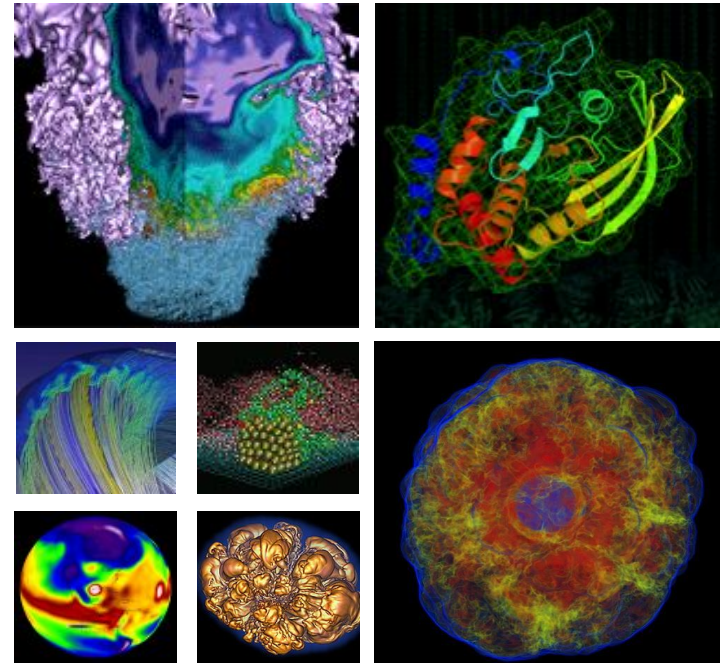


A Quantitative Approach to Architecting All-Flash Lustre File Systems



Kirill Lozinskiy

Glenn K. Lockwood et al.

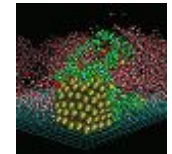
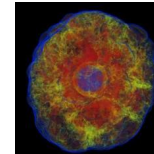
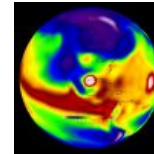
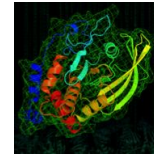
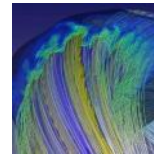
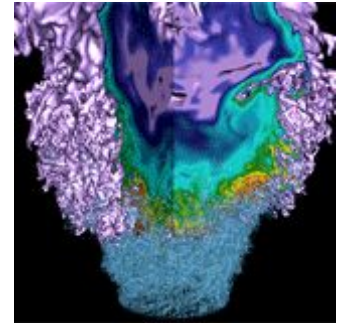
LUG 2019 / May 15, 2019

Agenda

- **Introduction and Methods**
 - NERSC Overview
 - Reference System (Cori)
 - New System (Perlmutter)
- **File System Capacity**
- **Drive Endurance**
 - Parity and Write Amplification
 - Anticipated Write Load
 - Endurance Requirements
- **Metadata Configuration**
 - MDT Capacity Required by DOM
 - MDT Capacity Required for Inodes
 - Overall MDT Capacity
- **Conclusion**



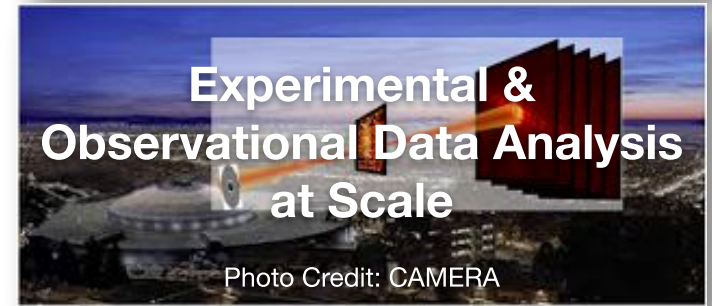
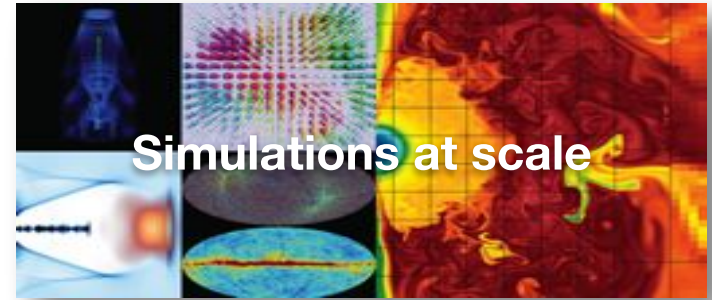
NERSC + Systems Overview



NERSC @ Berkeley Lab (LBNL)



- 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-8 aka Cori (Cray XC-40)



Compute

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

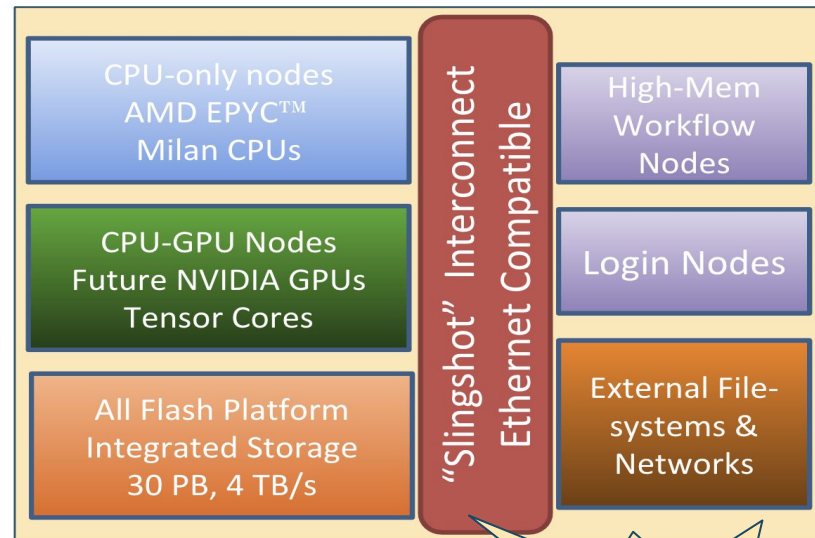
Storage

- 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-9 aka Perlmutter (Cray)



- Designed for both large scale simulation and data analysis from experimental facilities
- Overall 3x to 4x capability of Cori
- Includes both NVIDIA GPU-accelerated and AMD CPU-only nodes
- Slingshot Interconnect
- Single Tier, All-Flash Lustre scratch filesystem



Delivery in late-2020



Multiple Storage Tiers



Lustre “scratch” and Burst Buffer

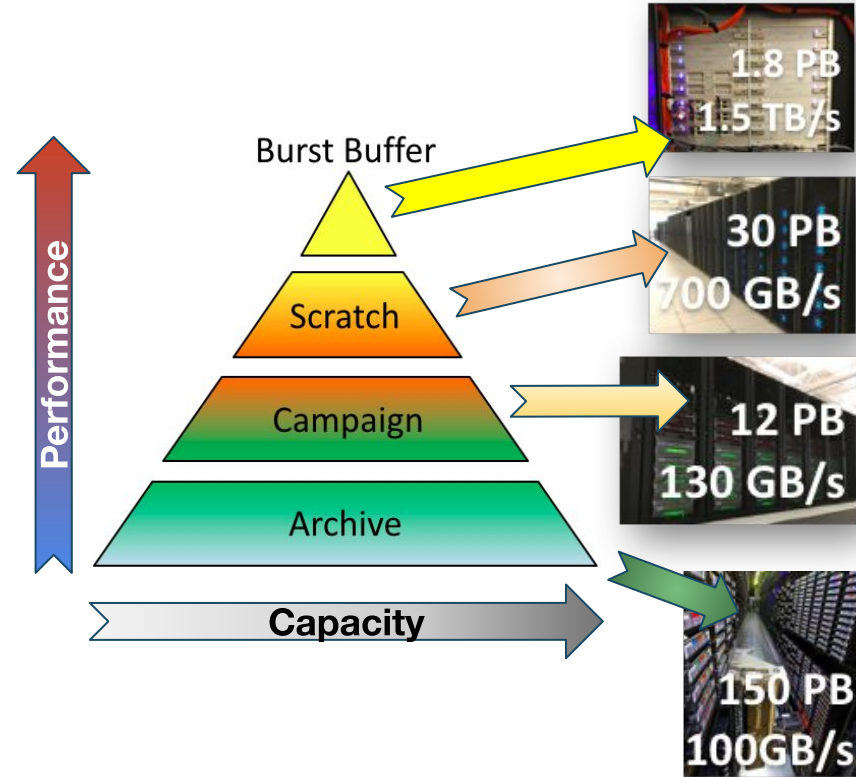
- Ephemeral storage, data purged if not accessed, user-based quotas and permissions
- Intended for high speed access to active data used for running computations

Spectrum Scale “project” file system

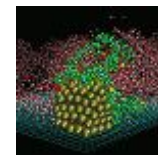
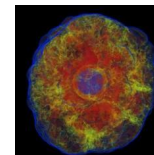
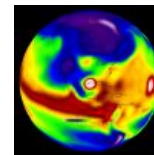
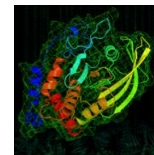
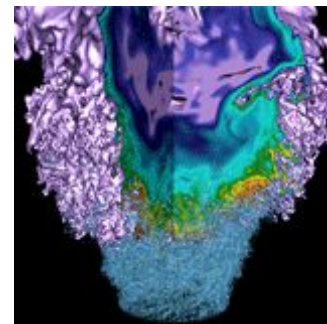
- Medium term storage, data never purged, group quota and permissions
- Intended for shared data needed by entire science group, will be used for computing in the near future

HPSS Tape Archive

- Long term storage, data never purged, user-based and group quotas and permissions
- Permanent archival of scientific data



Introduction + Methods



Statement: Today, it *is* economically possible to deploy enough flash capacity to replace the scratch tier and burst buffer tier

Motivating questions:

- How much capacity is enough capacity for a scratch file system?
- What should the purge policy be to manage this capacity?
- Will the SSDs wear out too quickly?
- What drive endurance rating is required?

- Quantitative approach to design the 30 PB all-flash Lustre file system
- Integrated analysis of *current* workloads and *projections* of future performance and throughput
- We were able to constrain many critical design space parameters and quantitatively demonstrate that Perlmutter will deliver:
 - Optimal performance
 - Effectively balance cost
 - Effectively balance capacity
 - Endurance
 - Modern features of Lustre

NERSC-9's All-Flash Architecture

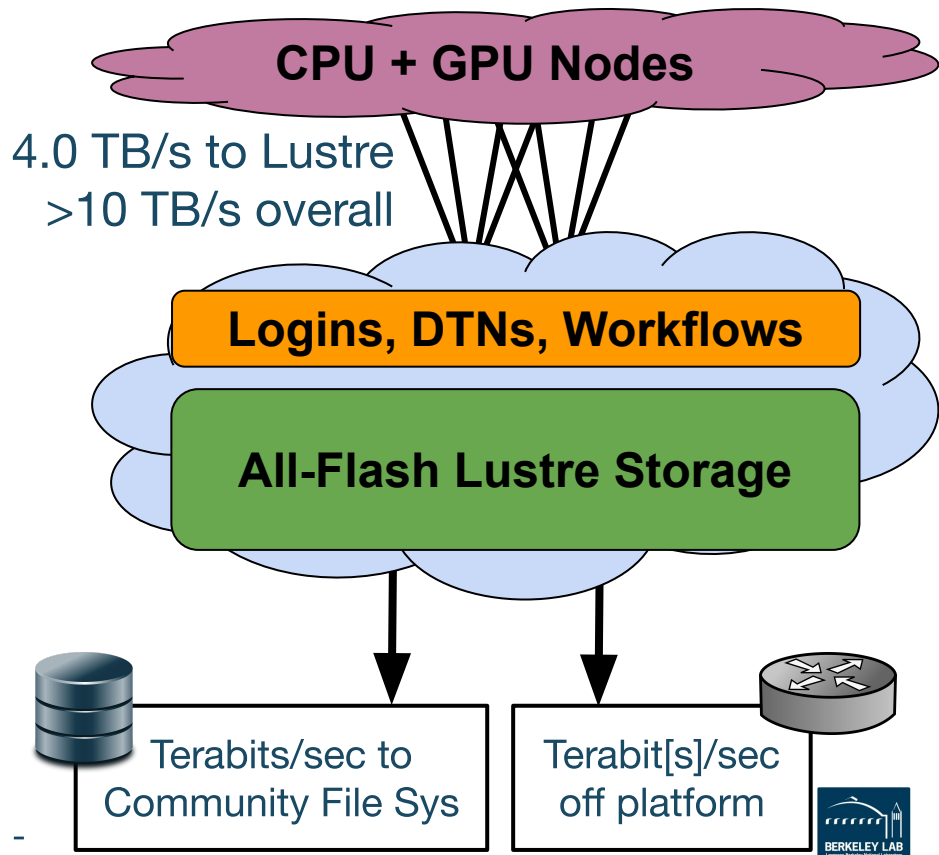


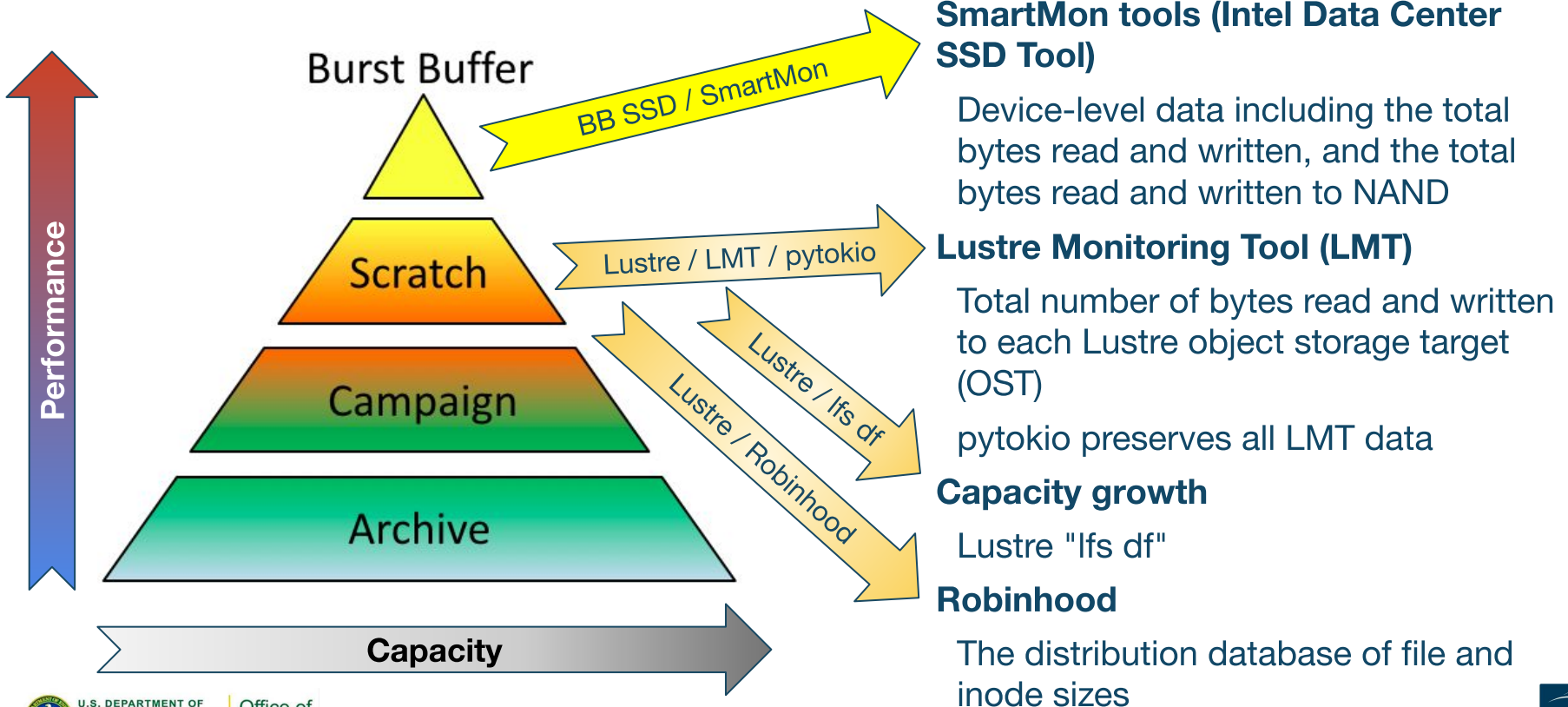
Fast across many dimensions

- 30 PB usable capacity
- ≥ 4 TB/s sustained bandwidth
- $\geq 7,000,000$ IOPS
- $\geq 3,200,000$ file creates/sec

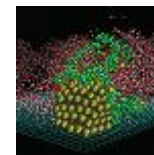
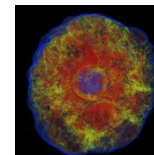
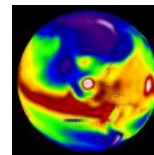
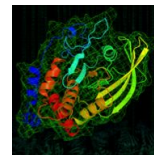
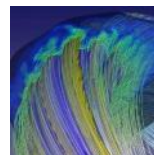
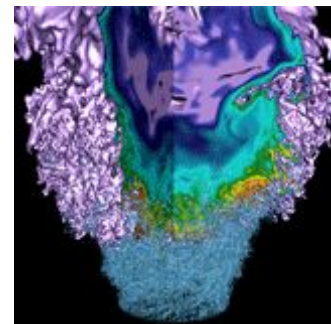
Integrated network, separate groups

- Storage/logins remain up when compute is down
- No LNET routers between compute and storage





File System Capacity



File System Capacity



Data management policy

A measure of time between purge cycles
or time after which files are eligible for purging

Reference
system capacity
change

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
3x - 4x output capacity over Cori

Desired capacity to
be reclaimed

Change in time

File System Capacity



$\left(\frac{\partial C^{\text{ref}}}{\partial t}\right)$ Mean daily growth projected for Perlmutter at 133 TB/day

$\left(\frac{\lambda_{\text{purge}}}{\text{PF}}\right)$ Data retention policy for Perlmutter is $t_{\text{time}} > 28$ days

- OK to purge after that time
- Each purge aims to remove or migrate 50% of the total capacity

SSI Anticipated 3x to 4x sustained system improvement

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

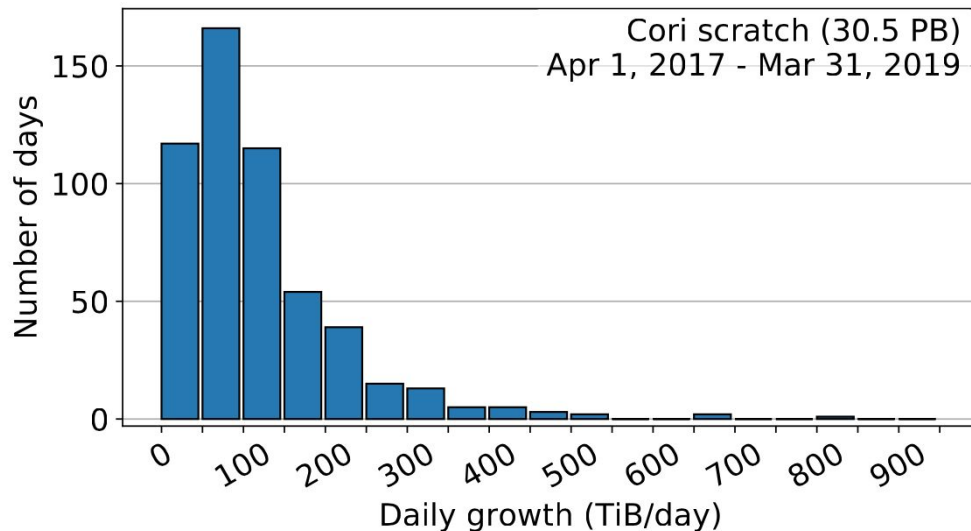


Figure 1 - Distribution of daily growth of Cori's scratch

File System Capacity



$\left(\frac{\partial C^{\text{ref}}}{\partial t}\right)$ Me... projected for... TB/day
We set
 Data retention policy for Perlmutter is at $\text{time} > 28$ days
 $\left(\frac{\lambda_{\text{purge}}}{\text{PF}}\right)$

- OK to purge after that time
- Each purge aims to remove or migrate 50% of the total capacity

SSI Anticipated 3x to 4x sustained system improvement

C^{new} Minimum Per...
Set
 between 22 PB and

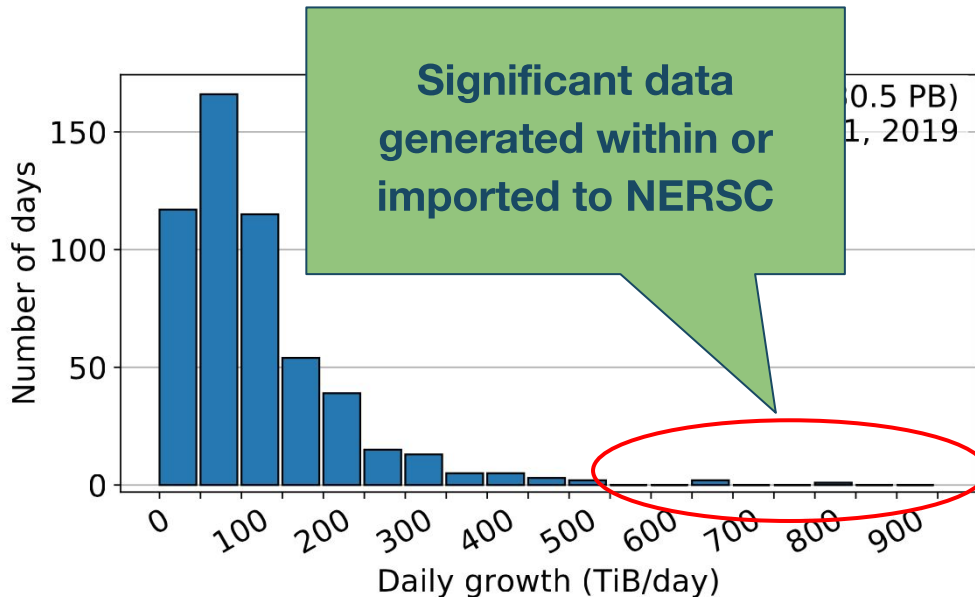
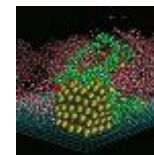
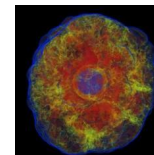
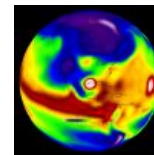
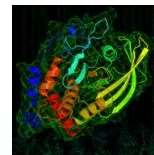
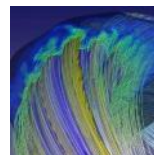
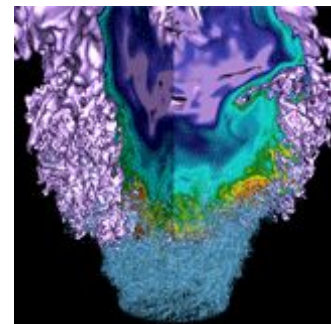


Figure 1 - Distribution of daily growth of Cori's scratch

Drive Endurance



Drive Endurance



Drive Writes Per Day
required for Perlmutter

File System Writes Per Day
of Cori's total write volume

Perlmutter
parity blocks

$$DWPD^{new} = SSI \cdot FSWPD^{ref} \cdot \left(\frac{D + P}{D} \right) \cdot WAF$$

Sustained System Improvement
3x - 4x output capacity over Cori

Perlmutter
data blocks

Write
Amplification
Factor

WAF The Write Amplification Factor (WAF), which results from factors intrinsic to the application workload, accounts for writes that are smaller than a full RAID stripe (read-modify-write)

- This read-modify-write penalty is a function of the anticipated workload
- SSD garbage collection a contributor as well

$\left(\frac{D + P}{D}\right)$ Data and Parity blocks need to be accounted for, as a single user write is accompanied by additional parity blocks when written to physical media

FSWPD^{ref} File System Writes Per Day (FSWPD) can be derived from Cori directly via telemetry or indirectly from device-level counters

- Lustre file system level data unambiguously shows the user workload in absence of device-level buffering or amplification specific to RAID

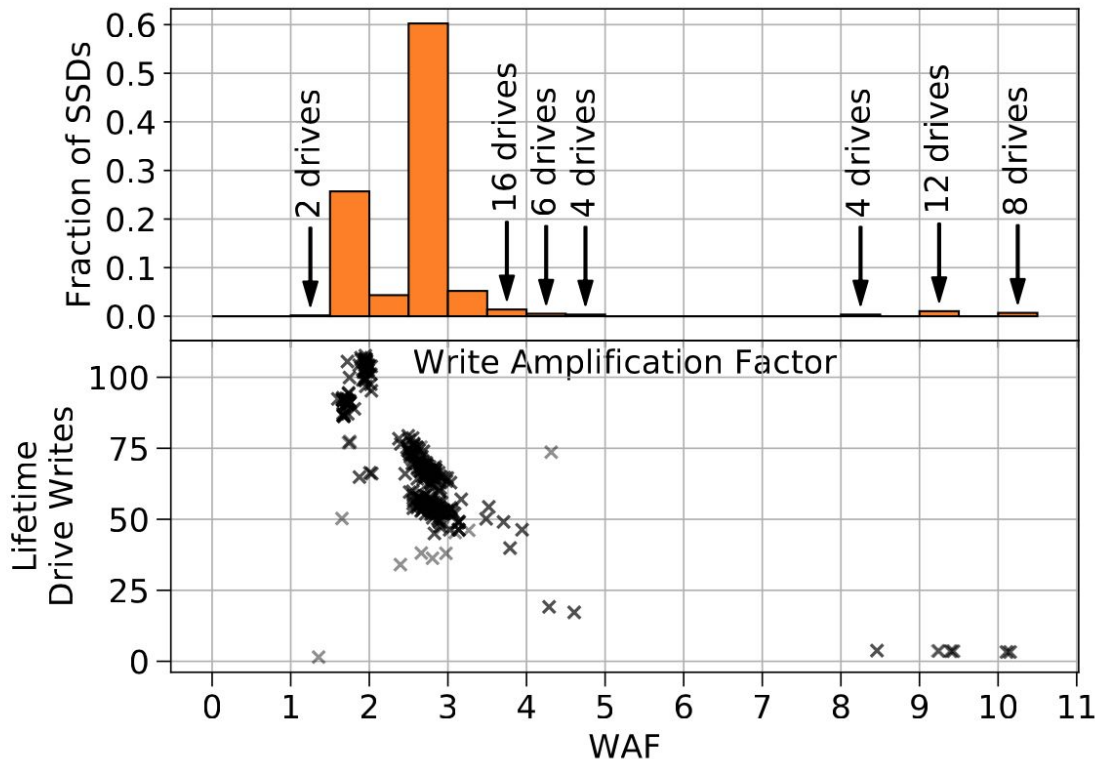
Drive Endurance



Figure 2

Distribution of SSD WAFs on the Cori Burst Buffer after ~ 3.4 years in service (top)

Total lifetime write volumes, normalized to formatted drive capacity, for the WAF distribution (bottom)



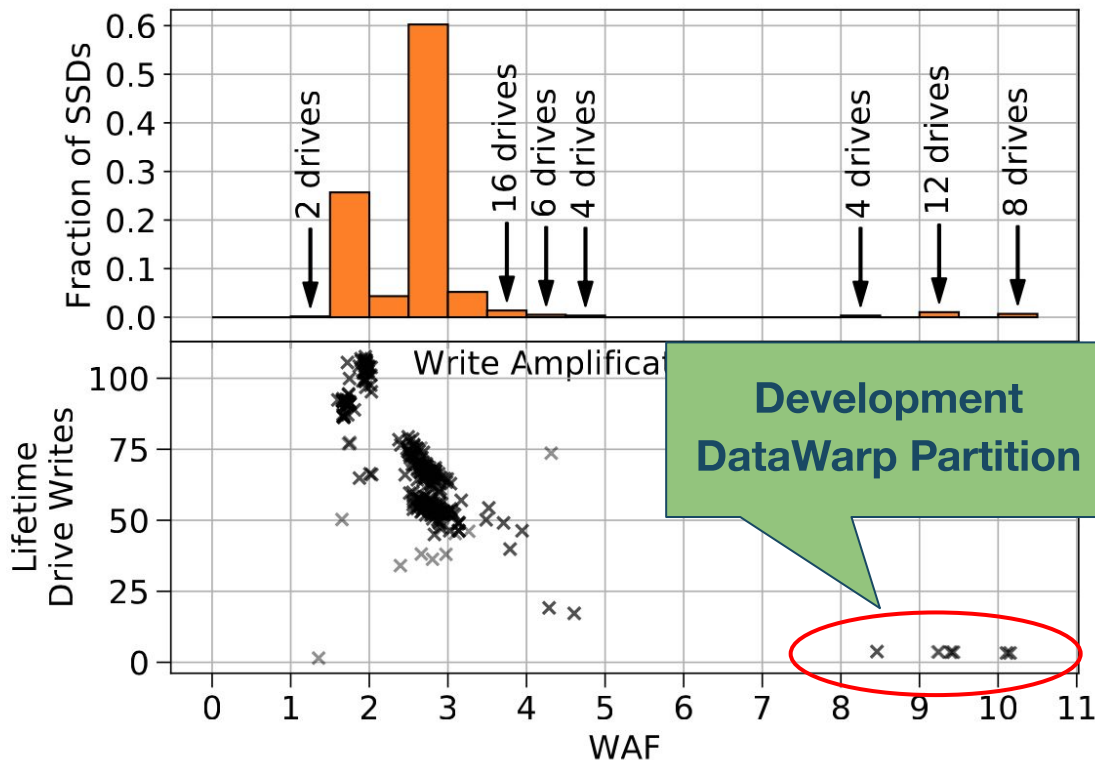
Drive Endurance



Figure 2

Distribution of SSD WAFs on the Cori Burst Buffer after ~ 3.4 years in service (top)

Total lifetime write volumes, normalized to formatted drive capacity, for the WAF distribution (bottom)

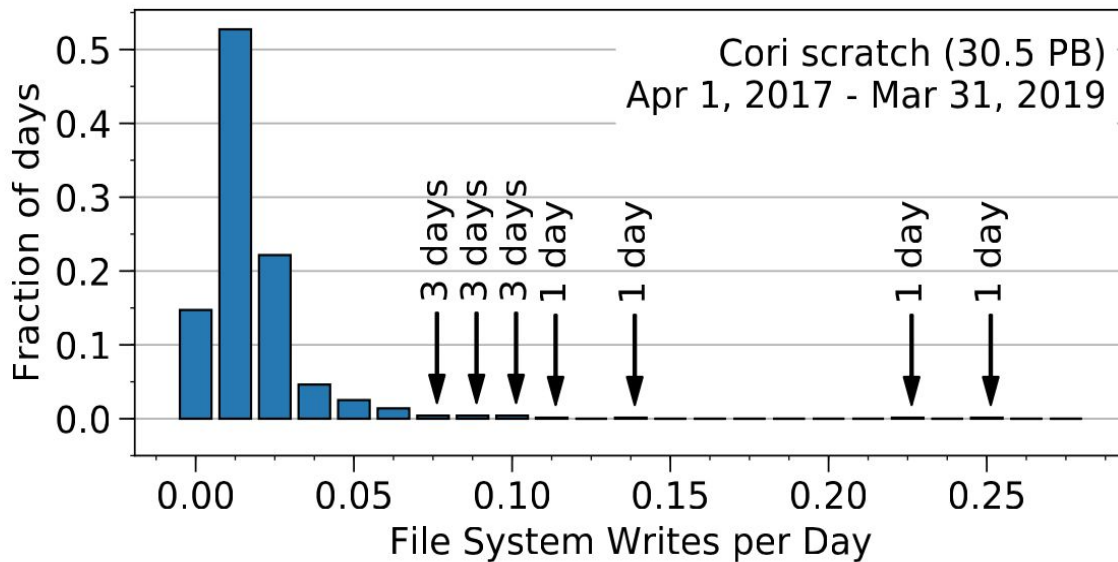


Drive Endurance



Figure 3 - Cori scratch write distribution over 2 years, using LMT

- 1 FSWPD = 30.5 PB of writes per day
- Nonzero fractions in the tail are annotated in absolute days
- Long tail of days that experience abnormally high write volumes (scratch being used as a data processing capability)

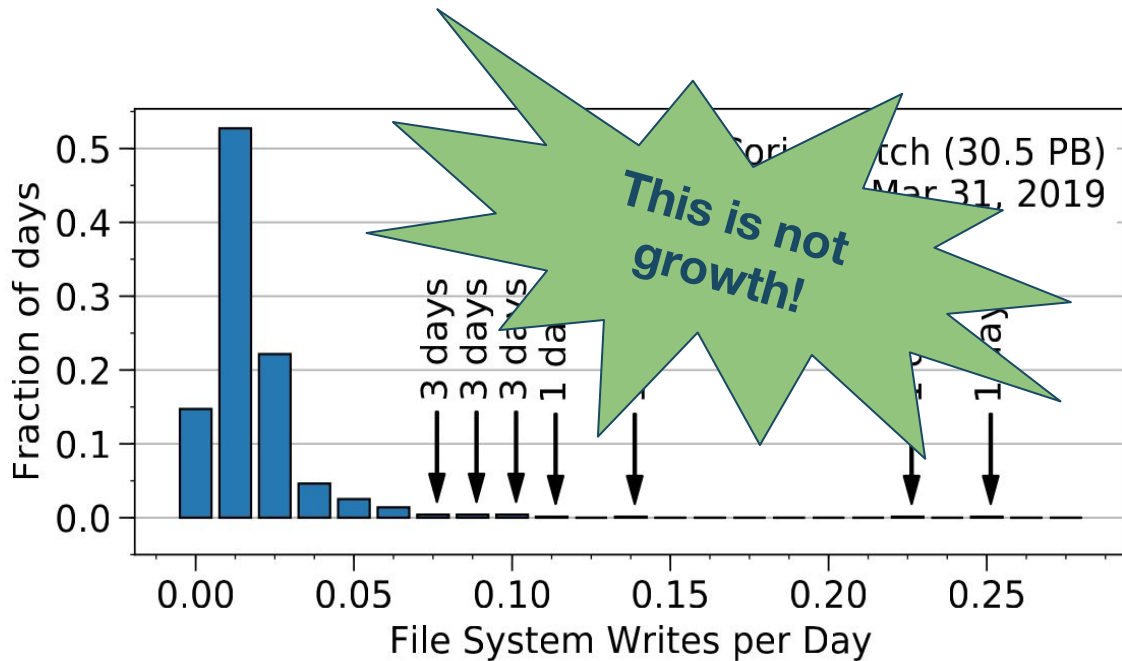


Drive Endurance



Figure 3 - Cori scratch write distribution over 2 years, using LMT

- 1 FSWPD = 30.5 PB of writes per day
- Nonzero fractions in the tail are annotated in absolute days
- Long tail of days that experience abnormally high write volumes (scratch being used as a data processing capability)



Drive Endurance

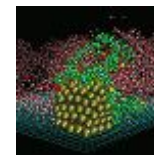
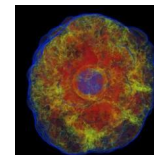
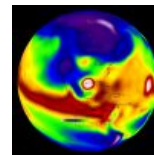
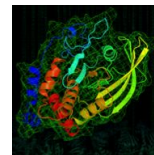
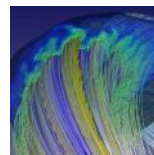
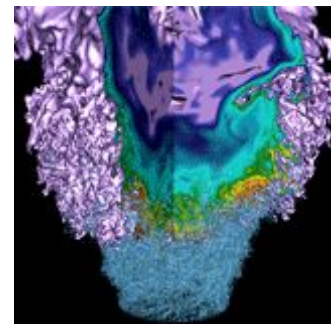


- Many HPC deployments utilize extreme-endurance SSDs (\$\$\$\$)
- NERSC reserves up to 20% of Cori's Burst Buffer SSDs for wear leveling
 - Effectively enduring 10 DWPD
 - Instead of the 3 DWPD as per factory default
- **This endurance is not needed!**

- The continually increasing bit density of NAND, allows for larger drives, and an increased DWPD (absolute endurance)
- There is a trade of performance for endurance, since per-SSD performance does not scale with per-SSD capacity
- Looking at file system-level load data and sources of write amplification:

$DWPD^{new}$ **1 DWPD leaves significant headroom for the anticipated Perlmutter workload**

Metadata Configuration



- Lustre's Data-on-MDT (DOM) feature allows for a configurable number of bytes of every file to be stored on the same storage devices as their file metadata
- Major benefits are
 - Lock traffic is reduced since data and metadata are colocated
 - File size can be determined without sending RPCs to OSSes
 - Small file I/O interferes much less with large-file I/O on OSTs
- However, DOM adds additional complexity to system design, must now account for:
 - Capacity to store inodes
 - Capacity required to store small files' contents
- **Precise definition of what constitutes "small" is site-configurable**

Metadata Configuration

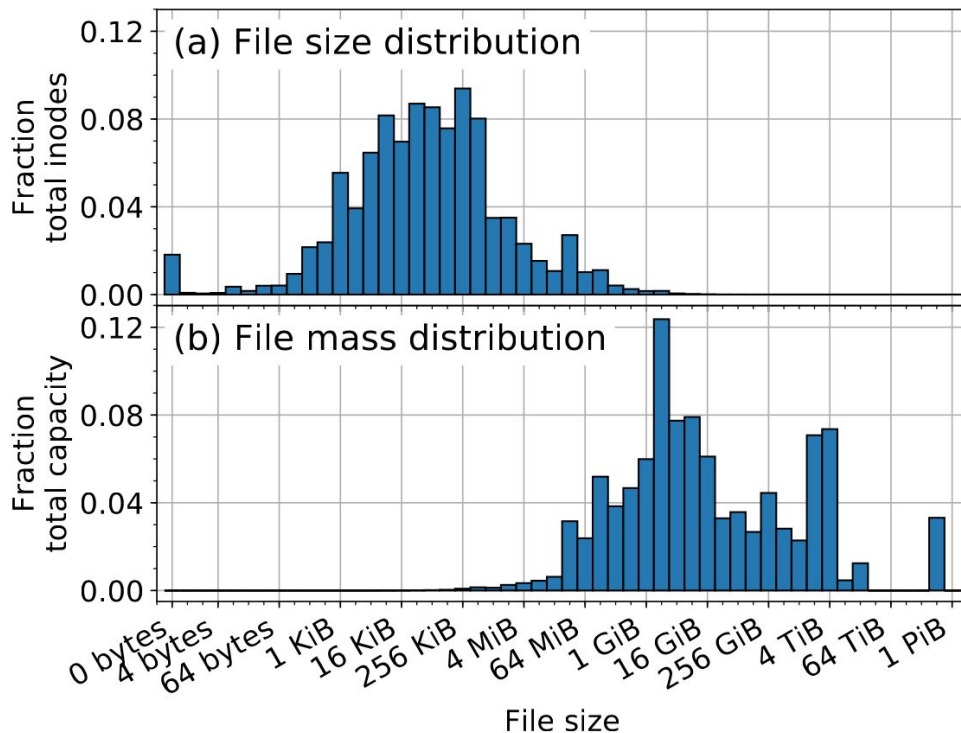


Figure 4 - Probability distribution of file size and file mass on Cori's file system in January 2019

95% of the files comprise only 5% of the capacity used

MDT capacity for a new system is a function of the *expected* file size distribution

- Average file size alone is not enough because HPC file size distribution skews towards small files
- Small changes to the mean file size could represent a significant change to where the optimal DOM size threshold should be



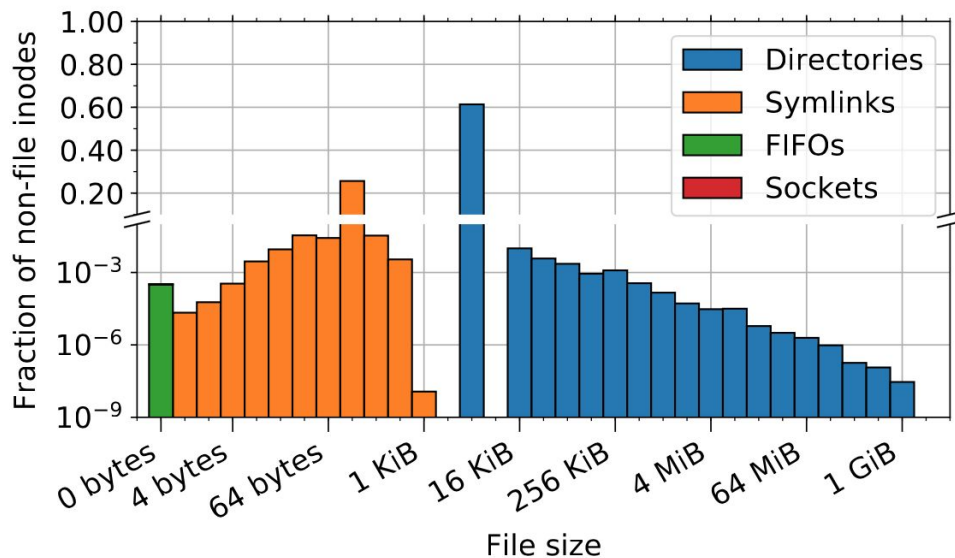
Metadata Configuration



Figure 5 - Probability distribution of inode sizes on Cori's file system in January 2019

MDT Capacity Required for Inodes

- Lustre reserves 4 KiB of MDT capacity per inode
- BUT Directories with millions of files are significantly larger
- Most extreme case is 1 GiB in size for 8 million child inodes



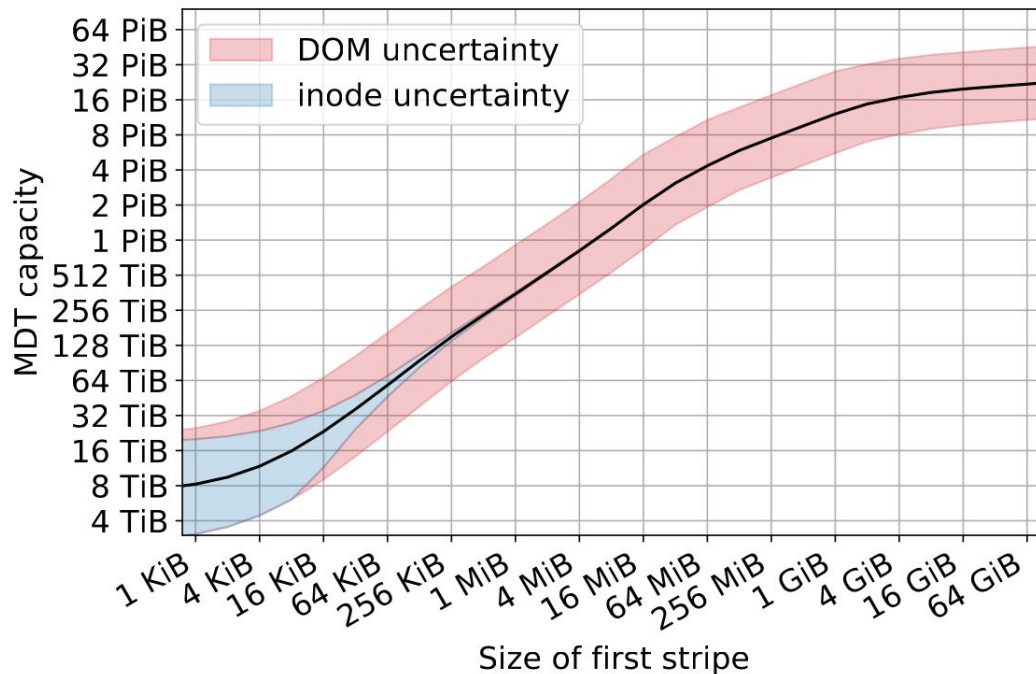
Metadata Configuration



Figure 6 - Required MDT capacity as a function of DOM threshold

Shaded area bounded by the minimum and maximum estimated requirements dictated by the DOM component and the inode capacity component of MDT capacity

- At a very small DOM threshold, the large number of small files does not consume much MDT space
- At a very large DOM threshold, the great majority of files are stored entirely within the MDT, and only a small number of very large files dictates a higher MDT capacity



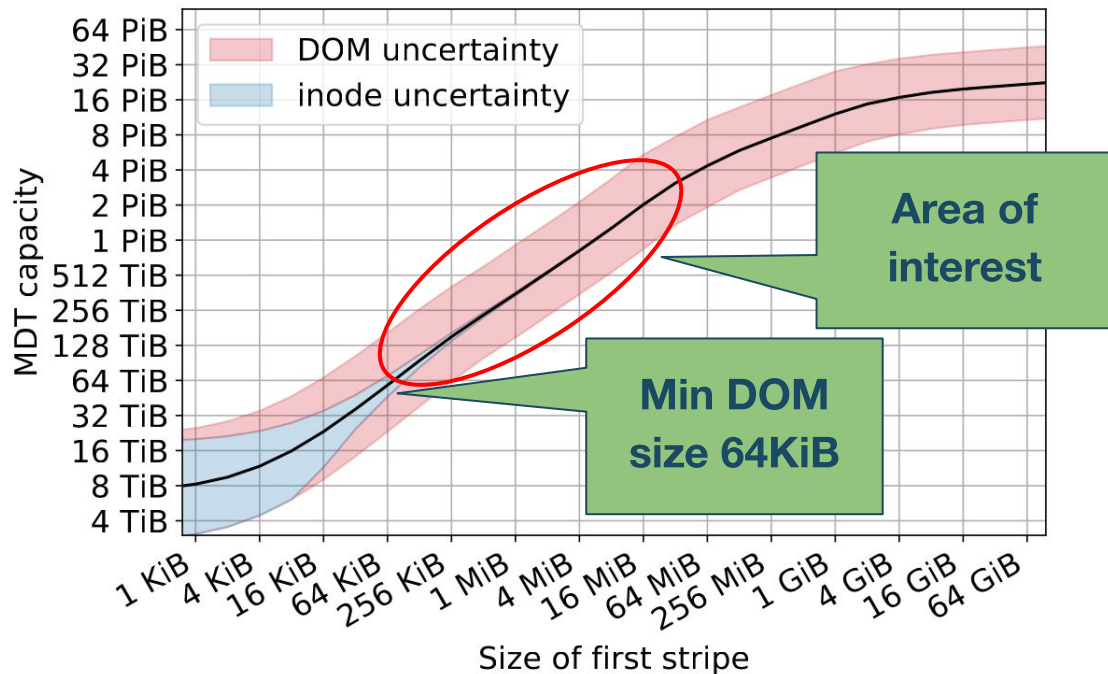
Metadata Configuration



Figure 6 - Required MDT capacity as a function of DOM threshold

Shaded area bounded by the minimum and maximum estimated requirements dictated by the DOM component and the inode capacity component of MDT capacity

- At a very small DOM threshold, the large number of small files does not consume much MDT space
- At a very large DOM threshold, the great majority of files are stored entirely within the MDT, and only a small number of very large files dictates a higher MDT capacity



- Assuming the capacity of DOM is proportional to cost and the DOM threshold is proportional to IOPS performance:

Figure 6 becomes a price-performance curve as well!

- In this case, increasing the DOM threshold above several GiB is not an optimal configuration for price/performance
- DOM threshold is inversely proportional to bandwidth performance since DOM is not striped, so choosing a large DOM threshold would have a negative impact on a per-file bandwidth

- Workload data from a reference system can be used to determine the best balance of
 - Cost
 - Performance
 - Usability
- Quantifying the relationship between
 - Purge policy
 - Growth rate
 - File size distribution
 - Design space parameters
 - Data capacity
 - SSD endurance
 - Metadata configuration
- As the economics of flash continue to displace hard disk drives from HPC performance storage tiers, these analytical methods will become increasingly important in future system deployments

**National Energy Research Scientific Computing Center (NERSC) at
Lawrence Berkeley National Laboratory (LBNL):**

- **Glenn K. Lockwood**
- **Kirill Lozinskiy**
- **Ravi Cheema**
- **Lisa Gerhardt**
- **Damian Hazen**
- **Nicholas J. Wright**

{glock, klozinskiy, rcheema, lgerhardt, dhazen, njwright}@lbl.gov



Thank You



U.S. DEPARTMENT OF
ENERGY

Office of
Science

