NERSC

National Energy Research Scientific Computing Center

Greg Butler
Ravi Cheema
Damian Hazen
Kristy Kallback-Rose
Rei Lee
Glenn Lockwood

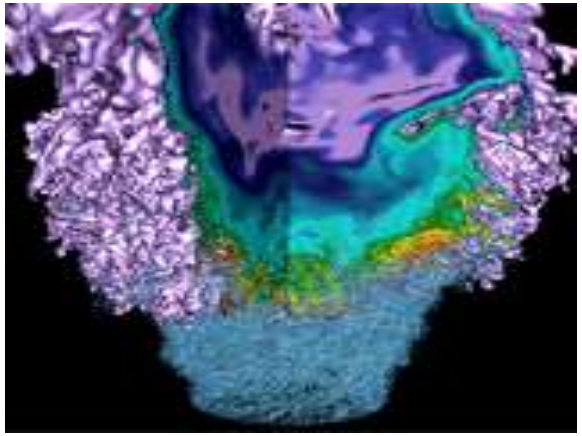
NERSC Storage Systems Team

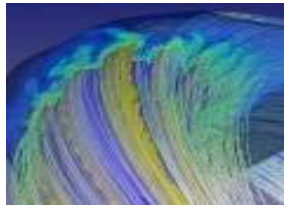
March 4 , 2020

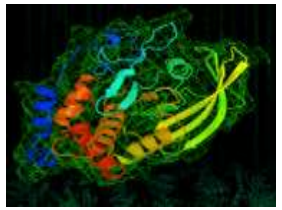
Agenda

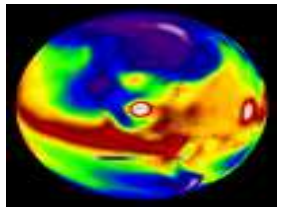
- **General NERSC & Systems Overview**
- **Storage 2020 Strategy & Progress**
- **Community File System & Data Migration**

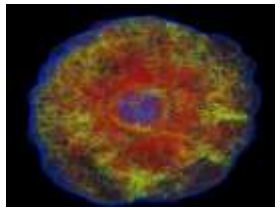
NERSC & Systems Overview





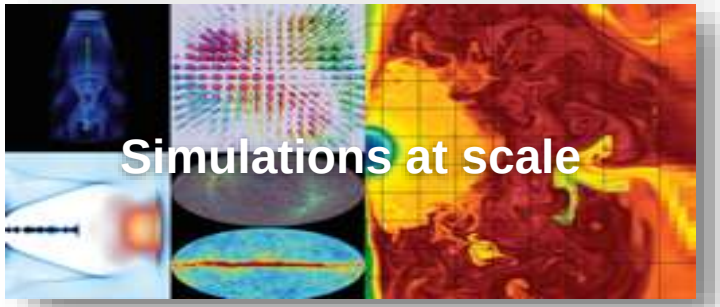




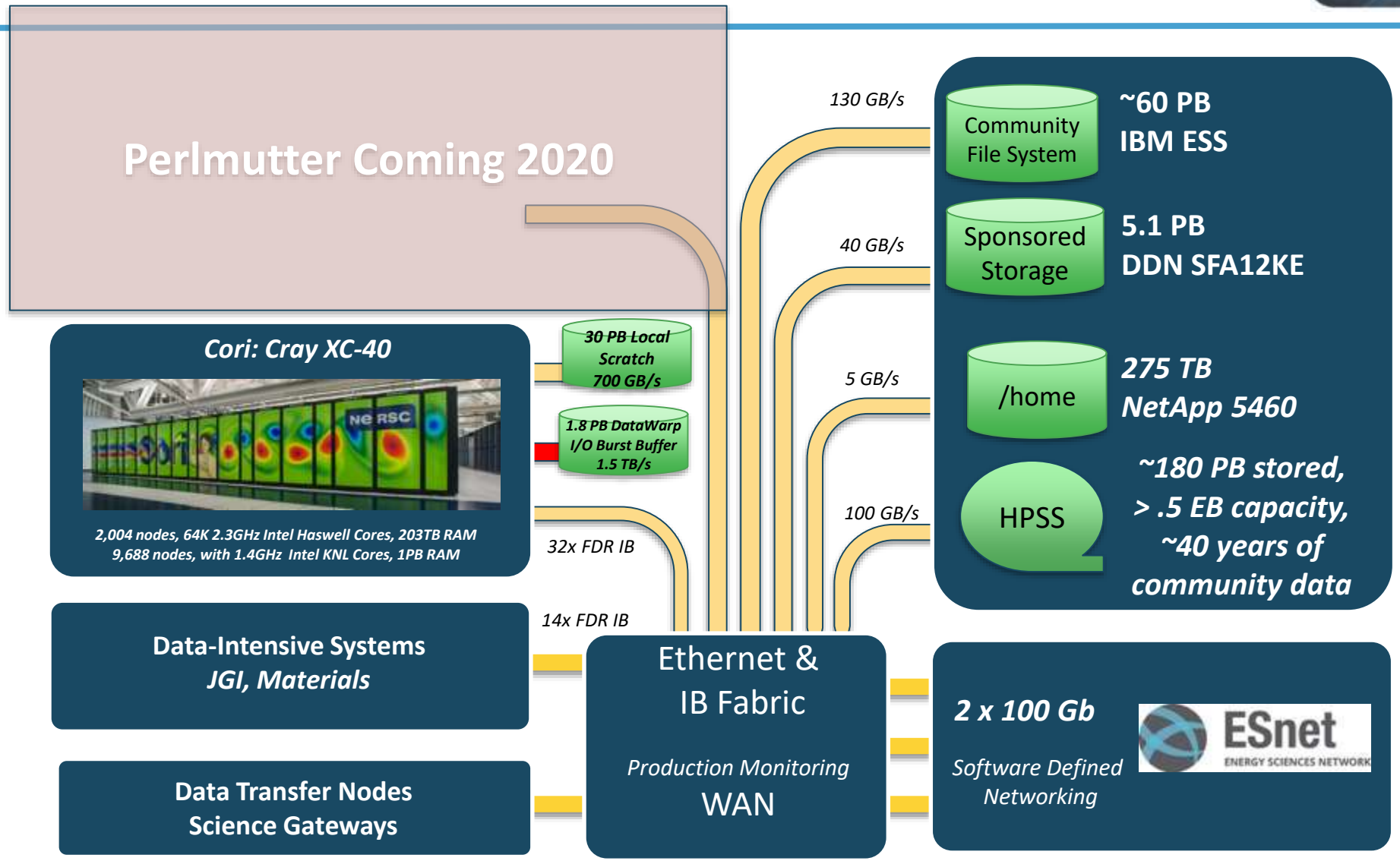


NERSC @ Berkeley Lab (LBNL)

- **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**

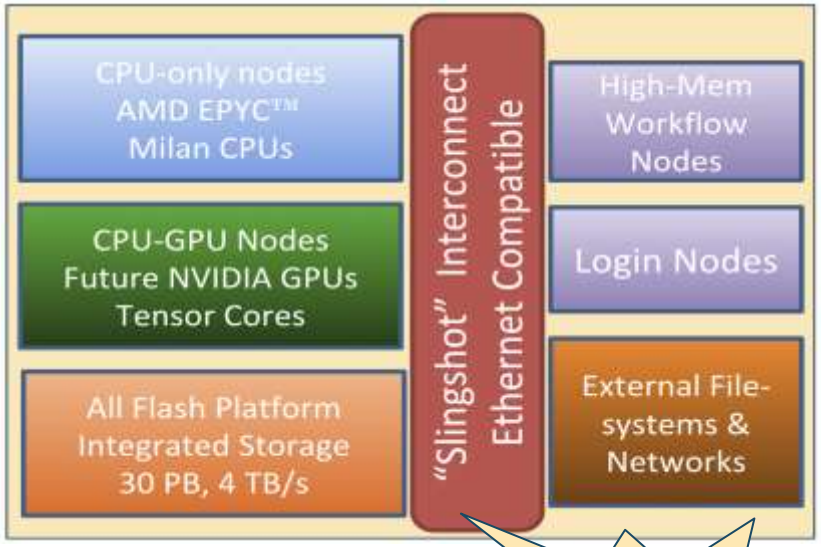


NERSC - Resources at a Glance 2020



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

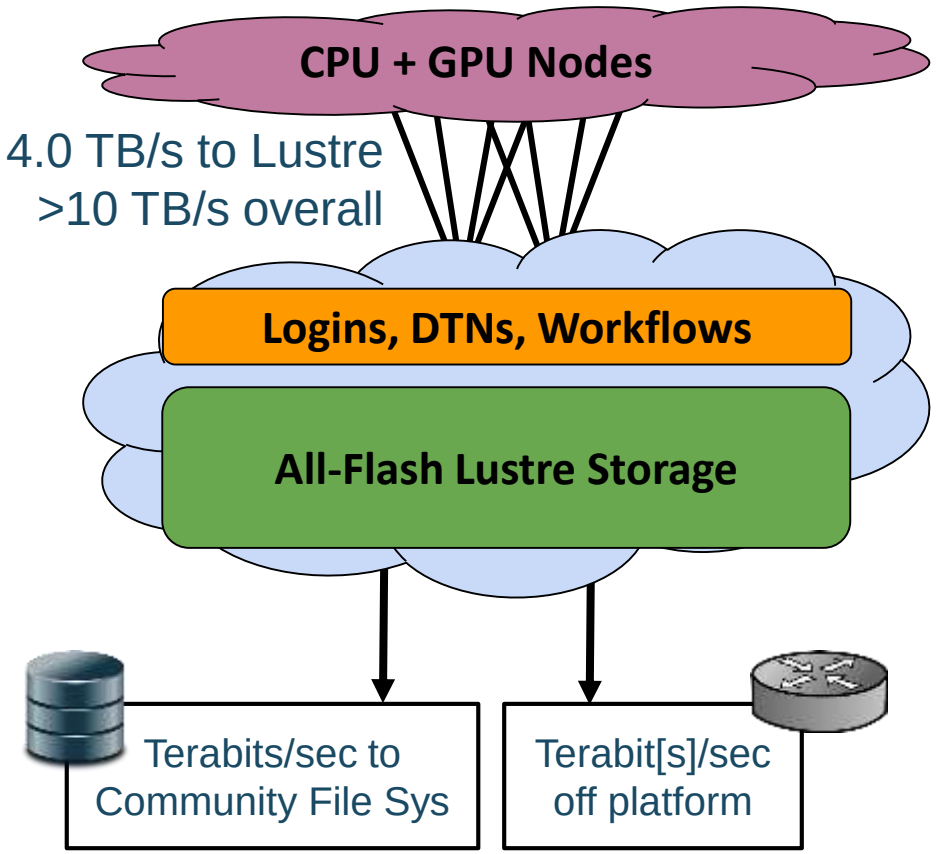
NERSC-9's All-Flash Architecture

Fast across many dimensions

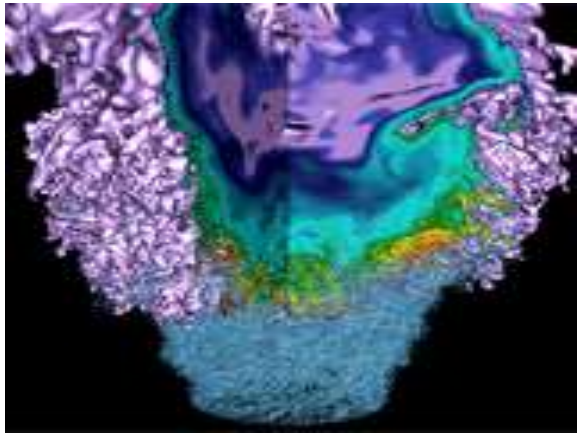
- 30 PB usable capacity
- ≥ 4 TB/s sustained bandwidth
- ≥ 7,000,000 IOPS
- ≥ 3,200,000 file creates/sec

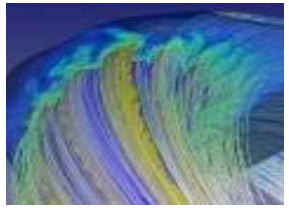
Integrated network, separate groups

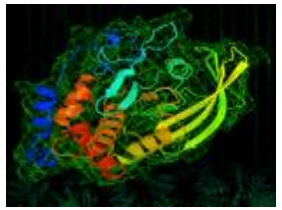
- Storage/logins remain up when compute is down
- Lustre on same fabric as computes (No LNET needed)

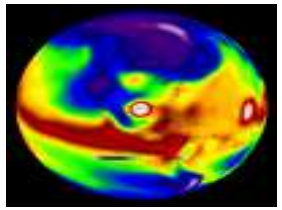


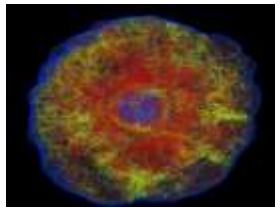
Storage 2020
Strategy & Progress





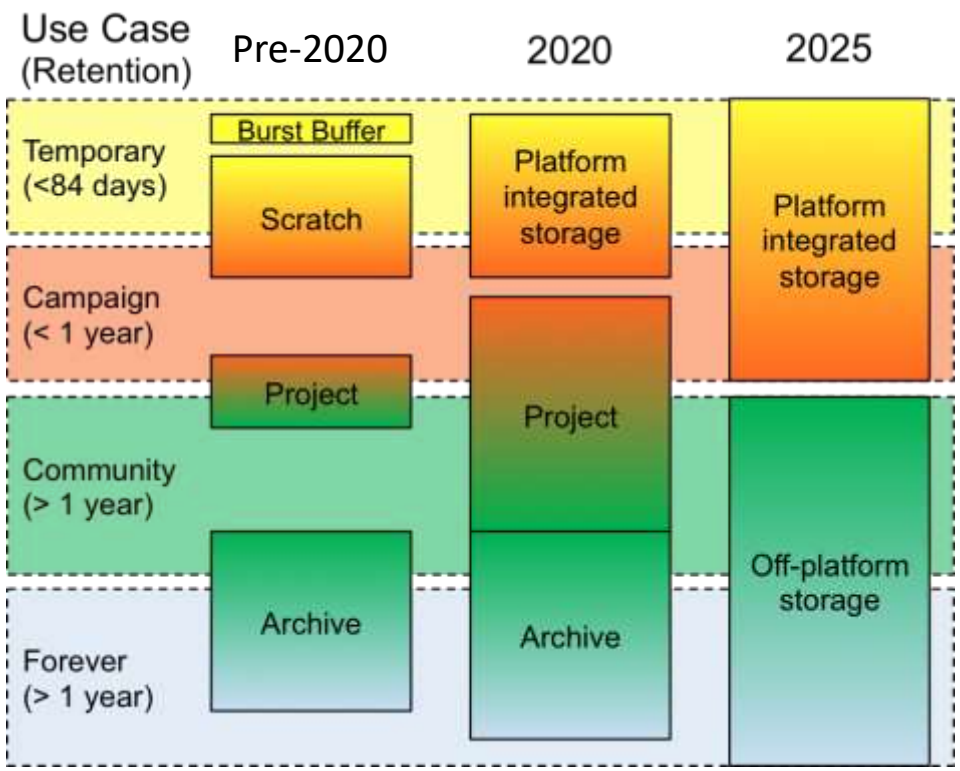






NERSC Storage 2020: Design goals

- **Target 2020**
  - Collapse burst buffer and scratch into all-flash scratch
  - Invest in large disk tier for capacity
  - Long-term investment in tape to minimize overall costs
- **Target 2025**
  - Use single namespace to manage tiers of SCM and flash for scratch
  - Use single namespace to manage tiers of disk and tape for long-term repository



Storage 2020: A Vision for the Future of HPC Storage https://escholarship.org/uc/item/744479dp
G. K. Lockwood *et al,* Berkeley, CA, 2017.

Multiple Storage Tiers

Lustre "Scratch" and Burst Buffer

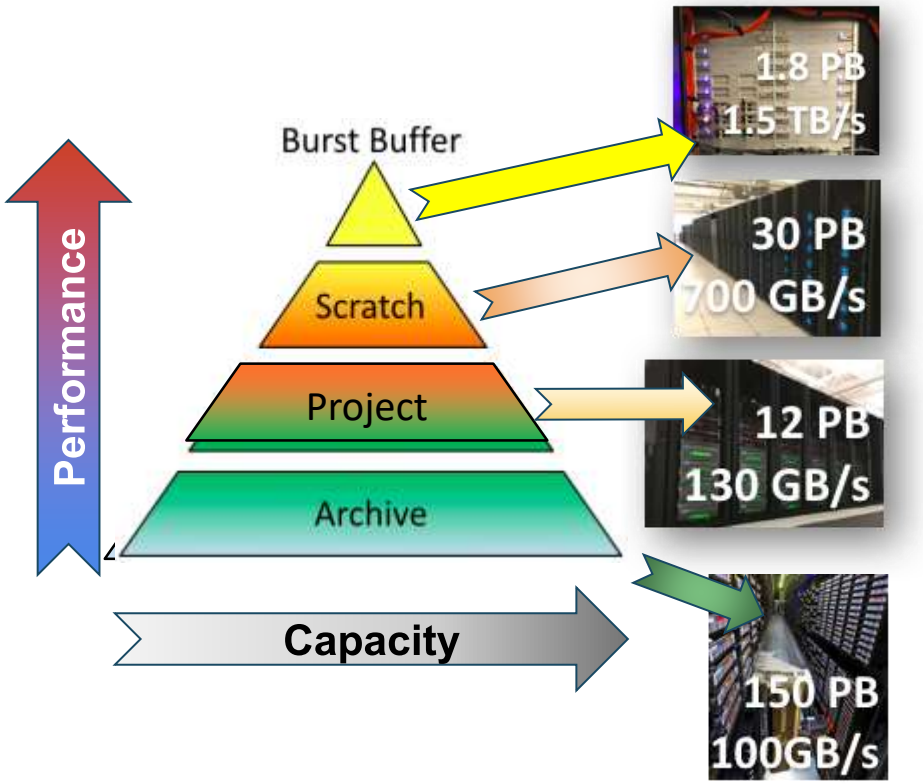
- **Ephemeral storage, data purged if not accessed, user-based quotas and permissions**
- **Intended for high-speed access to active data used for running computations**

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**

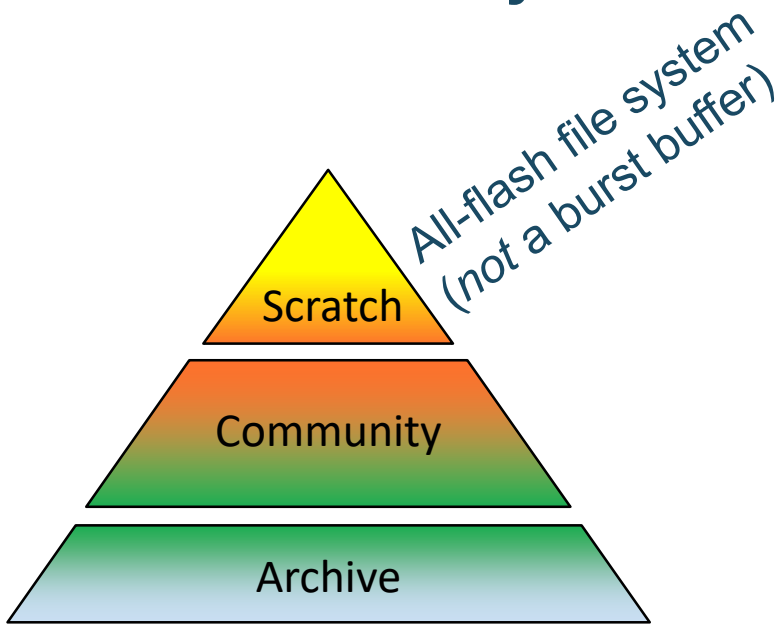
HPSS Tape Archive

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

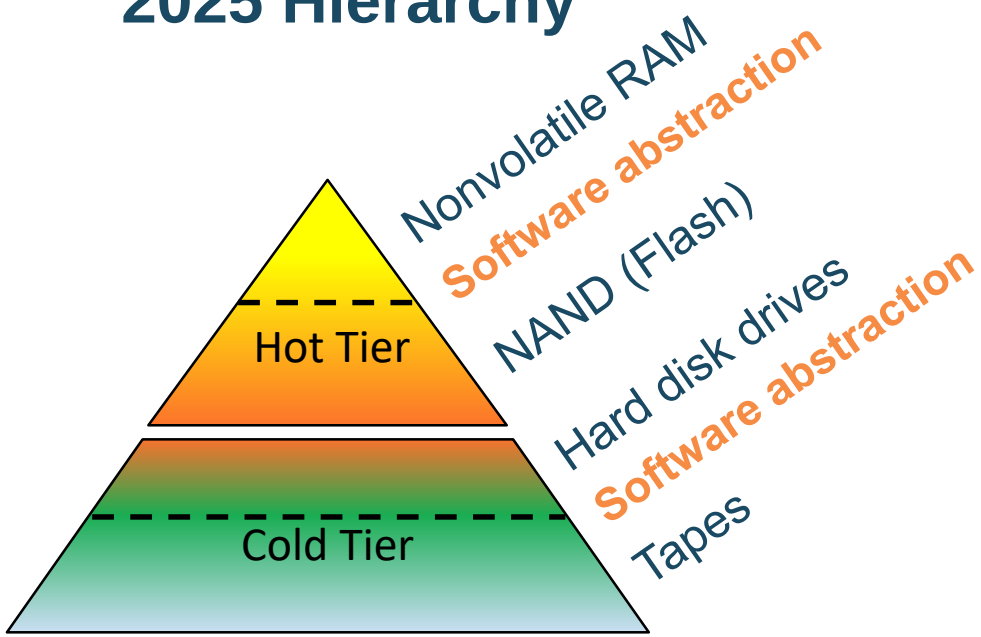


The future requires smarter storage systems

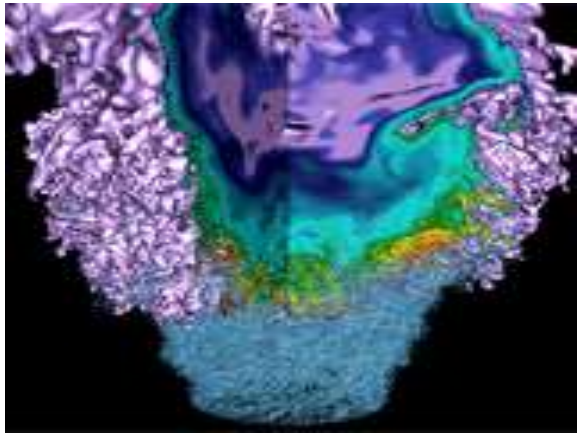
NERSC's 2020 Hierarchy

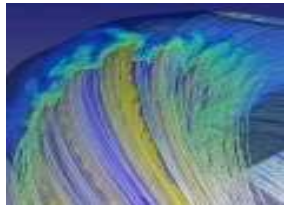


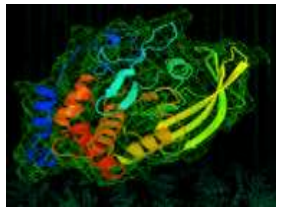
NERSC's 2025 Hierarchy

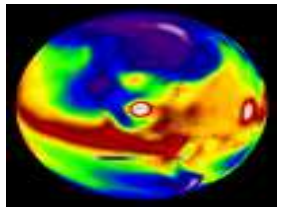


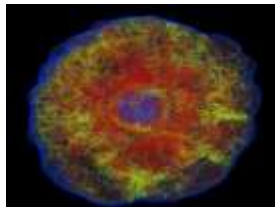
Community File System & Data Migration



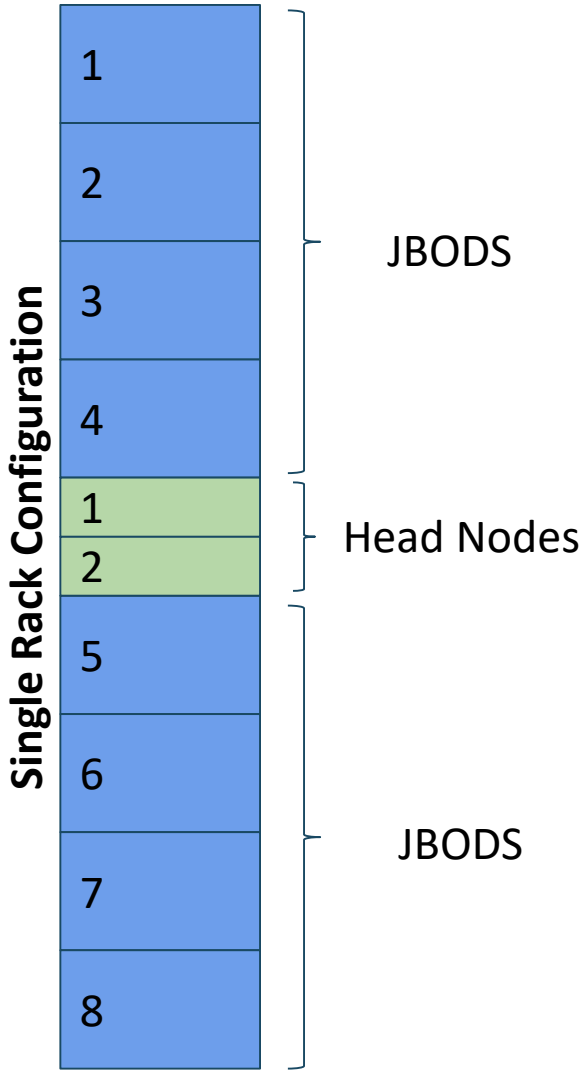








Community File System
ESS GL8C Rack Layout



Production System:

- 7 Racks x 11.8 PB per rack
- + ESS Management System (1 server)

7 x 2 head nodes:

- 2x 10-core 2.45 GHZ POWER8 processors
- 256 GiB memory
- RHEL 7 (LE)
- IBM ESS software (5.3.5)
- 2x Mellanox dual-port EDR IB HCAs (CX-5)
- 5x quad ported 12Gb/s SAS PCIe Gen3 x8 HBAs

Read: GPFS w/DCR

7 x 8 JBOD enclosures:

- 4U/106 bay (Seagate)
- 14TB drives (Seagate)

Test System:

- 2x head nodes
- 2x JBOD enc w/10 TB drives
- + ESS Management System

GPFS/SS → GPFS/SS – Data Migration

- **Began after system acceptance; that process ended December 2019**
- **Immovable go-live mid-January (allocation year roll over)**
- **New filesystem (Community) and existing (Project); both GPFS**
- **Significant amount of data to move:**
  - ~6PB
  - ~1B files and dirs
  - ~1500 dependent filesets (roughly one per project)
- **No consistent patterns we can rely on**
  - sparse, small, large files everywhere
  - Occurrences of lots of small files in one dir
  - “creative” filenames (spaces, pipes in names)

Data Migration– What's the plan?

- **Could have users to migrate there own data**
  - + Users *might* prune their data
  - + Admins use their time elsewhere
  - - Each project directory could be associated with 50 users
    - - Difficult to coordinate
  - - Undefined beginning and end of migration
  - - Expect significant number of Qs (== not less work)
  - - Likely migration *storm* as deadline approached
  - - Decided this is *not* the plan

https://waitbutwhy.com/2013/10/why-procrastinators-procrastinate.html

Data Migration– What's the plan?

- **GPFS AFM *almost* there for data migration**
  - ~1500 dependent filesets in Project
    - AFM code for this came with 5.0.4
  - Unsure how we could see if files were successfully copied
  - Code fix for directory-level time stamps came May 2019
  - Lots of AFM improvements in 5.0.2, 5.0.3, 5.0.4, but
    - Not keen to test new code w/r/t data migrations
  - Decided this is *not* the plan

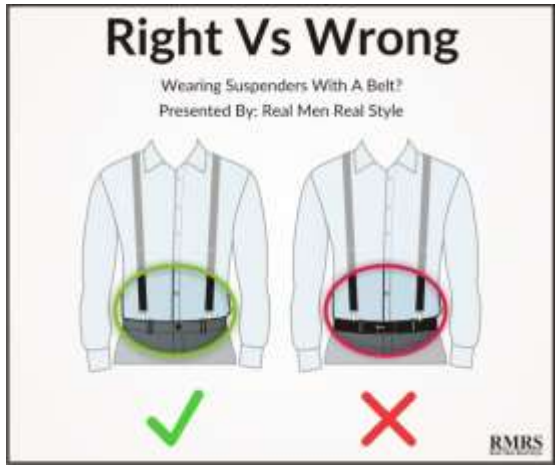
Data Migration – What's the plan?

- **GPFS Restripe would be great!**
  - + Successfully used for data migration from Oakland to Berkeley
  - - Can't go to new filesystem format
    - – No variant sub-blocks
    - – Can't go to new block size;
      - » blocksize old: 4MB; new: 16MB
    - – General concerns about keeping old filesystem format
  - This is *not* the plan

- **MPI fileutils (https://hpc.github.io/mpifileutils), fpsync, Globus and other tools promising, but**
  - – Testing required at scale, did limited testing on small scale
    - Run as user vs root
    - checksum capability
    - GPFS ACLs, EAs
    - Sparse files
    - hard/symlinks, etc.
  - – Could be a large project to examine all the tools (intern?!)
  - – Timeline is tight, so this is *not* the plan

Data Migration– What's the plan?

- **Timeline is tight, data migration is not where we want to be experimental**
  - Rsync, fpart, parallel, cp and tar it is then!
  - This *is* the plan.



Respectfully disagree ^^

https://www.realmenrealstyle.com/menswear-faux-pas-mistakes/

Data Migration Plan Details

Step 0:

- **Used GPFS/SS ILM engine to gen file list**
  - cp/tar files to new filesystem (Project → CFS)
    - 70 largest projects by volume
    - .5PB to 10s of TB
  - Rsync if it was a "small" amount of data
    - For the remainder
    - Regardless of # of files
  - *fpart* to divide per-project file list across servers at 10K files per list
    - *parallel* to distribute tasks across servers
      - 16 concurrent tasks per server
- **Over ~ 3 weeks (holidays), 14 servers used to do this initial copy**

Data Migration Implementation

Step 1:

- **Continue to copy file-level changes**
  - Using updated version of snapshot for subsequent runs
  - rsync per project directory
  - Exceptions: rsyncs >> 24 hours
    - Run dryrun to see determine changes, process separately
    - ~10 projects out of ~1500 required this step
      - 10 with most churn
- **2 separate servers for this step**
- **Inode count from quota report was used to determine if all files were copied**
  - If there was a mismatch between filesystems then ran rsync with rsync dryrun
    - e.g. Target would have a higher count if hard link copied as a file

Data Migration Implementation

Step 1:

- **2 people managed step 0 and step 1 (Rei and Ravi)**
  - Handing off a project when the initial copy was done, so it could be added to the process catching on-going changes
- **Step 1 could be completed in < 2 days for all projects on an ongoing basis**
  - So every 2 days, you've caught up with all the changes
  - Nearing the deadline this was done every 12 hours, so increased snapshot to every 12 hours

Data Migration Implementation

Step 2:

- **Time for final rsync**
  - Take Project filesystem read-only
  - All 16 servers can be employed
  - Took ~12 hours, back sooner than expected
    - Would vary depending upon changes leading up to RO mode

Data Migration Implementation

Reflections on the process:

- Final rsync will catch (almost) everything
  - So, use best tool for initial data migration (probably not rsync)
- Rsync --dryrun is your friend
  - Good sense of churn
- Restartability
  - Keep reasonably-sized units of work, in case of a data mover crash
- Checksums can build confidence, but time consuming
  - Over holiday, after initial copy rsync dryrun with checksum for 99% of files just to verify process
- Migration rates will vary
  - depending upon file size
  - ~5GB/s – ~32 GB/s (aggregate across nodes)
- There's no easy way out,
  - examine data and make a choice based on current condition
- GOAL: Evaluation of other tools at scale under less time pressure

Data Migration Implementation

In the end, the process was a bit hand-held, but we met the most important goals:

- Users did not migrate there own data
- Confidence that the data successfully reached the target filesystem
- Implemented with newer filesystem format
- Downtime from a user perspective was limited
  - We said could be up to 5 days with read-only access to Project filesystem, it was 12 hours
- Aside from minor hard link issue, we have not heard any complaints post-migration
- Many users may not have noticed their data was migrated as we put in a symlink matching the old filesystem path
  - We will expire this in summer

NERSC Storage Team & Fellow Contributors

Thank you.
Questions?

Right to Left:
Greg Butler
Kirill Lozinskiy
Nick Balthaser
Ravi Cheema
Rei Lee
Kristy Kallback-Rose
Wayne Hurlbert

+ Glenn Lockwood
(Acting Group Lead)

+ Melinda Jacobsen
(recently joined the team)
- Damian Hazen
(former Group Lead)

National Energy Research Scientific Computing Center