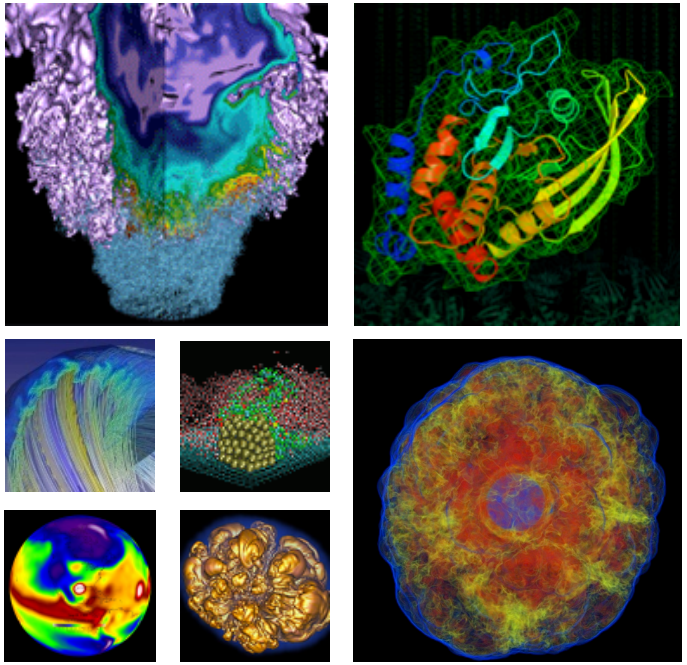
A quantitative approach to architecting all-flash Lustre file systems

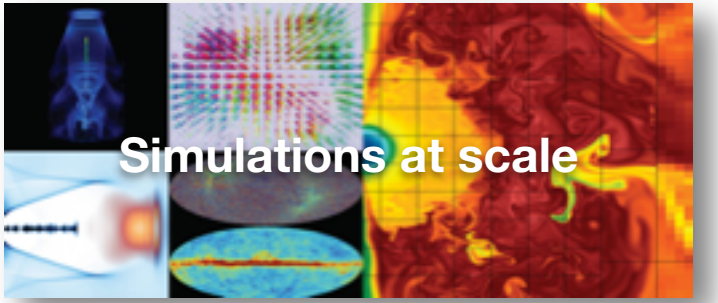


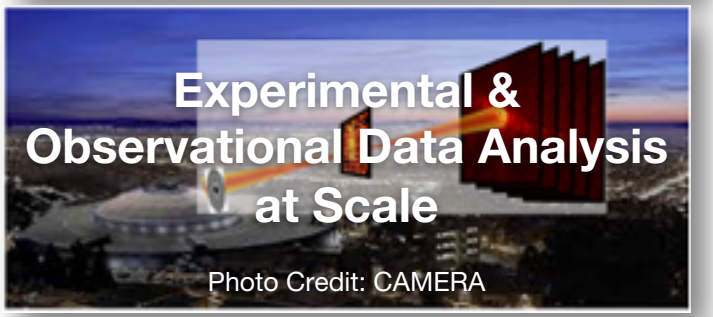
Glenn K. Lockwood
et al.

June 20, 2019

NERSC's mission and workload mix

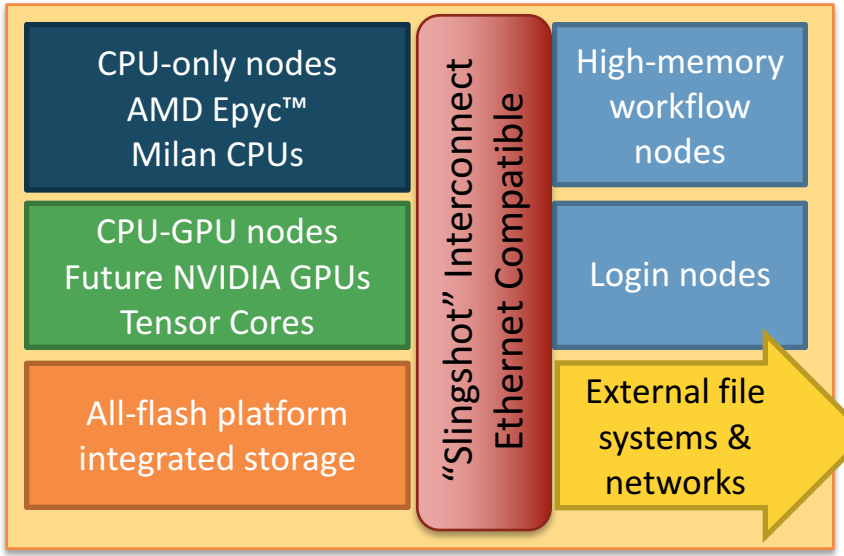
- **NERSC is the mission HPC computing center for the DOE Office of Science**
  - HPC and data systems for the broad Office of Science community
  - 7,000 Users, 870 Projects, 700 Codes
  - >2,000 publications per year
- **2015 Nobel prize in physics supported by NERSC systems and data archive**
- **Diverse workload type and size:**
  - Biology, Environment, Materials, Chemistry, Geophysics, Nuclear Physics, Fusion Energy, Plasma Physics, Computing Research
  - New experimental and AI-driven workloads





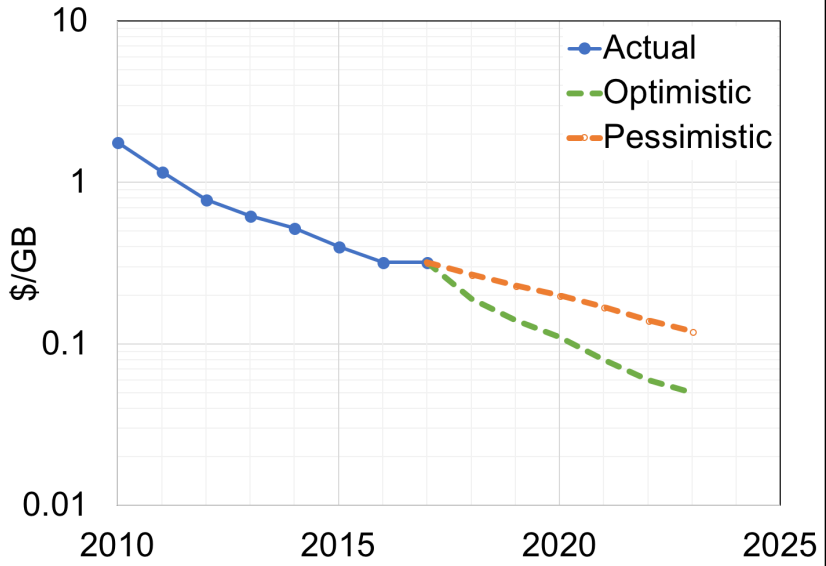
NERSC's 2020 System: Perlmutter

- **Designed for large-scale simulation *and* large-scale data analysis**
- **3x-4x capability of current system**
- **Include both NVIDIA GPU-accelerated and AMD CPU-only nodes**
- **200 Gb/s Cray Slingshot interconnect**
- **Single tier, all-flash Lustre file system**



Designing an all-flash parallel file system

- **In 2020, all-flash makes good sense for the performance tier**
- **With a fixed budget, how much should we spend on…**
  - OST capacity?
  - NVMe endurance?
  - MDT capacity for inodes and Lustre DOM?
- **Performance not discussed here**



NERSC's quantitative approach to design

- **Inputs:**
  1. **Workload data** from a reference system
  2. **Performance specs** of future compute subsystem
  3. **Operational policies** of future system
- **Outputs:**
  - Range of minimum required **OST capacity**
  - Range of minimum required **SSD endurance**
  - Range of minimum required **MDT capacity for DOM**

Reference system: NERSC Cori

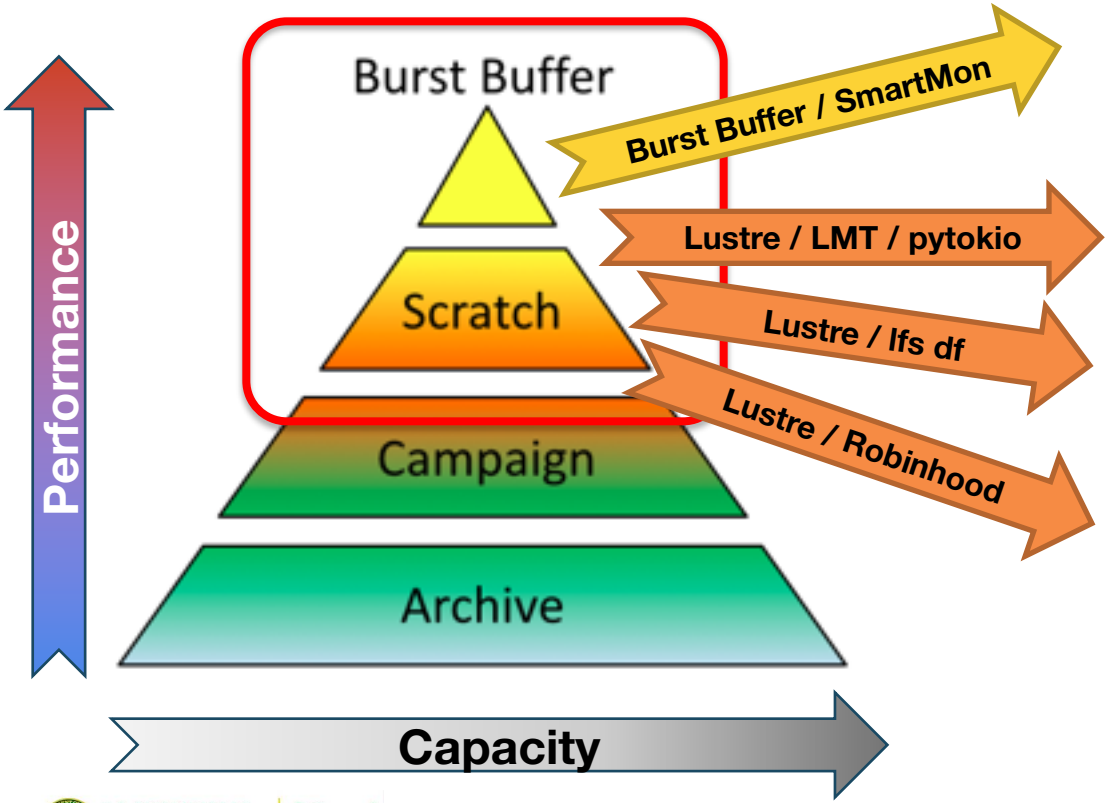
Compute

- **9,688 Intel KNL nodes**
- **2,388 Intel Haswell nodes**

Storage

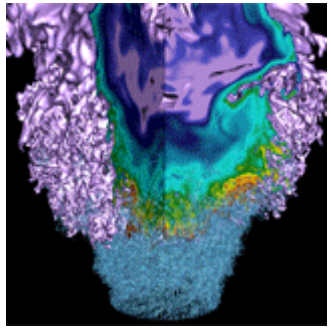
- **30 PB, 700 GB/s scratch**
  - Lustre (Cray ClusterStor)
  - 248 OSSes x 41 HDDs x 4 TB
  - 8+2 RAID6 declustered parity
- **1.8 PB, 1.5 TB/s burst buffer**
  - Cray DataWarp
  - 288 BBNs x4 SSDs x 1.6 TB
  - RAID0

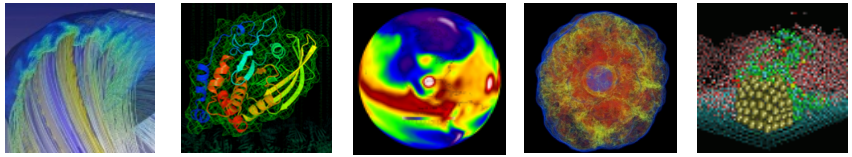
NERSC workload data sources



- **smartmon tools**
  - Per-device metrics vs time
  - Write amplification factor
  - Total bytes written
- **Lustre Monitoring Tools (LMT)**
  - Per-OST measurements vs time
  - Bytes written to OSTs
- **Lustre "lfs df“ & cron**
  - Per-OST fullness vs time
  - LMT also has this information
- **Robinhood**
  - Snapshot of entire namespace & POSIX metadata
  - File size distribution
  - Non-file inode sizes

Minimum required file system capacity

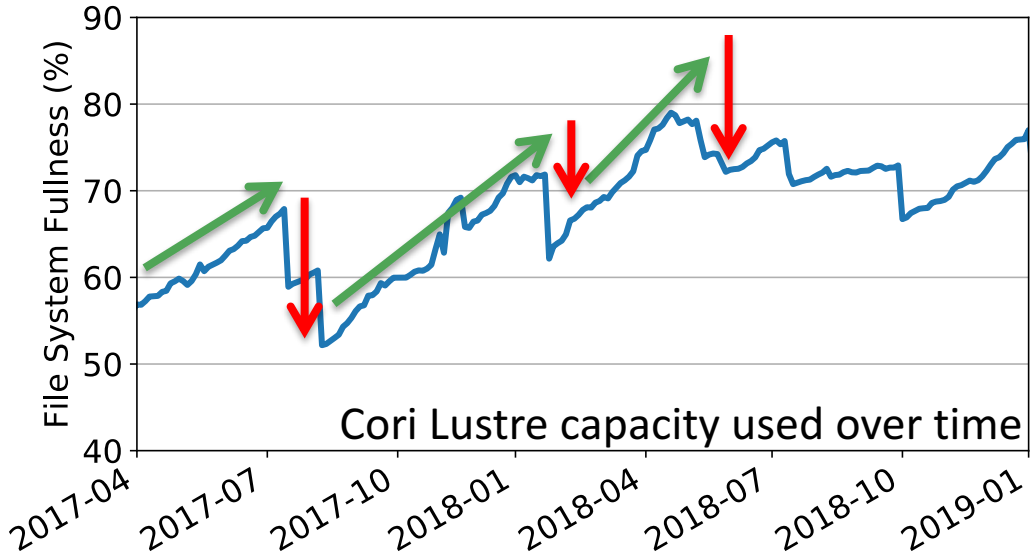




How much flash capacity do we need?

Intuitively, we need to balance…

1. How quickly **users fill** the file system
2. How frequently the **facility drains** the file system



File system capacity model

"time between purge cycles" or
"time after which files are eligible for purge"

Daily file system growth of Cori

Minimum capacity of Perlmutter scratch

$$C^{\text{new}} = \text{SSI} \cdot \left(\frac{\lambda_{\text{purge}}}{\text{PF}}\right) \cdot \left(\frac{\partial C^{\text{ref}}}{\partial t}\right)$$

Sustained System Improvement

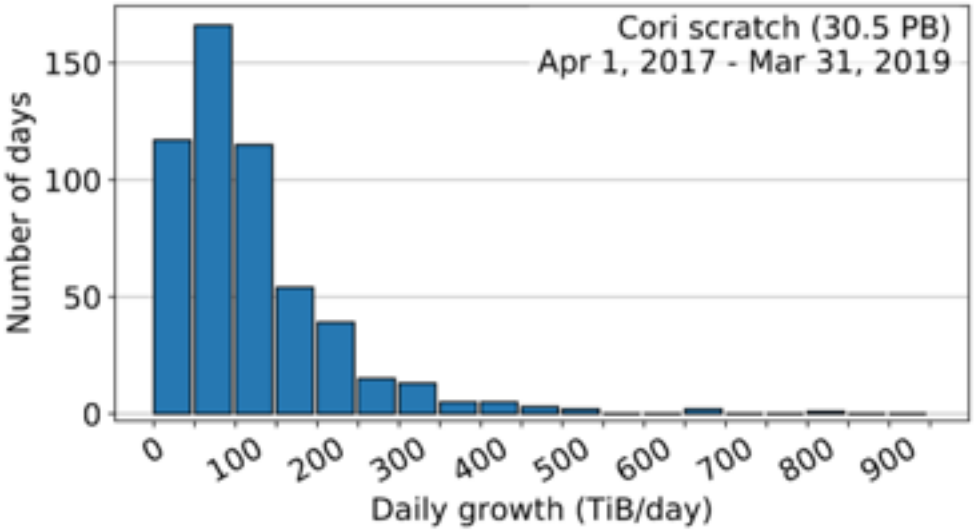
Purge Fraction – fraction of fs capacity to reclaim

Calculating C^{new} for Perlmutter

$\left(\frac{\partial C^{ref}}{\partial t}\right)$ Mean daily growth for Cori is 133 TB/day

$\left(\frac{\lambda_{purge}}{PF}\right)$ Data retention policy for Perlmutter:

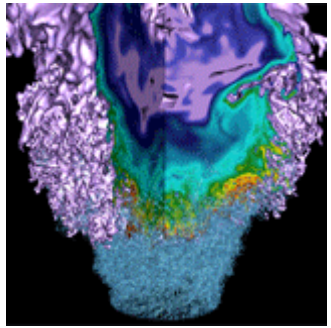
- Purge/migrate after 28 days
- Remove/migrate 50% of total capacity

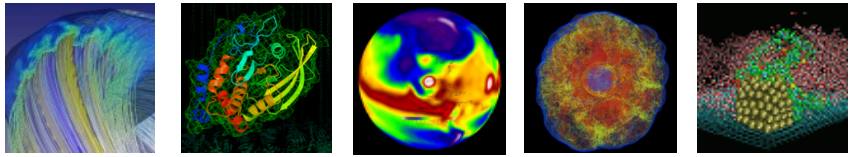


SSI Perlmutter will be 3x to 4x "faster" than reference system

C^{new} **Minimum capacity is between 22 PB and 30 PB**

Do we need [to pay for] high-endurance SSDs?





How many drive writes per day do we need?

Intuitively, required drive writes per day depends on…

1. **How much data users write to the file system daily**
2. **How much extra RAID parity is written**
3. **How much hidden “stuff” is written**
    1. Read-modify-write write amplification
    2. Superblock updates
4. **How big a "drive write" actually is**

Calculating drive endurance requirements

Drive Writes Per Day required for Perlmutter

File System Writes Per Day on Cori

Total drive counts

RAID code rates

$$\mathrm{DWPD}^{\mathrm{new}} = \mathrm{SSI} \cdot \mathrm{FSWPD}^{\mathrm{ref}} \cdot \mathrm{WAF} \cdot \left(\frac{1}{\chi}\right) \cdot \left(\frac{N^{\mathrm{ref}}}{N^{\mathrm{new}}}\right) \left(\frac{c^{\mathrm{ref}}}{c^{\mathrm{new}}}\right) \left(\frac{R^{\mathrm{ref}}}{R^{\mathrm{new}}}\right)$$

Sustained System Improvement

Write Amplification Factor caused by other "stuff"

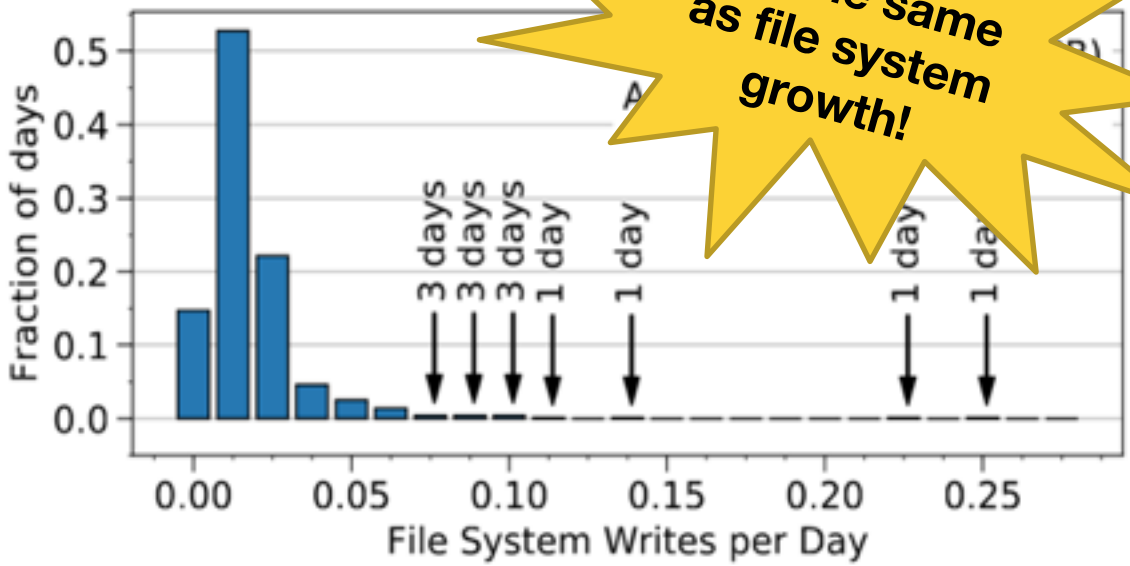
Lustre formatting efficiency

Per-drive capacities

Cori's FSWPD over two years

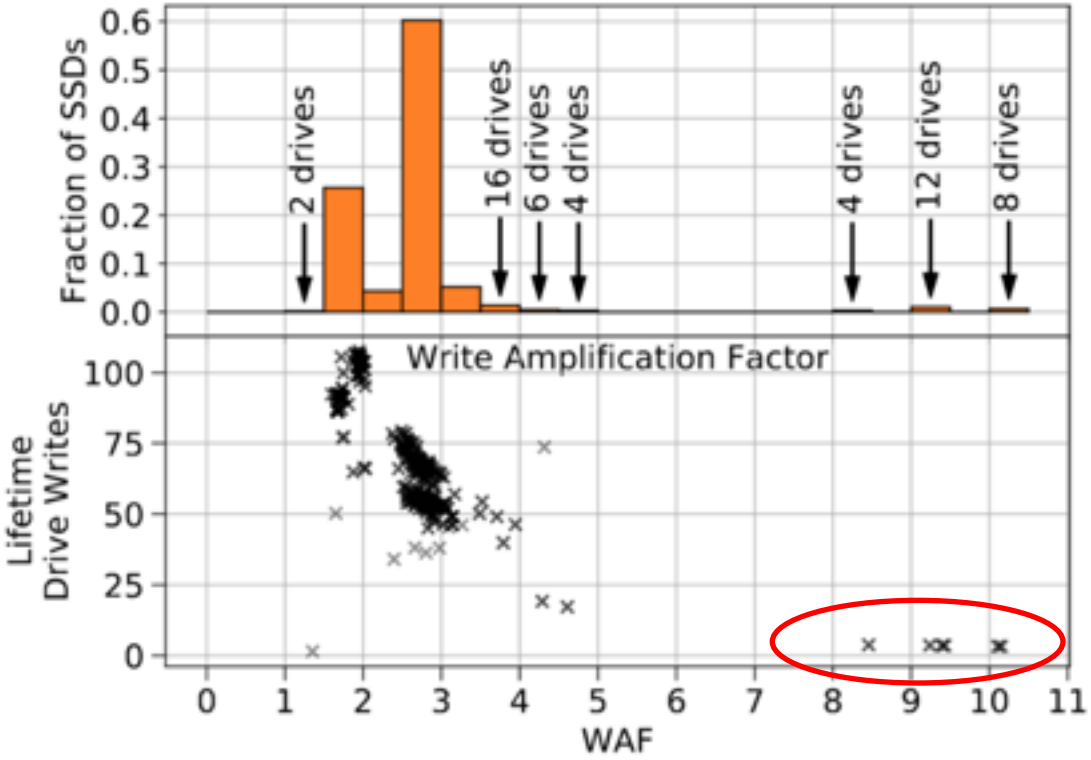
From LMT/pytokio

- 1 FSWPD = 30.5 PB
- Mean FSWPD 0.024 (~700 TB/day)

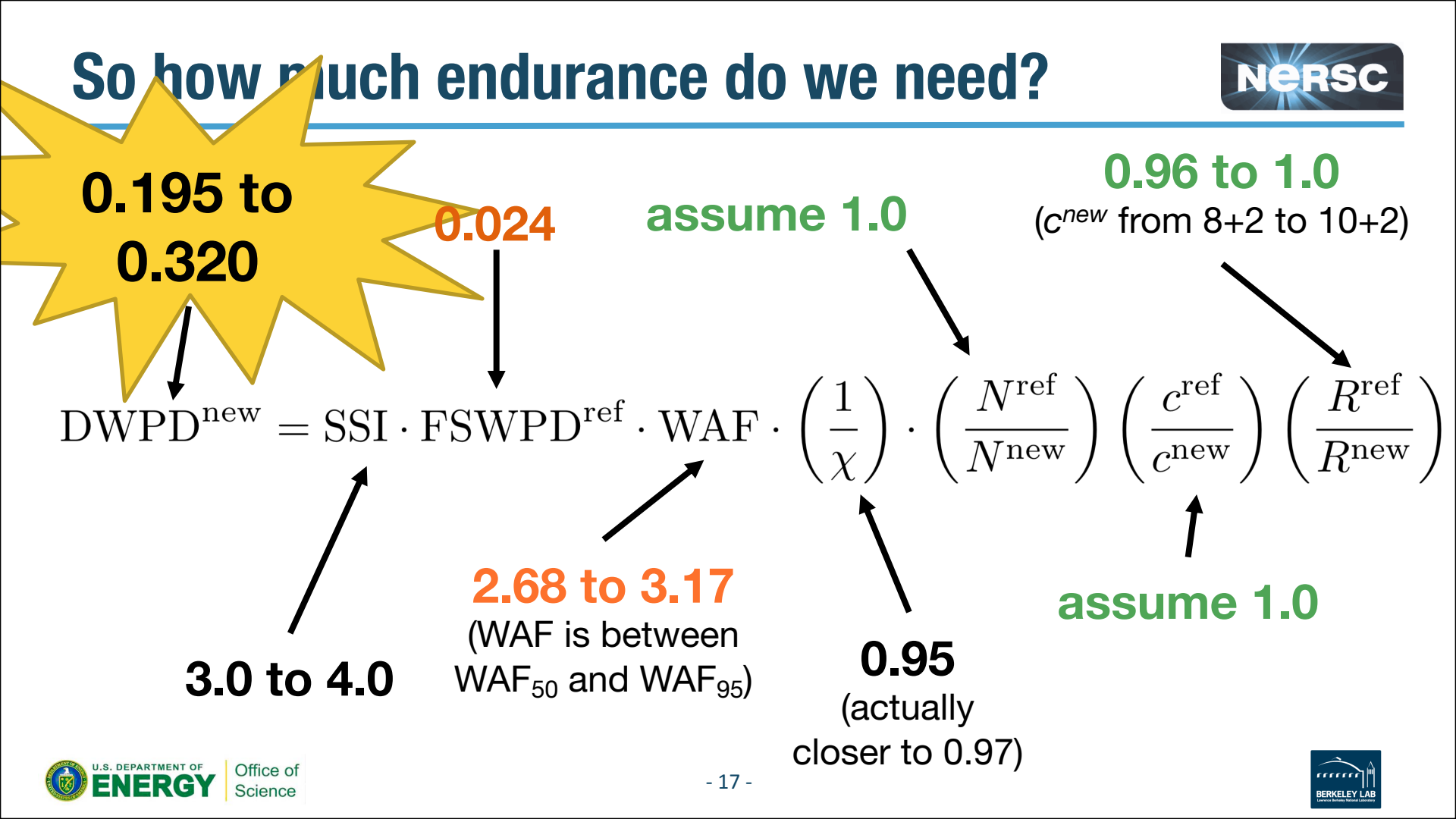


Estimating WAF in production

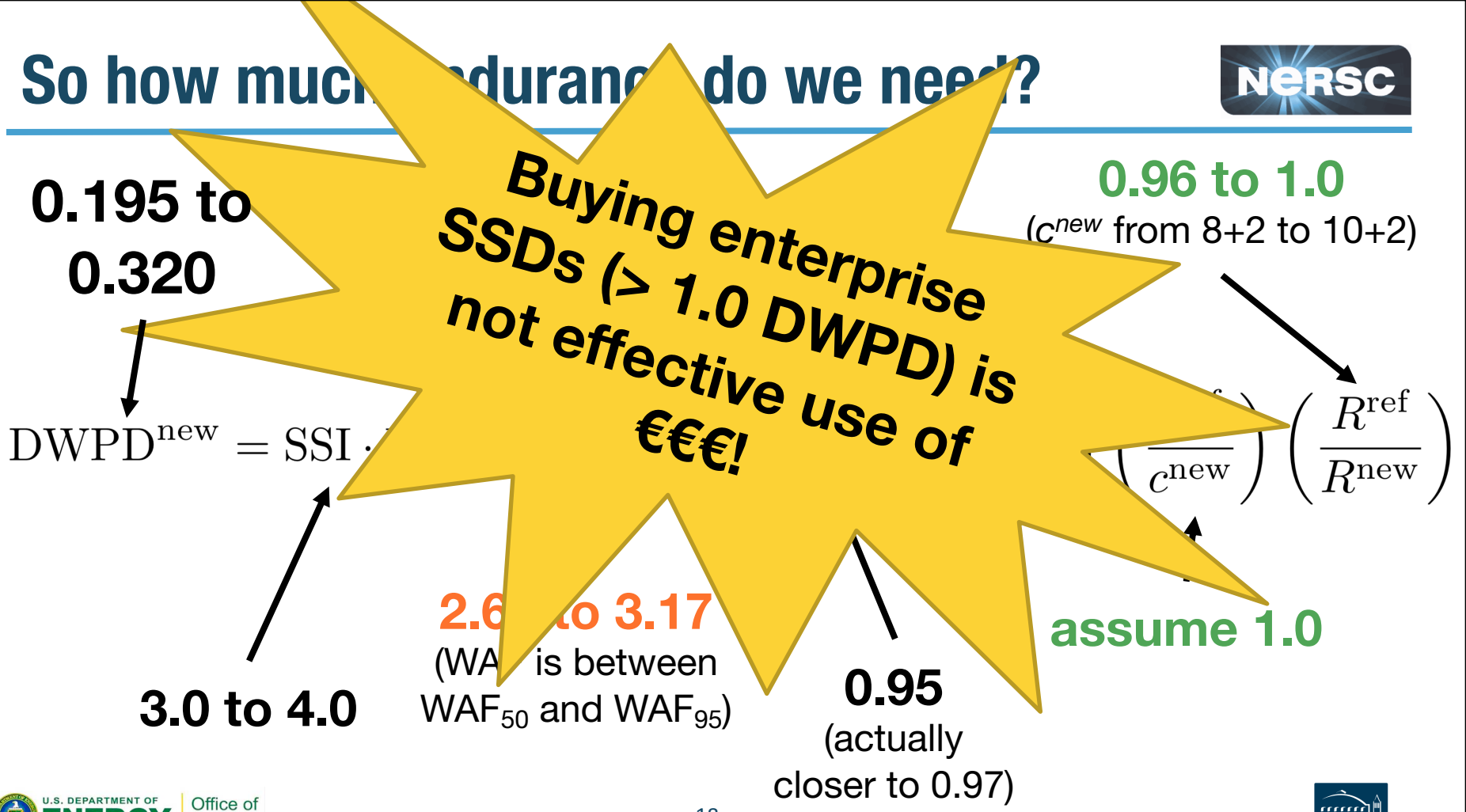
- **Not internal SSD WAF**
- **Difference between what Lustre writes and what disks write**
  - Can use OST-level writes (LMT) and SMART data
  - Had to use DataWarp SSD WAFs here
- **Median WAF: 2.68**
- **95th percentile: 3.17**



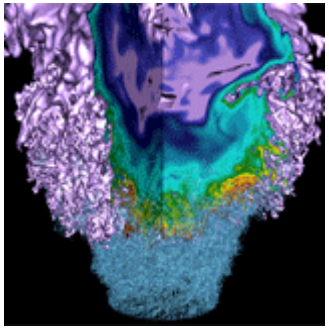
So how much endurance do we need?

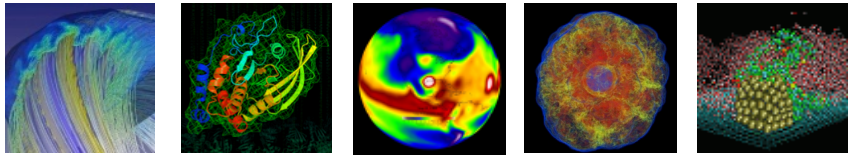


So how much endurance do we need?



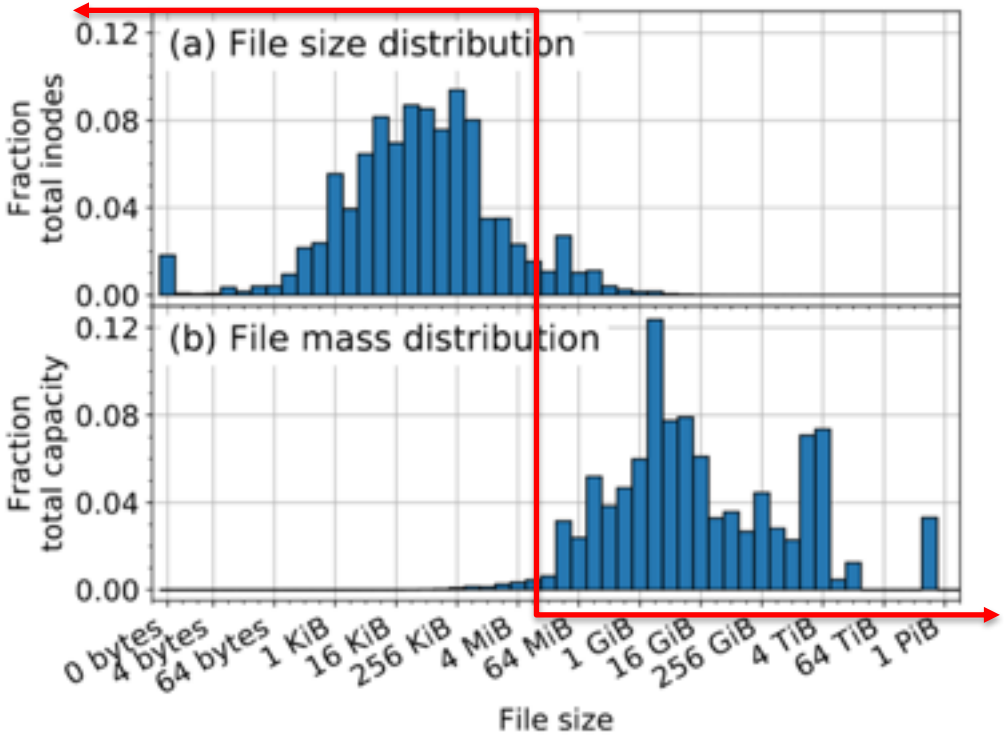
How much MDT capacity is needed for DOM?





Lustre Data-on-MDT (DOM)

- **Store first S_0 bytes of a file directly on Lustre MDT**
- **Reduces number of RPCs, small-file interference, etc**
- **But HPC facilities now have to define:**
  - How big should S_0 be?
  - How much MDT capacity is required to store all files' DOM components?
  - How much MDT capacity is required to store inode data?



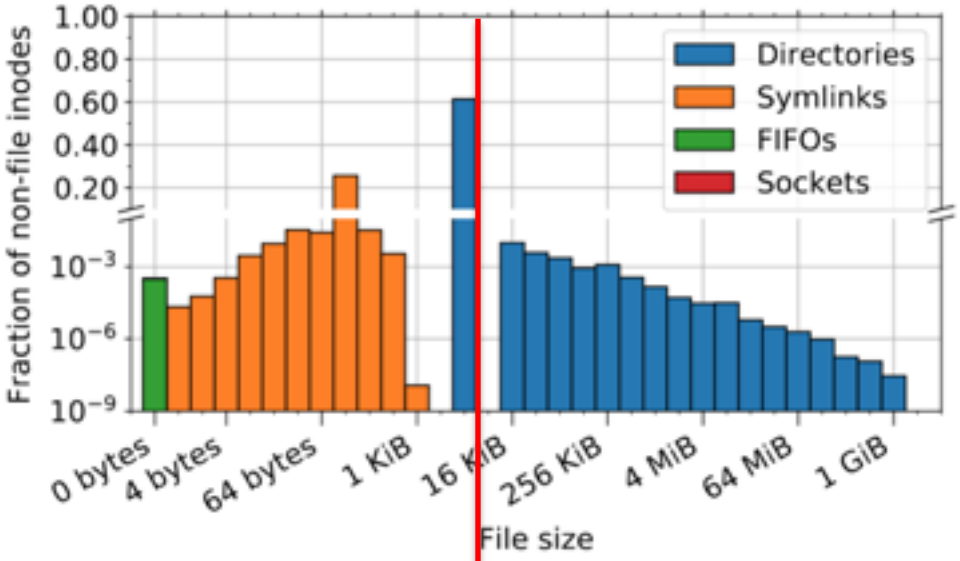
How much MDT capacity do we need?

Intuitively, MDT capacity depends on…

1. **How much capacity we need to store inodes (files, directories, symlinks, …)**
2. **How many files we have with size smaller than S_0**
3. **How many files we have with size larger than S_0**
4. **Our choice of S_0**

MDT capacity required for inodes

- 1 inode = 4 KiB
- ...but big dirs need more MDT capacity!
- Assume size dist is constant for any C^{new}
- Then estimate the size dist for our C^{new}



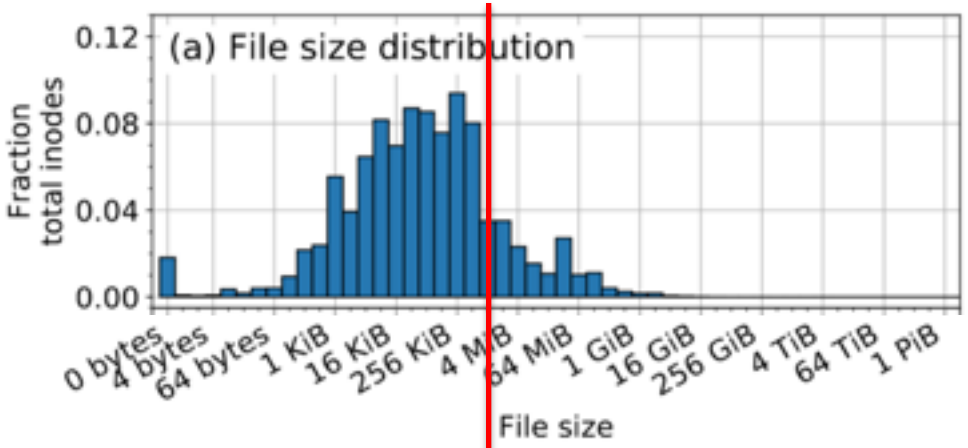
For inodes <= 4 KiB:

inodes * 4 KiB

For dirs > 4 KiB:

Just add up size of all directories

MDT capacity required for DOM



For files <= S_0:

Just add up size of all files

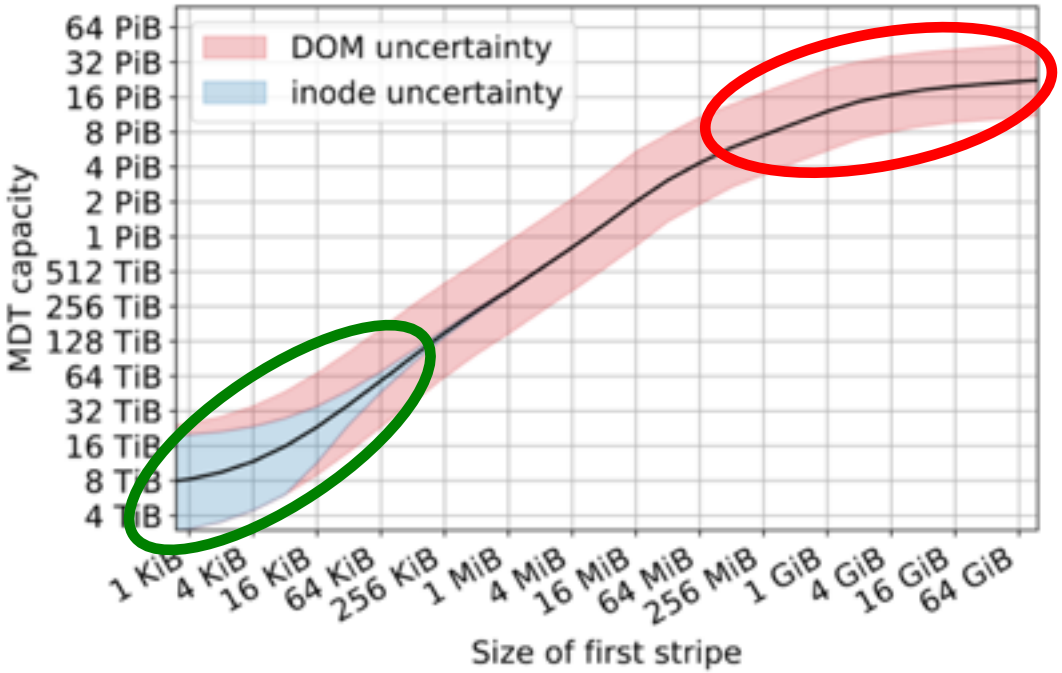
For files > S_0:

$S_0 \times$ (# files < S_0)

S_0

Total MDT capacity required vs. S_0

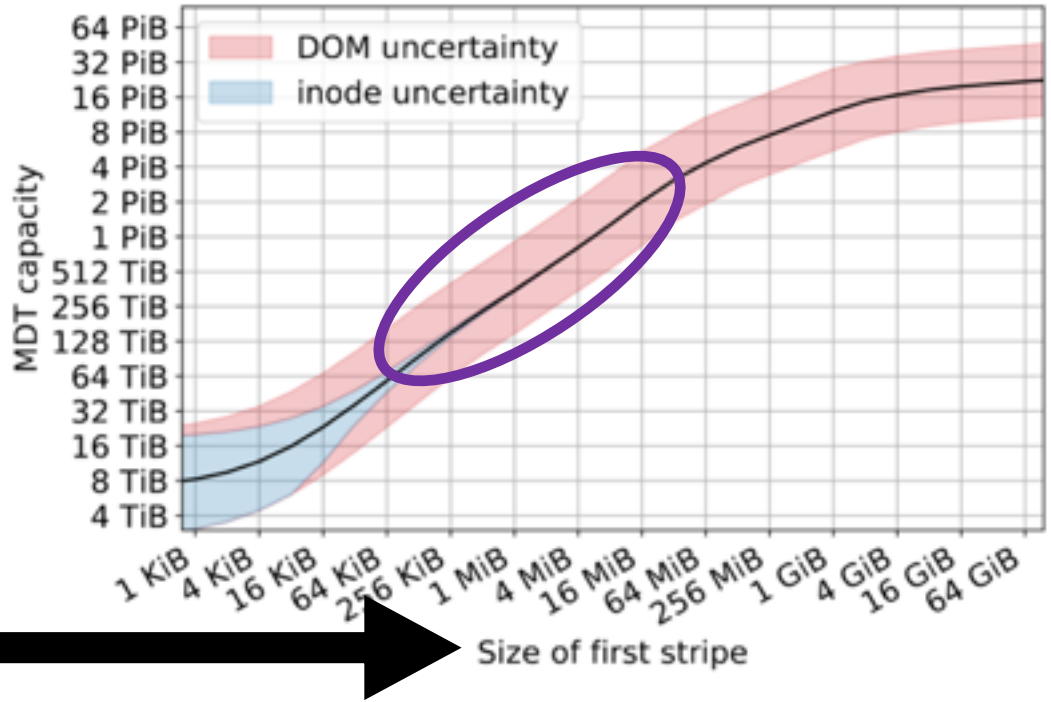
- **Uncertainty from scaling distributions between Cori and Perlmutter**
- **Extreme values of S_0 uninteresting**
  - Tiny S_0 = nothing fits in DOM
  - Huge S_0 = everything fits in MDT (and I/O is no longer parallel)



Cost-performance tradeoff of DOM

Cost—MDTs are usually RAID10, not RAID6

(R=0.5 vs. R=0.8)



Performance—better small-file IOPS and less large-file interference

Conclusions

- **We have defined models to quantify relationships between**
  - Workload (I/O and growth rates, file size distributions)
  - Policies (purge/data migration policy)
  - File system design parameters
    - File system capacity
    - SSD endurance
    - MDT capacity and DOM configuration
- **Imperfect models, but still**
  - identify key workload parameters
  - bound on design requirements based on facts, not instincts
  - serve as a starting point for sensible system design

All code and workload data available online!*

https://doi.org/10.5281/zenodo.3244453

Thank you!

(and we're hiring!)