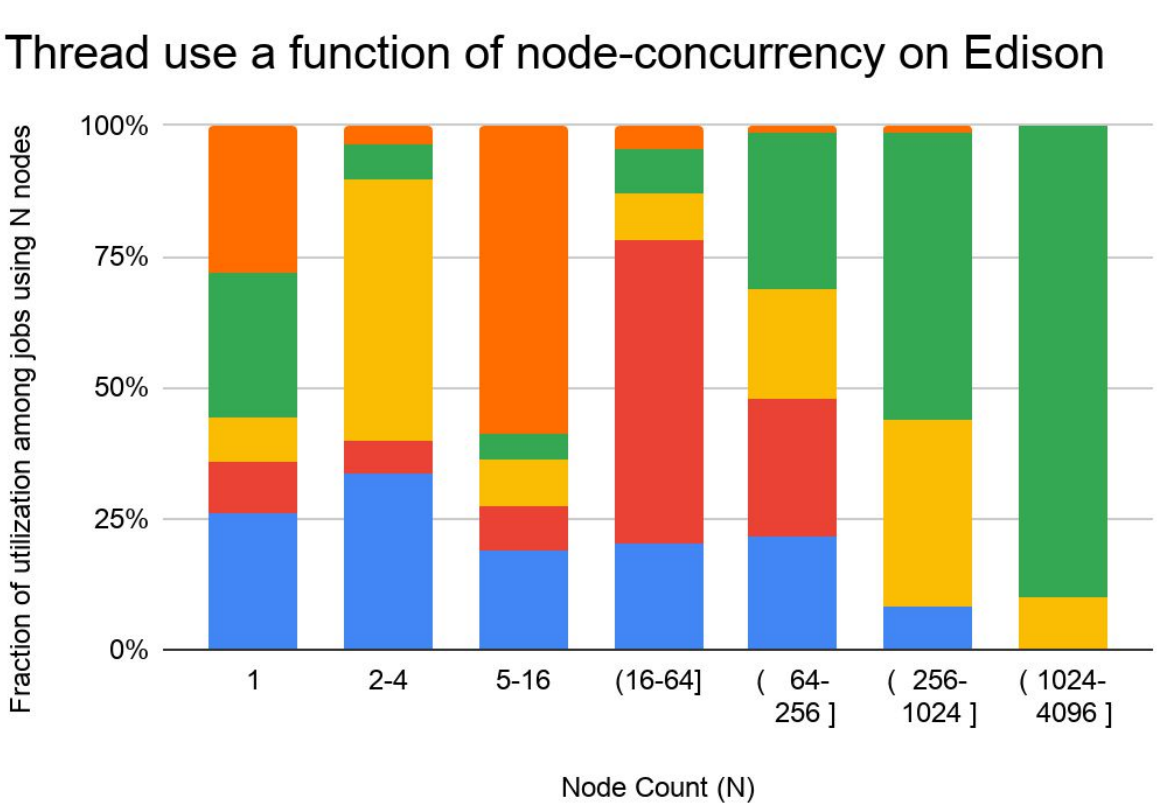


- 80% of Edison & Haswell uses 1 thread
- 50% of KNL workload uses 1 thread.
- 20% of KNL workload uses over 8 threads

Why so different?

- **Memory capacity?** Unlikely…
  Thread utilization does not match memory per process ratios.
- **Runtime balance?** Maybe…
  Slower KNL cores could shift the MPI / OpenMP trade-off.
- **Code readiness?** Maybe…
  OpenMP - savvy users may prefer KNL

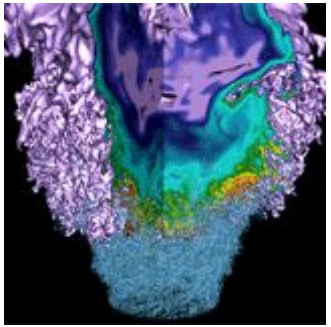
High concurrency jobs use more threads.

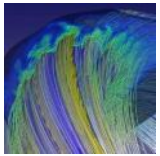
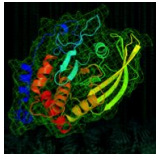
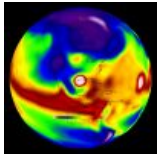
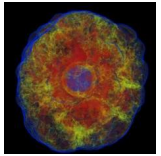
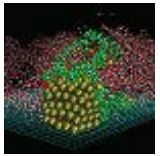


OpenMP use increases at large scales where MPI scaling inefficiencies outweigh (on-node) OpenMP inefficiencies.

Zero single-threaded jobs using over 1024 nodes.

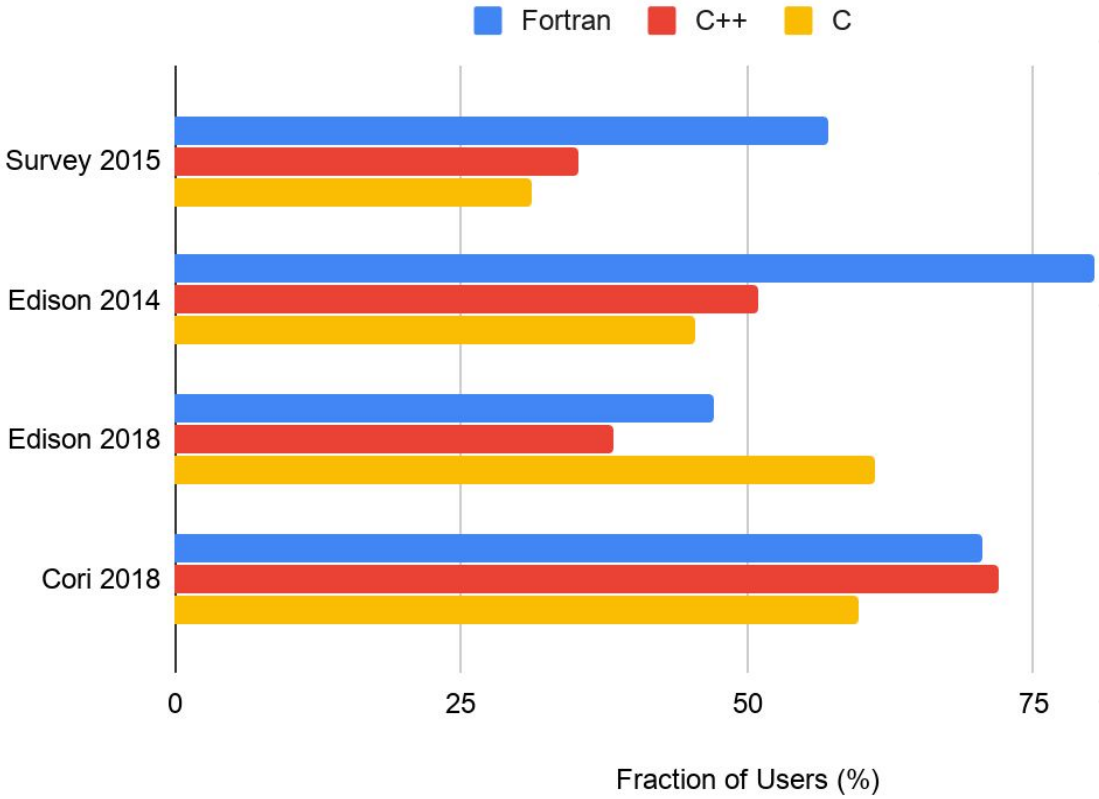
Languages, Libraries & Programming Environment



NERSC

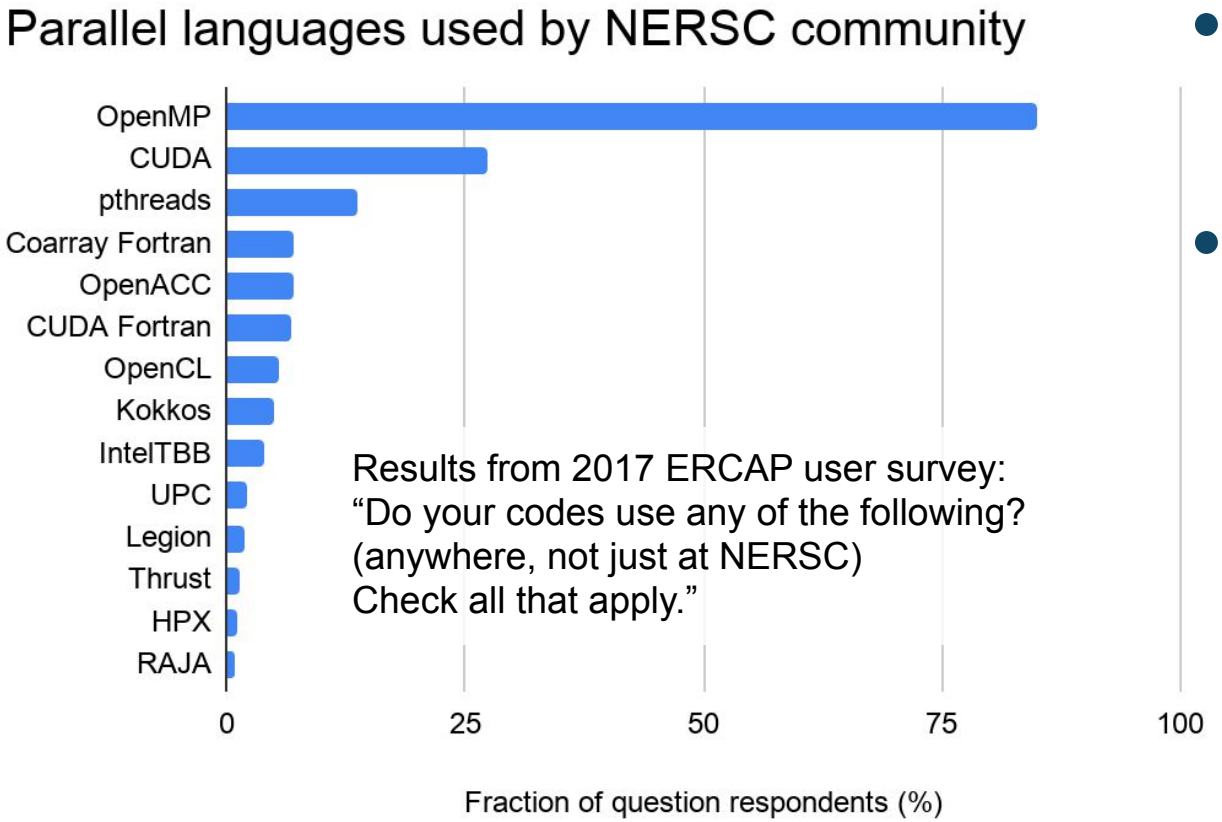
Compiled languages used at NERSC



Totals exceed 100% because some users rely on multiple languages.

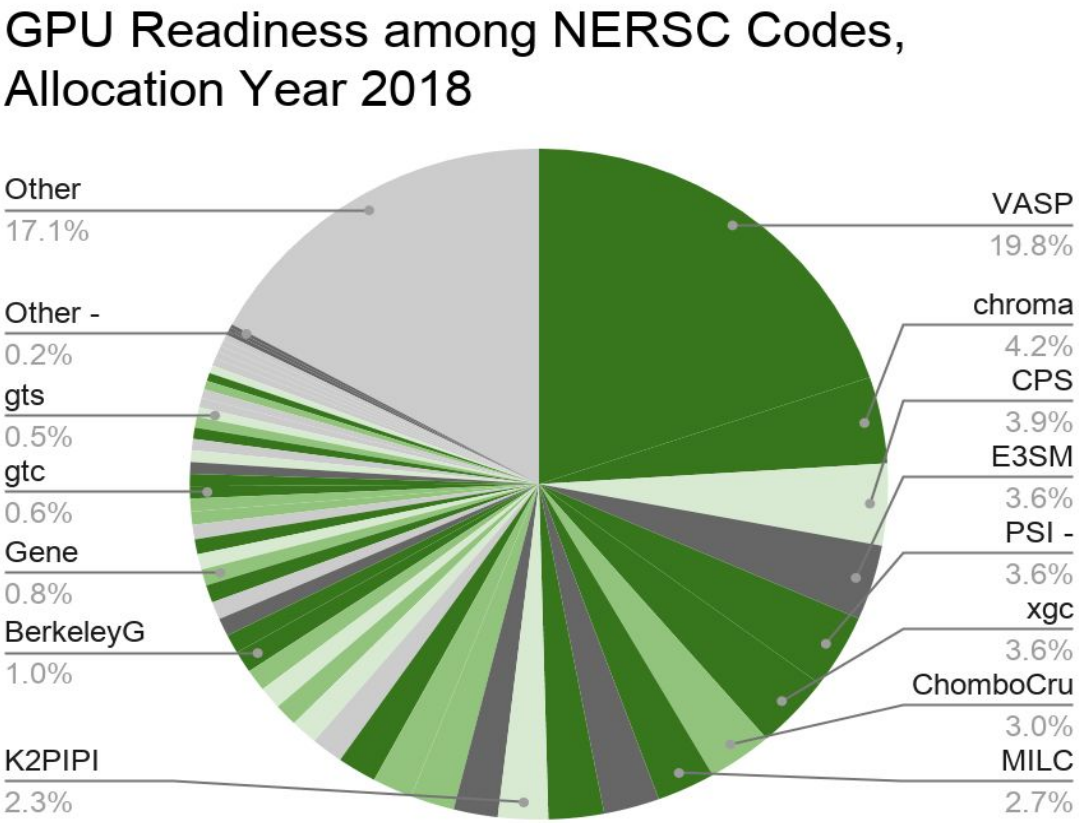
- Fortran remains a common language for scientific computation.
- Noteworthy increases in C++ and multi-language
- Language use inferred from runtime libraries recorded by ALTD. (previous analysis used survey data)
  - ALTD-based results are mostly in line with survey data.
  - No change in language ranking
  - Survey underrepresented Fortran use.
- Nearly ¼ of jobs use Python.

OpenMP has been widely adopted by NERSC users.



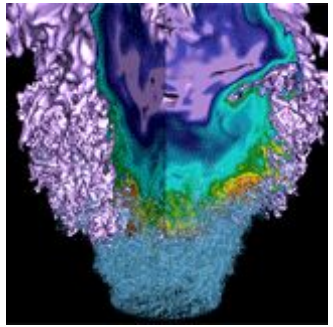
- MPI used in vast majority of compiled codes.
  (Not included in survey)
- Over 25% use CUDA

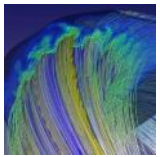
Much of the NERSC workload already runs well on GPUs

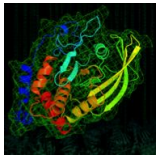


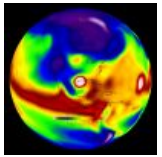
GPU Status & Description	Fraction
Enabled: Most features are ported and performant	43%
Kernels: Ports of some kernels have been documented.	8%
Proxy: Kernels in related codes have been ported	14%
Unlikely: A GPU port would require major effort.	10%
Unknown: GPU readiness cannot be assessed at this time.	25%

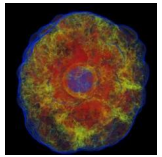
Memory

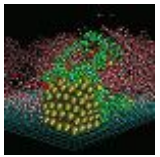






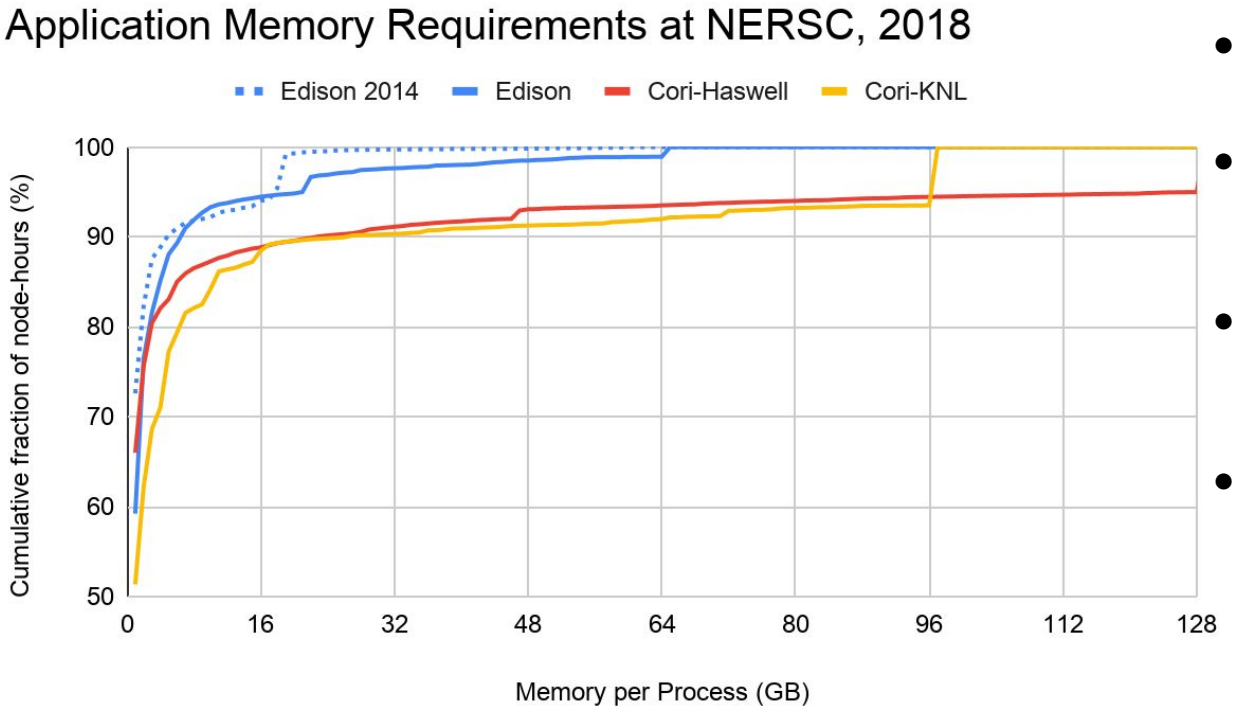






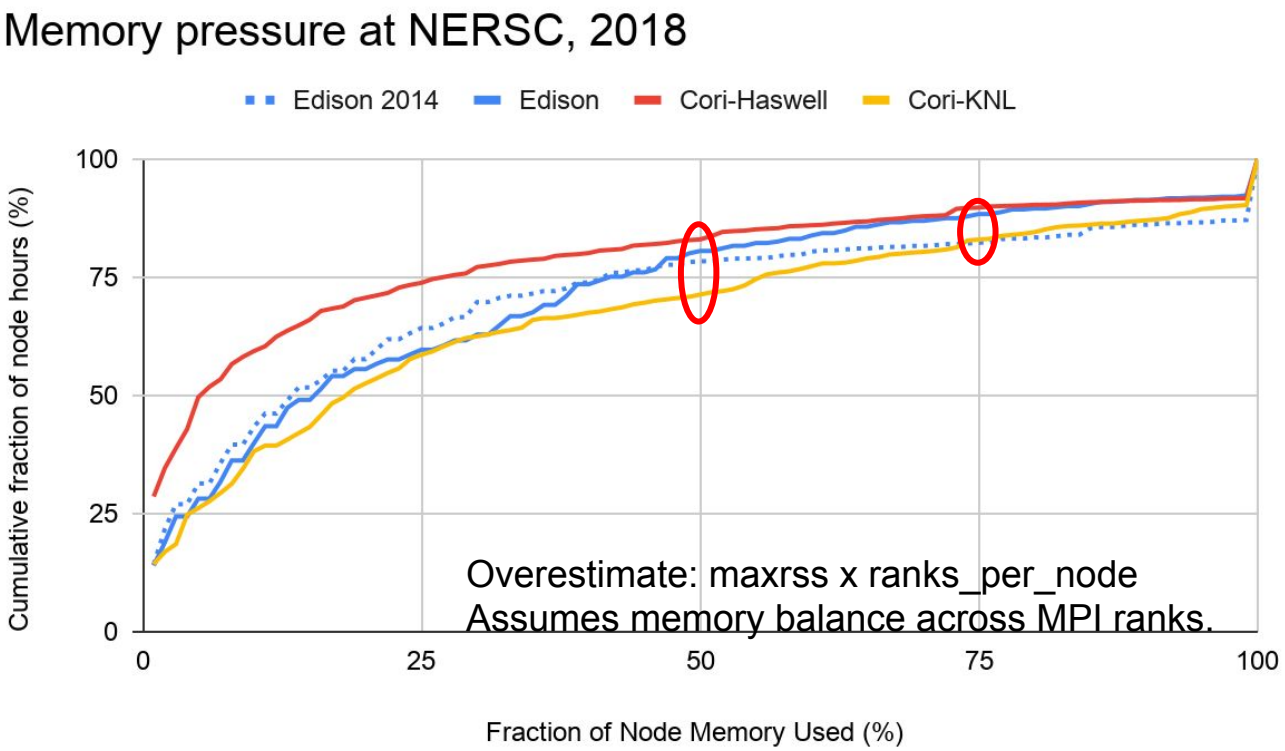
NERSC

Memory per Process



- 50% of NERSC workload requires less than 1GB / MPI rank
- 10% of NERSC workload requires more than 25 GB per MPI rank (max over ranks within job)
- Edison's "high-memory" workload uses noticeably less memory than Cori.
- On Cori, Haswell and KNL memory use is similar

Memory pressure is common among jobs at NERSC.

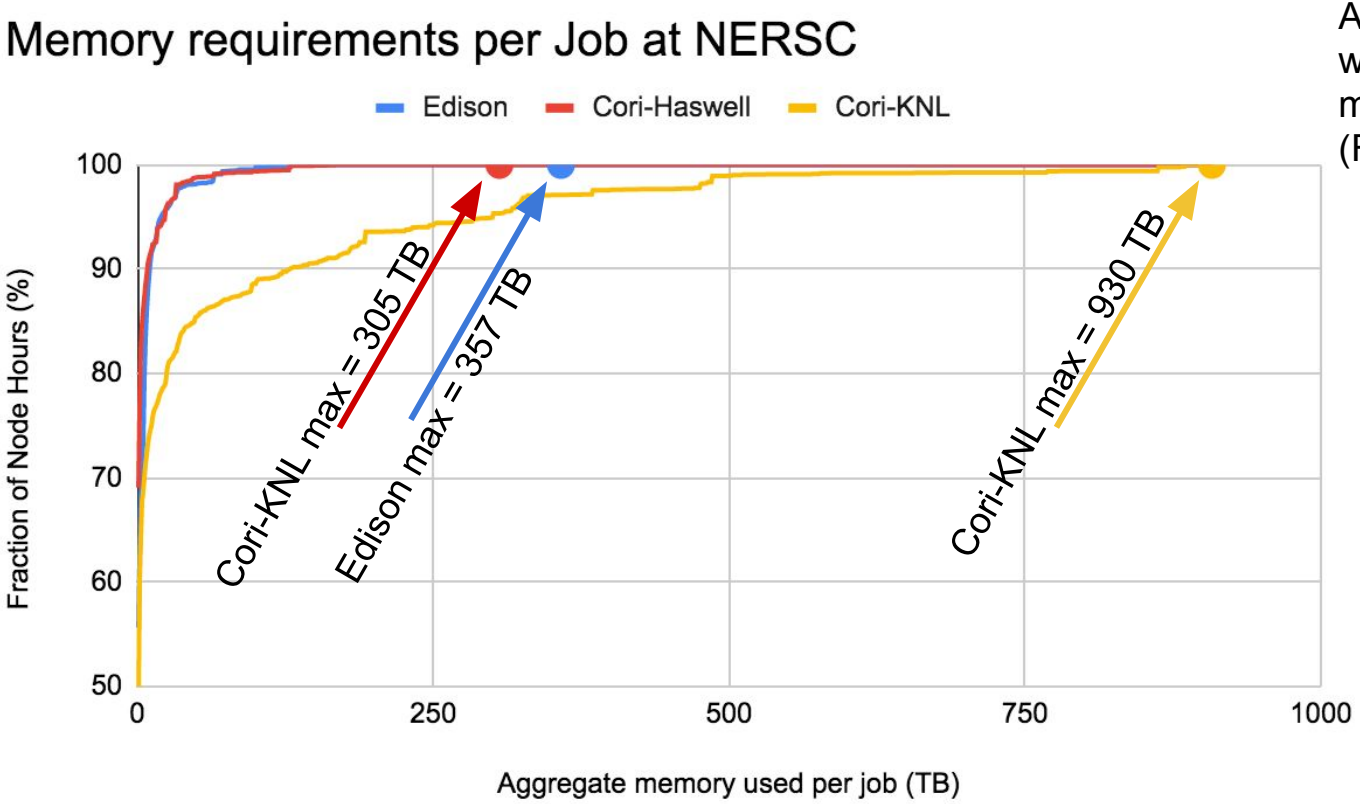


About 15% of NERSC workload uses more than 75% of the available memory per node.

And ~25% uses more than 50% of available memory.

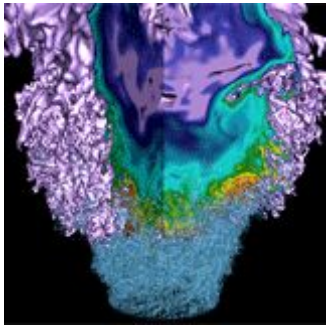
Compare to memory-per-rank analysis: this memory pressure can be relieved by strong-scaling.

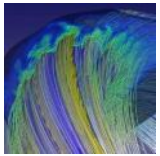
Jobs requiring large total memory run on Cori-KNL

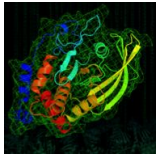


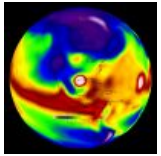
A small fraction (~1-2%) of the workload uses Cori-KNLs full memory capacity.
(Full-scale jobs *do* exist.)

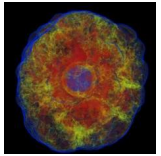
Workload Evolution

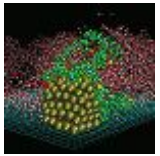






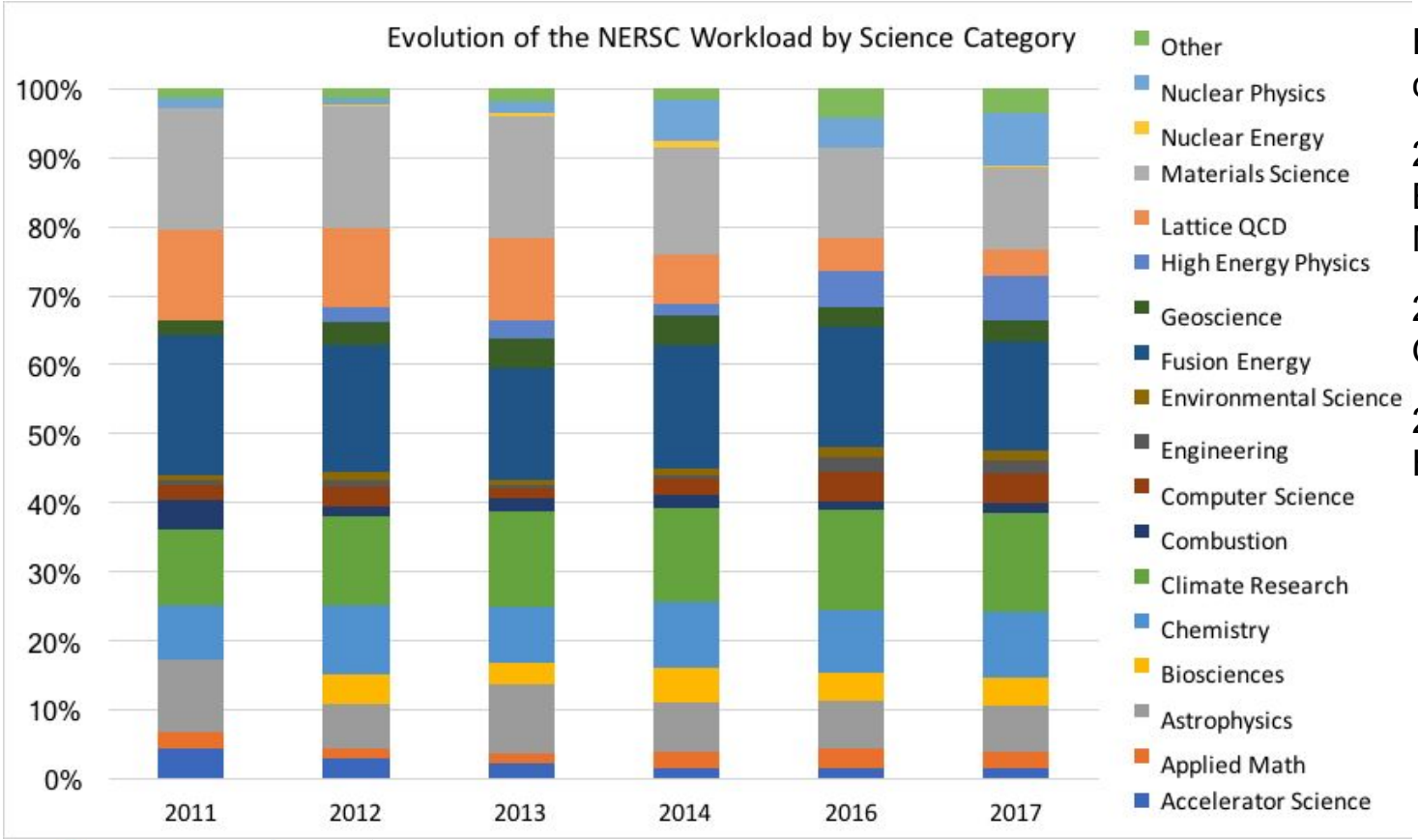






NERSC

NERSC's workload historically changes slowly



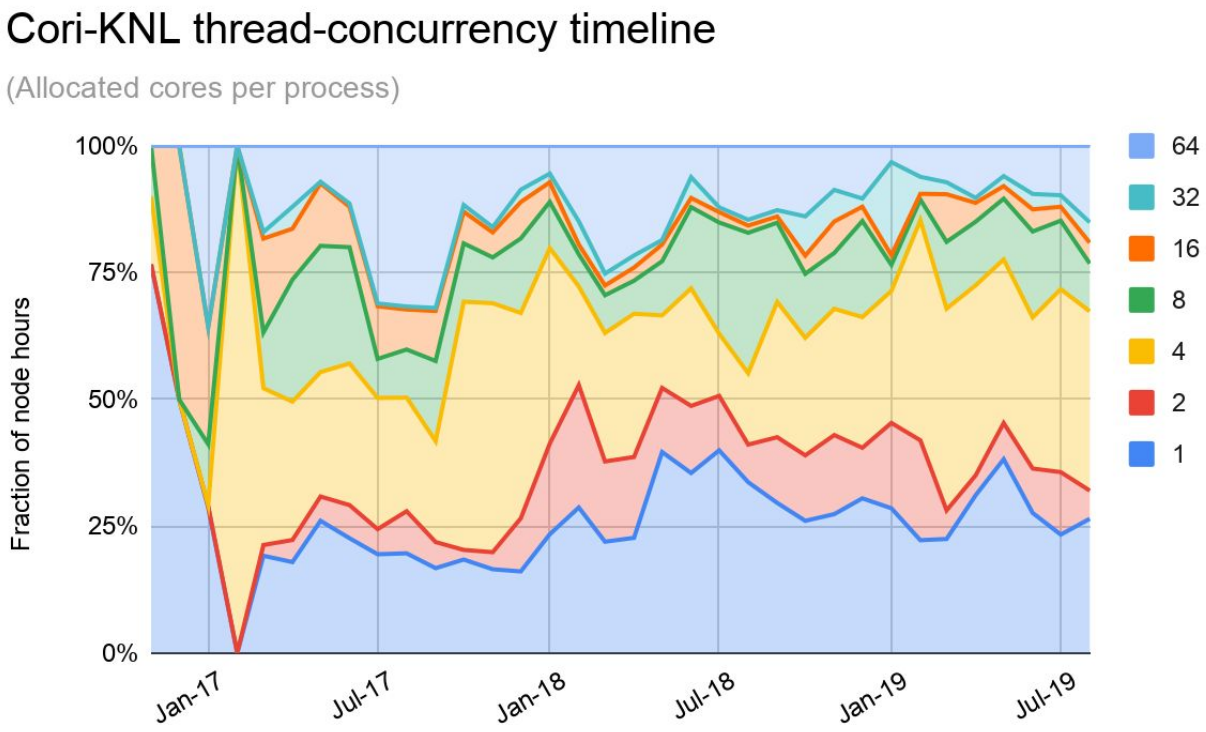
Fractions are roughly constant.

2012:
Biosciences debut at 4%
No change since.

2014:
QCD halved to 5%.

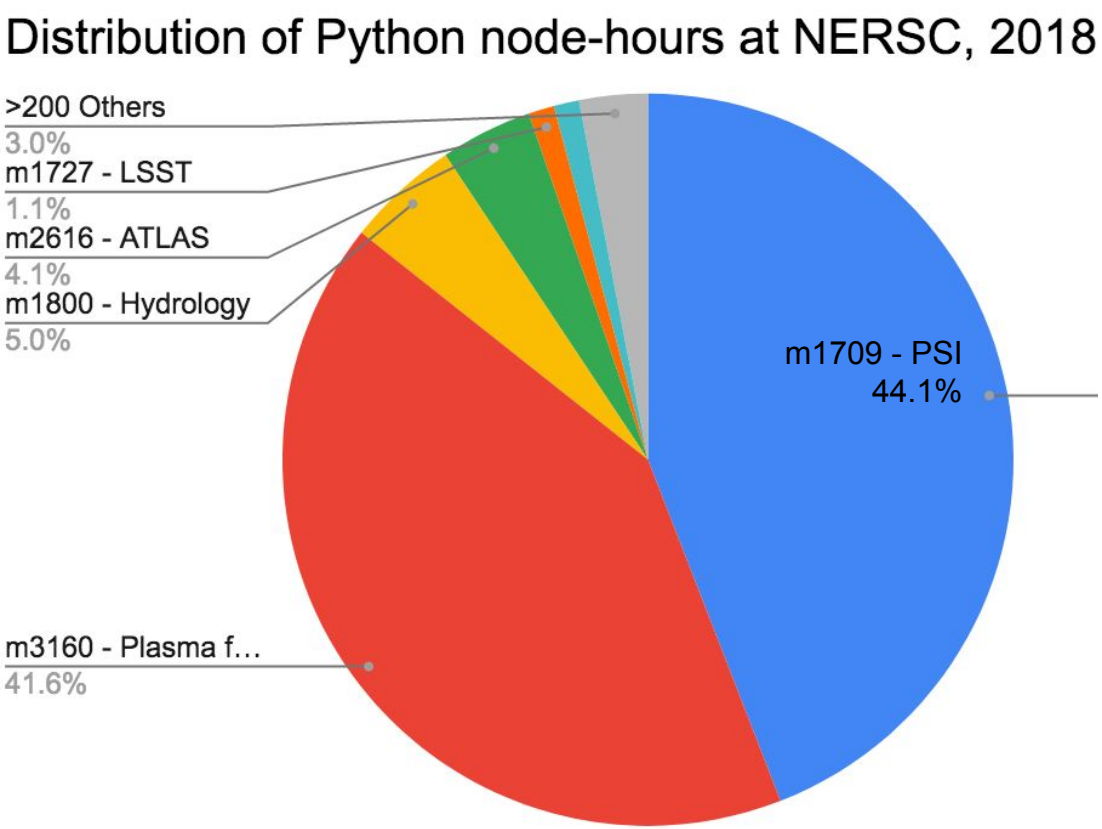
2016:
HEP doubled to 5%.

Users adapt to new architectures slowly.



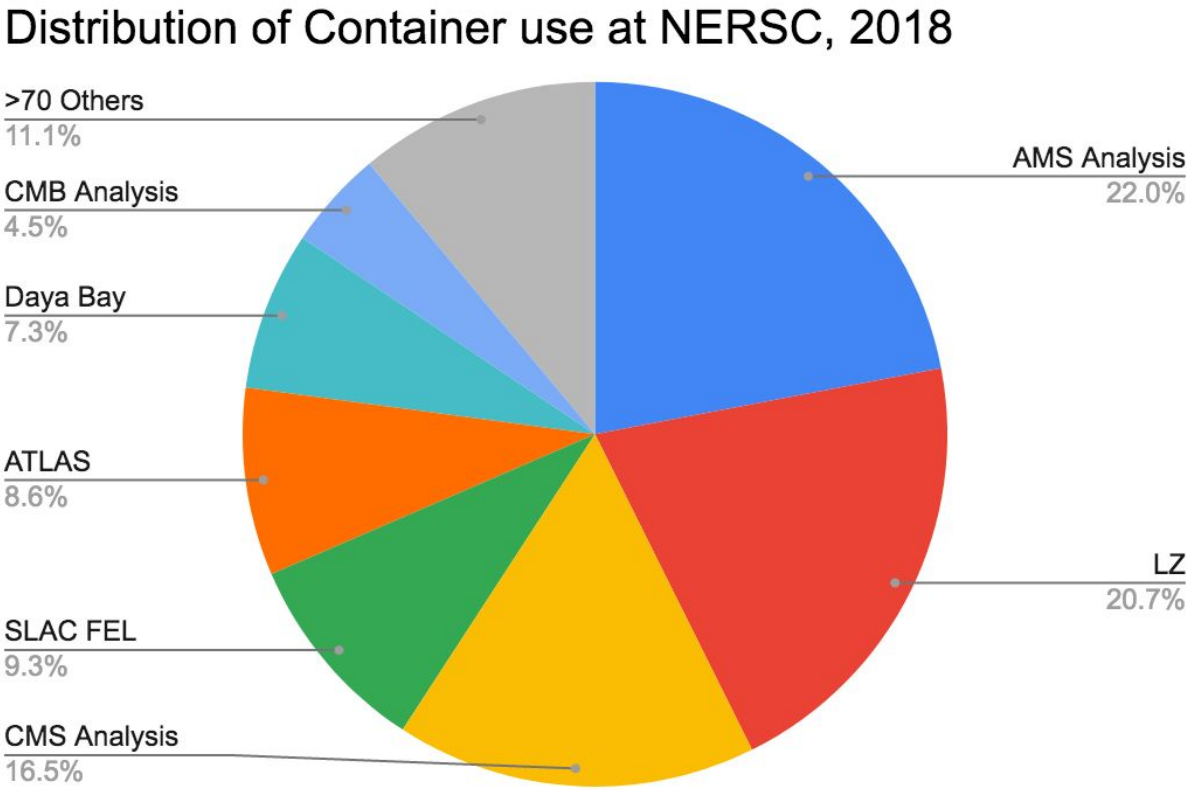
- Approximately 18 months to settle into new thread-use rates on Cori-KNL
  - a modest architecture shift from Edison or Cori-Haswell.
  - About 25% still MPI-only.
- Compare to 43% GPU enablement 8 years after Blue Waters installation.
- Porting to future accelerators may be even bigger lifts.

Surge in Python popularity; primarily for control flow.



- Python adoption is broad: over ¼ of projects and ¼ of jobs.
- Most python instances use limited compute resources.
  - Compare ¼ of jobs to 4.2% of node hours
  - 8x increase in node hours since 2014
- Two projects with a shared framework (PSI) account for 85% of Python node-hours.
- **P**lasma **S**urface **I**nteraction performs data management and workflow coordination for various compiled codes.

Container use is dominated by experimental analysis projects.



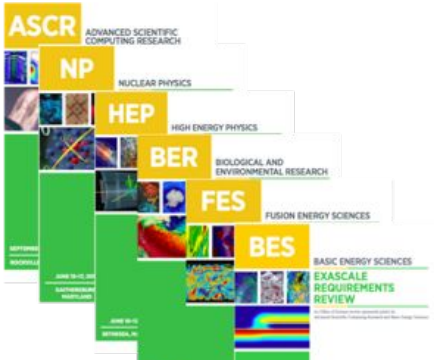
- Container use has increased dramatically:
  - 1% in 2014
  - 8% in 2018.

Machine Learning: a strategic development area

ML workload is expected to grow.

- In 2017, a small fraction of the NERSC workload:
  TensorFlow + Keras + Torch + SKlearn < 0.3%
- 4x increase in TensorFlow users between 2017 and 2018.
- NESAP for Perlmutter: ~90% of proposals expressed interest in ML.
- NERSC ML Survey: 120 of 168 respondents use or want to use NERSC for ML.

ML mentioned in exascale requirements reviews from all offices.



"Improved tools needed for machine learning and deep learning, which are now are a part of analysis (pattern recognition, anomaly detection)" (BES)

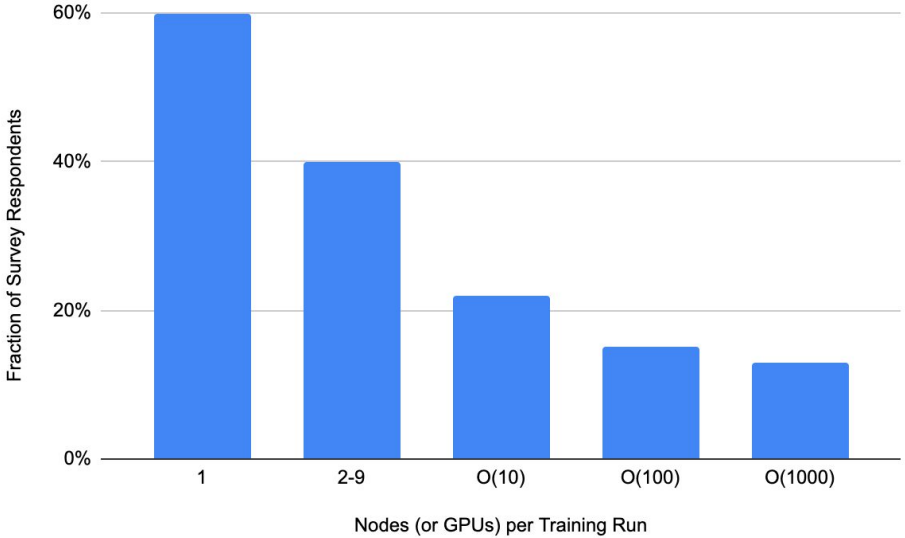
"Community would benefit from development of better algorithms (such as Machine Learning methods)... " (NP)

"New techniques for data analysis are urgently needed to address overwhelming data volumes and streams from both experiments and simulations" (HEP)

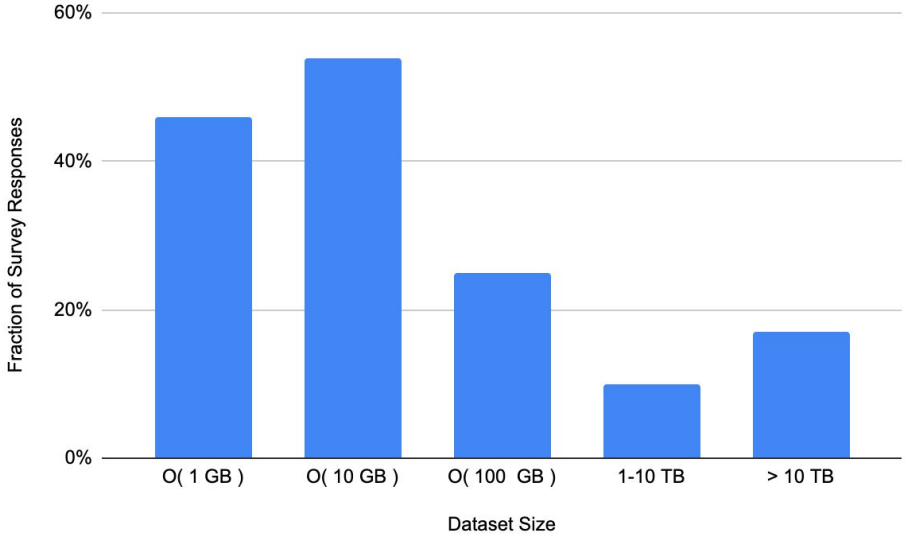
"New approaches to interpreting large data sets are needed and may include neural networks, image segmentation and other ML approaches." (BER)

HPC resources are needed to meet growing demands from Deep Learning.

- 40% of survey respondents need more than one node for training.



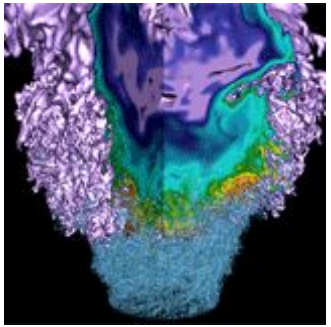
- Surveyed datasets are moderately large.

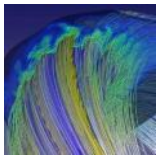


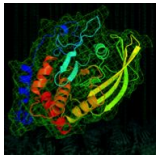
ML @ NERSC User Survey 2018
https://docs.google.com/presentation/d/1A8VGBhT4qZhKdByu5uTBQsklsR9rRIiWGEuaZudZB1Y/

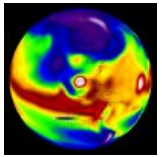
Multiple responses allowed; totals may exceed 100%.

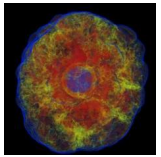
I/O & Storage

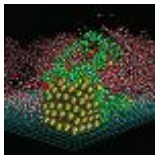






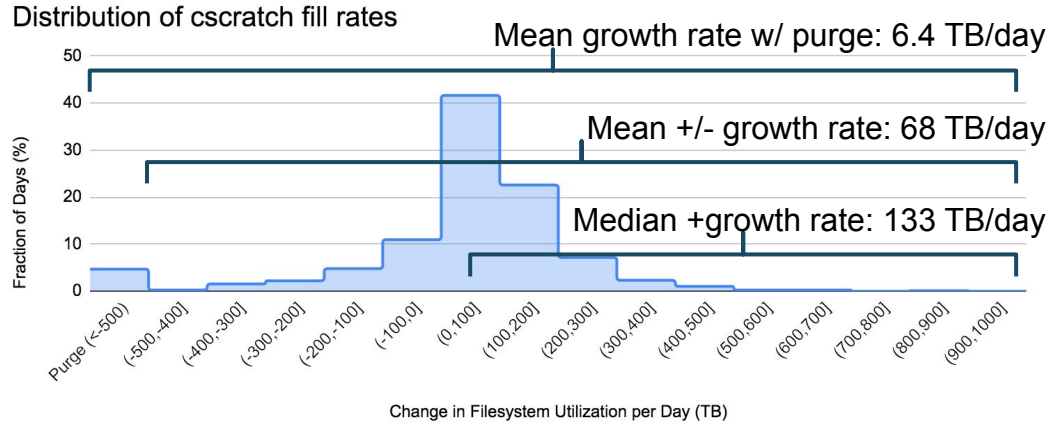




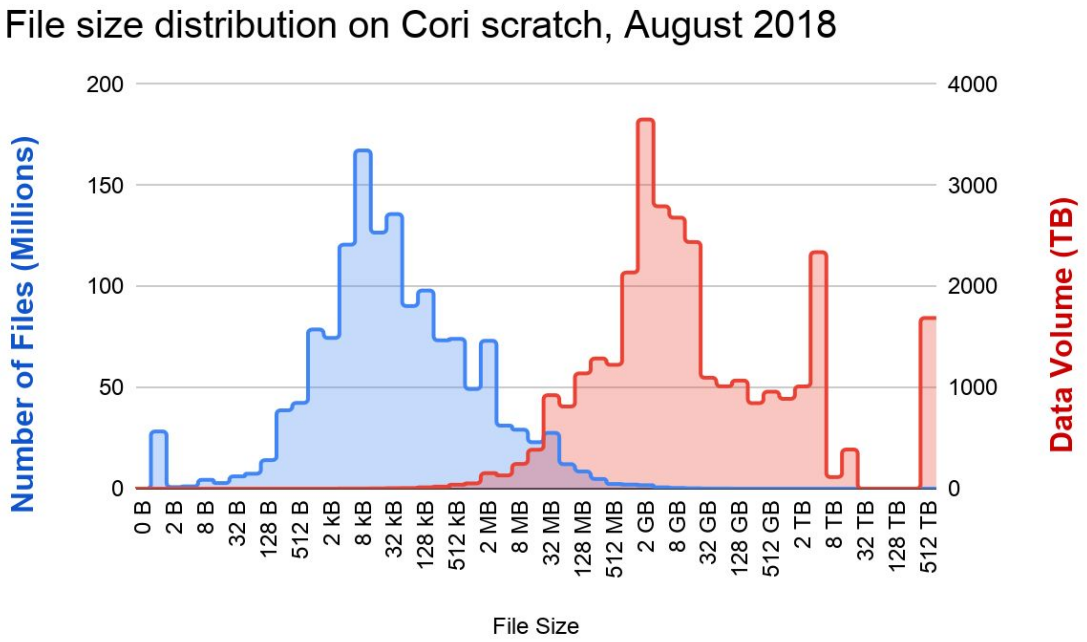


NERSC

Fast storage is precious: users would fill Cori's file system within 1-2 months



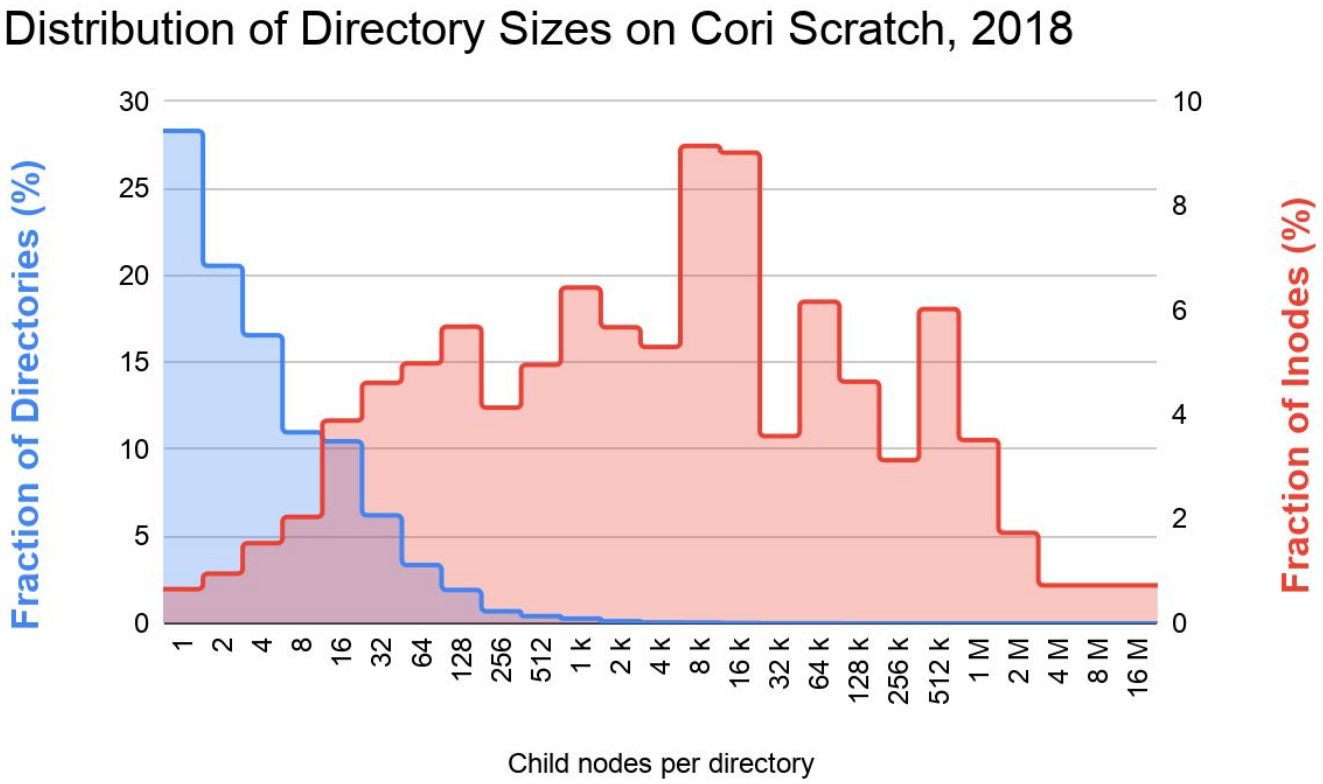
Files on Cori's scratch filesystem are generally small



- Most (85%) files are smaller than the 1 MB Lustre stripe size.
- Vast majority (~98%) of files are smaller than 32 MB.
- Most data (93%) is in files larger than 32 MB.
- Since 2014:
  - Average file size has grown 2.3x.
  - Maximum file size has grown 100x.

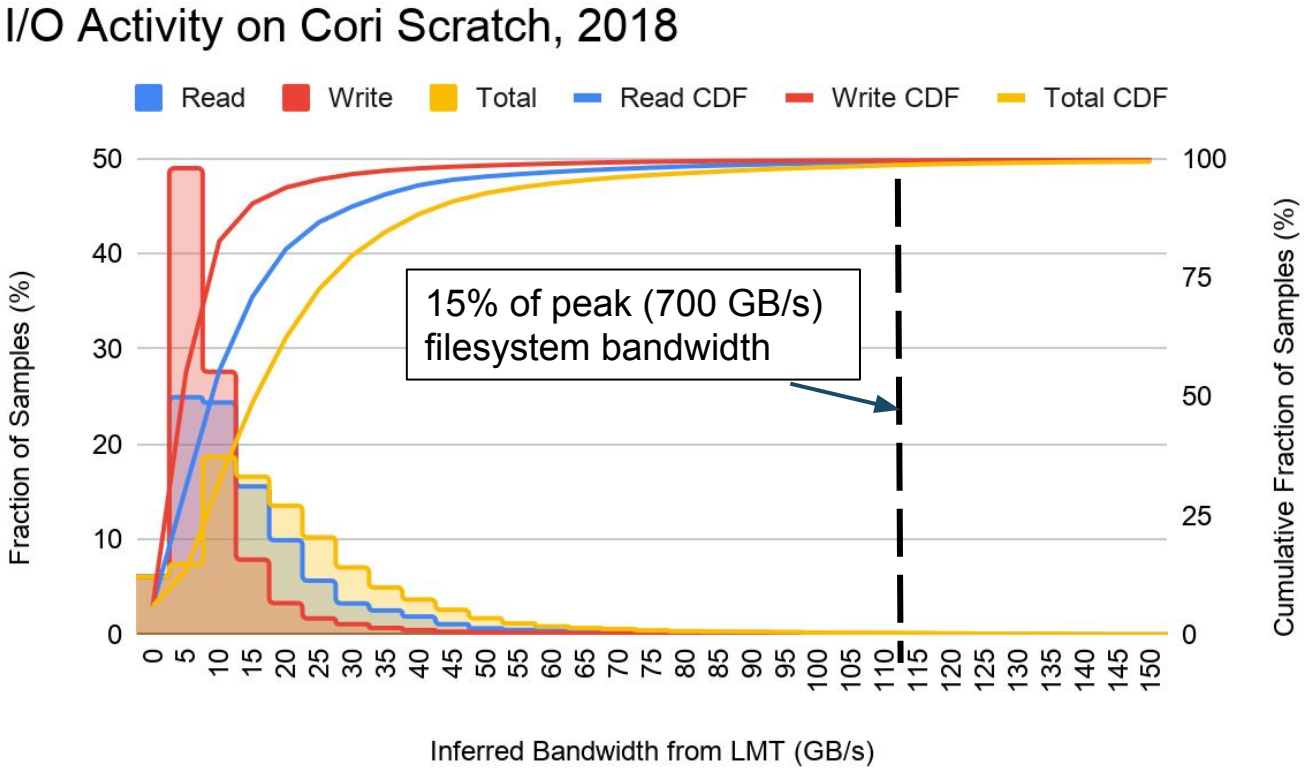
	Total Count	Total Volume	Average File Size	Max File Size
Cori Scratch (2018)	1.5 B	3156 TB	21.6 MB	512 TB
Edison Scratch2 (2014)	91 M	821 TB	9.4 MB	5 TB

Most directories are small, but most files are in large directories.



- ¾ of directories hold 8 files or fewer,
- But 95% of files are in larger directories.
- Half of all inodes are in directories with more than 4096 inodes.

Users seldom sustain large fractions of peak I/O bandwidth.



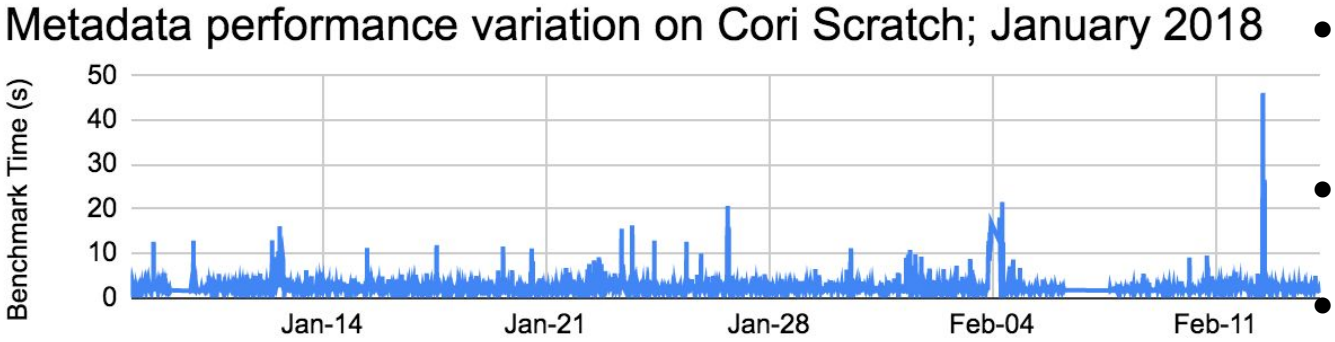
99% of scratch LMT samples used less than 15% of peak.

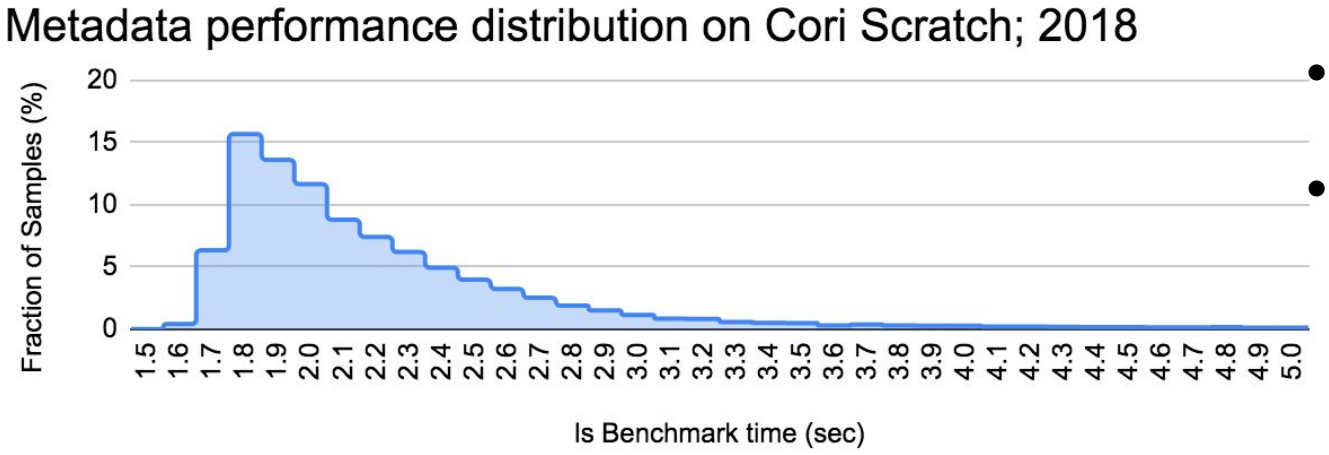
Significant fractions of peak I/O performance are routinely measured.

- 0.04% of LMT samples exceeded 80% of peak.
- Even poorly performing benchmark runs exceed the I/O rates observed in production.

Lustre Monitoring Tool (LMT) counts total data read/written within 5 second intervals. Actual I/O rates may exceed the inferred rates due to the large sampling window.

More reliable metadata performance would improve users' experiences.

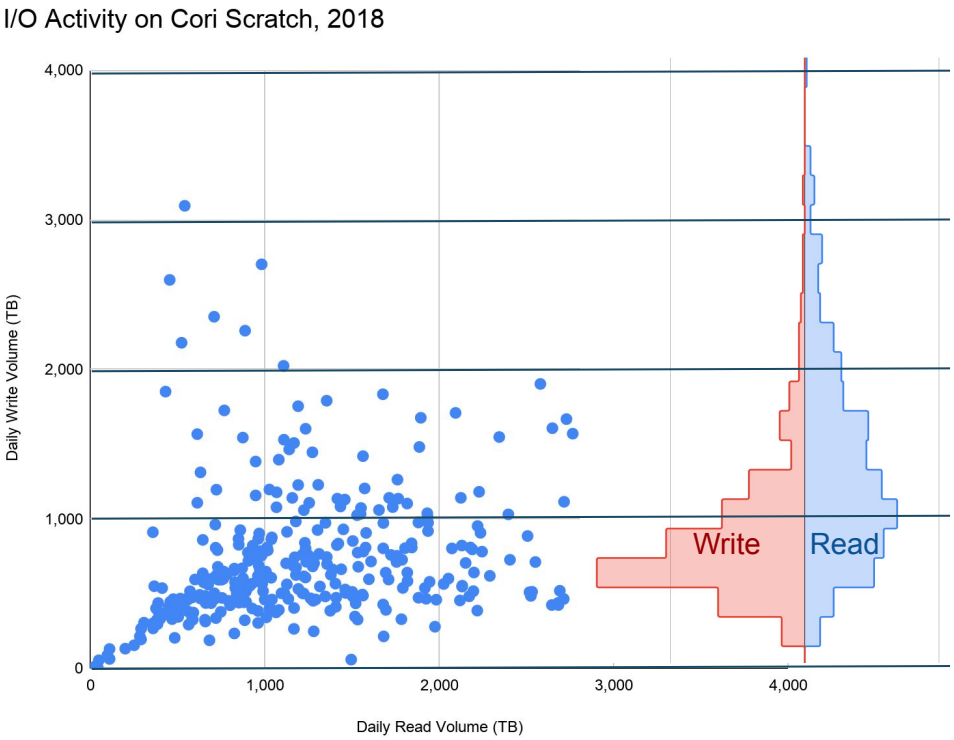




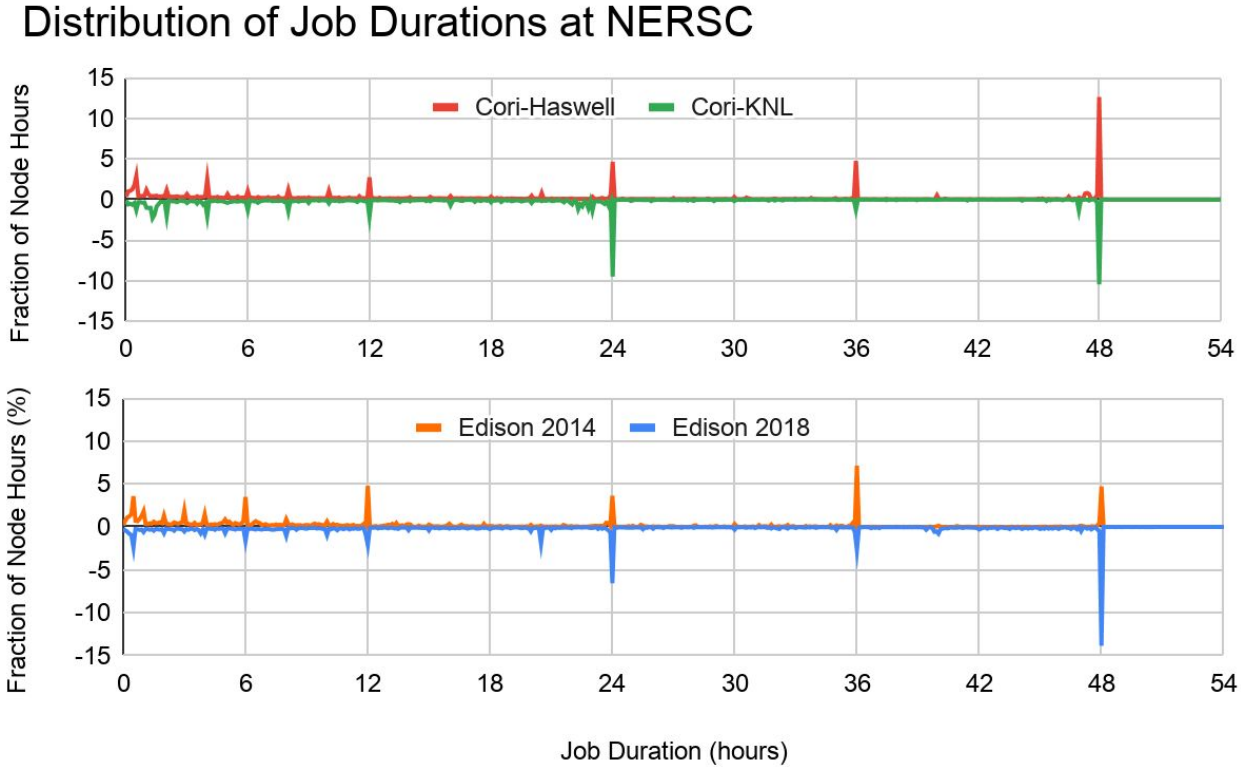
- Cron job measures "ls" response time to test I/O metadata performance.
- Measured once every 10 minutes.
- Good:
  Best = 1.6 seconds
  Median = 2.0 seconds.
- Bad: 4% of measurements take over 5.0 seconds.
- Ugly: A small fraction (0.1%) of tests take over 60 seconds. Not rare - once every 10 days.

Cori's I/O activity is *slightly* read-heavy. Some days are busier than others.

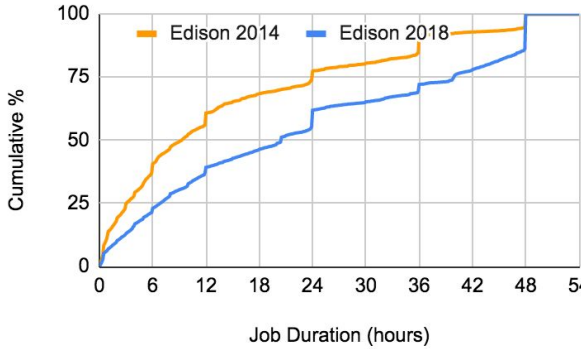
- On calm days, read and write volumes are similar and strongly correlated.
- On a typical (median) day, Cori will:
  - read its full 1.2 PB memory capacity
  - write half its memory
- Busy days are either read-heavy or write-heavy.
  - There are more read-heavy days than write-heavy days.
- On the busiest days, Cori's scratch filesystem will read or write 6x the memory capacity.



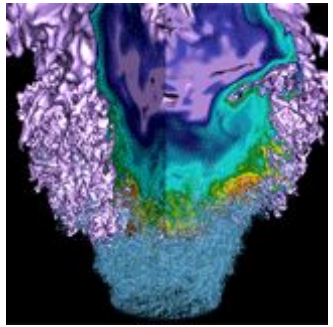
Much of the NERSC workload seems to use checkpoint restart functionality.

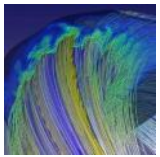


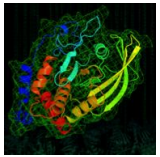
- 25% of node hours are used by jobs that reach wallclock limits.
- Users want longer-running jobs (and shorter wait times)
- Typical Edison job duration increased by 12 hours between 2014 & 2018.

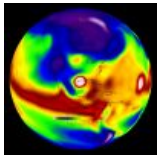


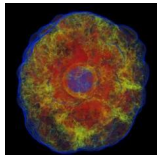
Wrapping up...

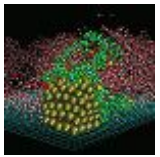












NERSC

Summary (1 of 2)

Application Demographics:

- Broad spectrum of scientific computation (many science domains, algorithms, projects, users)
- You can't please all the people all the time

Concurrency & Job Sizes:

- NERSC users run at every concurrency - capability and capacity.
- Largest jobs on Cori-KNL
- Threads are more heavily used for many-node jobs.

Languages, Libraries & Programming Environment

- MPI is ubiquitous
- OpenMP is the dominant shared-memory programming model
- CUDA is the dominant GPU programming model
- 50% of node hours likely to run on GPU's

Summary (2 of 2)

Memory - usage patterns are mixed, but generally increasing since 2014

- Memory per rank (sets a floor on memory per node)
  - Over half the workload uses less than 1 GB/rank
  - But 10% uses more than 25 GB/rank
- Half the workload does not experience significant memory pressure (per node)
  - But 15% uses over ¾ of the available memory per node
  - Users adapt to memory pressure by strong scaling.
- Jobs requiring large aggregate memory *do* exist and run on Cori-KNL

Workload Evolution

- **Machine Learning** is an area of increasing interest with growing computational demands.
  - 4x increase in TensorFlow users in one year.
  - 40% of ML survey respondents require multiple nodes for training.
- **Python** use has increased from 0.53% in 2014 to 4.2% in 2018.
  - Many projects use python a little bit.
  - Most growth from one framework PSI: a workflow coordinator for various compiled codes.
- **Container** use has increased from 1.1% in 2014 to 8% in 2018.
  - Various experimental data analysis projects account for the bulk of containers

Thank You

NERSC Top Codes & GPU Status, 2018

Rank	Code	% Node Hours	GPU Status
1	VASP	19.8	enabled
2	chroma	4.2	enabled
3	CPS	3.9	proxy
4	E3SM	3.6	unlikely
5	PSI - Python	3.6	enabled
6	xgc	3.6	enabled
7	ChomboCrunch	3.0	unknown
8	MILC	2.7	enabled
9	CESM	2.6	unlikely
10	HACC	2.6	enabled
11	K2PIPI	2.3	proxy
12	Compo_Analysis	2.0	unlikely
13	LAMMPS	2.0	kernels
14	cp2k	1.9	kernels
15	Espresso	1.8	enabled
16	ACME	1.5	unknown
17	Athena	1.3	proxy

Rank	Code	% Node Hours	GPU Status
18	NWCHEM	1.1	kernels
19	GYRO	1.1	proxy
20	phoenix	1.0	kernels
21	BerkeleyGW	1.0	enabled
22	GROMACS	0.9	enabled
23	AMS_Experiment	0.9	unlikely
24	osiris	0.9	proxy
25	S3D	0.8	enabled
26	Gene	0.8	proxy
27	Quark_Propagator	0.7	proxy
28	NAMD	0.7	enabled
29	toast	0.7	unknown
30	qchem	0.6	proxy
31	Pele	0.6	kernels
32	gtc	0.6	enabled
33	nplqcd	0.6	enabled
34	WRF	0.6	unlikely

Rank	Code	% Node Hours	GPU Status
35	disco_cEDM	0.6	proxy
36	SAURON	0.5	unknown
37	blast	0.5	enabled
38	sw4	0.5	kernels
39	gts	0.5	proxy
40	nimrod	0.5	unknown
41	DESC	0.4	unknown
42	aims	0.4	kernels
43	mini-em	0.4	enabled
44	Fornax	0.4	proxy
45	M3D	0.4	unknown
46	dg	0.3	unknown
47	cosmotools	0.3	unknown
48	b.sh	0.3	unknown
49	fastpm	0.3	unknown
50	Python - Other	0.6	unlikely
51	Other	17.1	unlikely