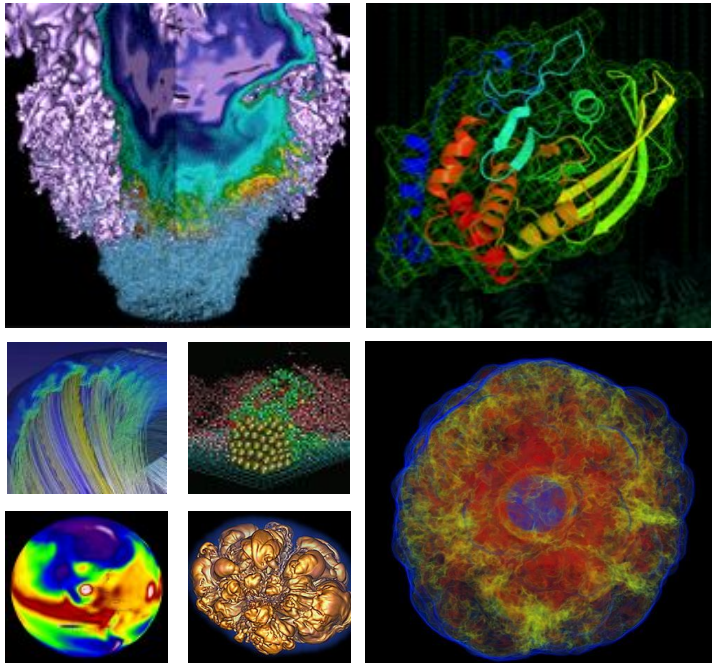
DVS: Best Practices for Reading and Writing Files



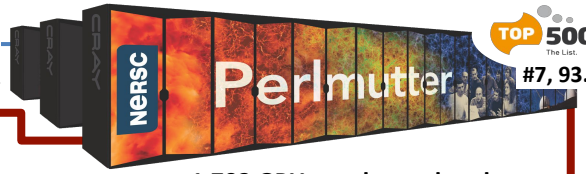
Lisa Gerhardt

NUG Monthly Meeting
August 24, 2023

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

File Systems at NERSC



TOP 500 The List. #7, 93.8PF Peak

>5 TB/s

35 PB All-Flash Scratch

HPE Slingshot 11 ethernet-compatible interconnect
4 NICs/GPU node,
1 NIC/CPU node

1,792 GPU-accelerated nodes
4 NVIDIA A100 GPUs+1 AMD "Milan" CPU
448 TB (CPU) + 320 TB (GPU) memory

3,072 CPU-only nodes
2 AMD "Milan" CPUs
1,536 TB CPU memory

1.6 TB/s

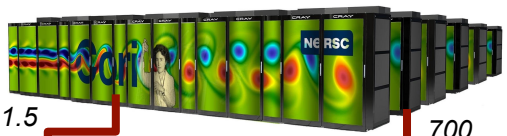
Off Platform Storage

- HPSS Tape Archive ~300 PB — 50 GB/s
- Community File System 130 PB — 100 GB/s
- /home 450 TB — 5 GB/s
- DTNs, Gateways
- Spin container services

Ethernet
Science Friendly Security
Production Monitoring
Power Efficiency
LAN

800 Gbps

Retired June 2023



1.5 TB/s — 2 PB Burst Buffer

700 GB/s — 28 PB Scratch

Cori

9,688 Intel Xeon Phi "KNL" manycore nodes
2,388 Intel Xeon "Haswell" nodes
700,000 processor cores, 1.2 PB memory
Cray XC40 / Aries Dragonfly interconnect
30 PF Peak

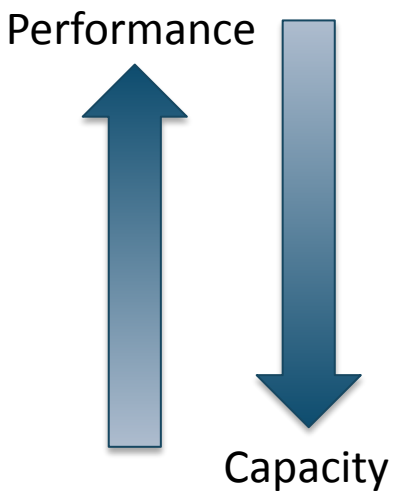
ESnet ENERGY SCIENCES NETWORK — 2 x 400 Gb/s, 2 x 100 Gb/s

Experimental Facility | ASCR Facility | Home Institution | Cloud | Edge

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

File Systems at NERSC



Memory

Scratch

Community

HPSS

Global Common

Global Home

35 PB SSD Perlmutter Scratch

Lustre 6 TB/s, temporary (purge)

130 PB HDD Community

Spectrum Scale (GPFS)

150 GB/s, permanent

300 PB Tape HPSS Archive

Long Term

20 TB SSD Global Common Software

Spectrum Scale, Permanent

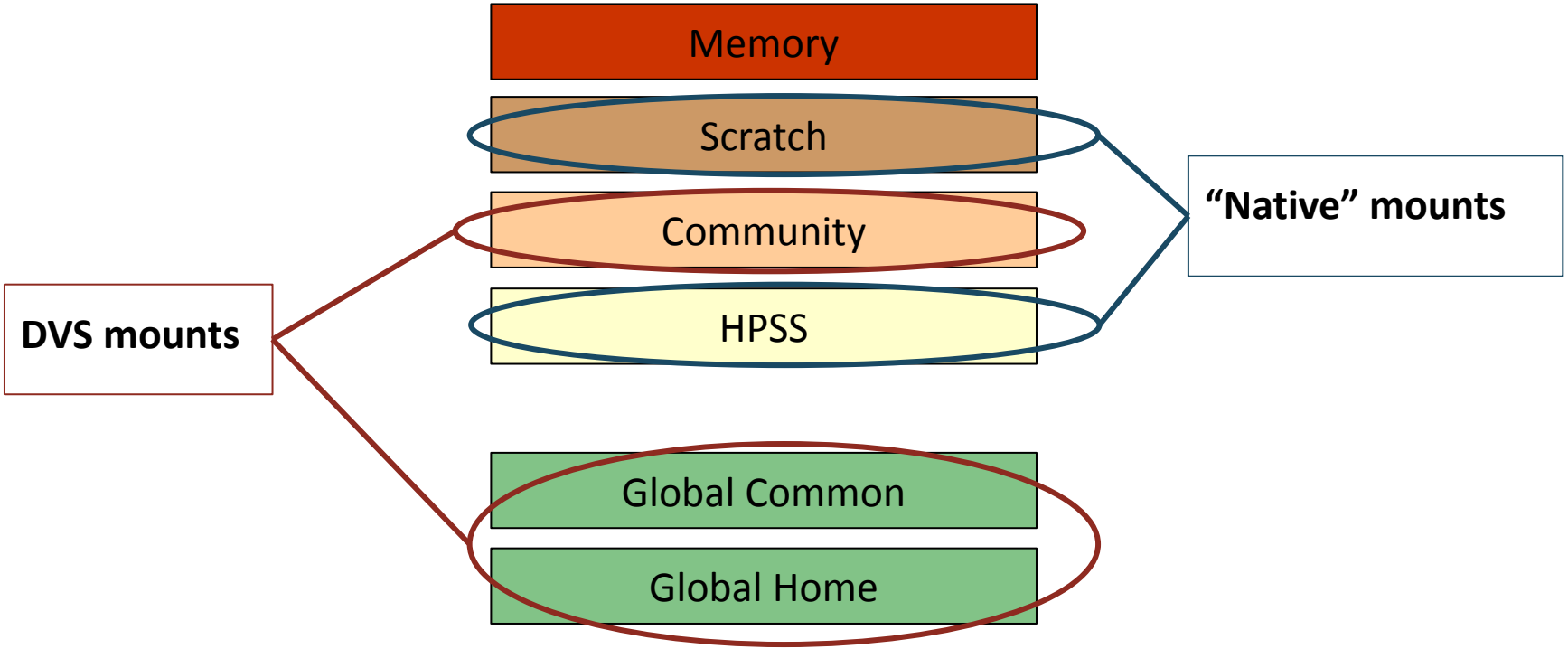
Faster compiling / Source Code

https://docs.nersc.gov/filesystems/

General Advice for I/O

- I/O from batch jobs should go to Perlmutter's scratch file system (/pscratch, $SCRATCH)
  - Input data
  - Configuration files
  - Output data
- Software for batch jobs should go in a container or to Global Common (/global/common/software/<your_project_name>)
  - Conda environments
  - Anything you install with config / make / cmake etc.
- If you're doing anything else, you should pay attention to the rest of this talk

Mounting Matters

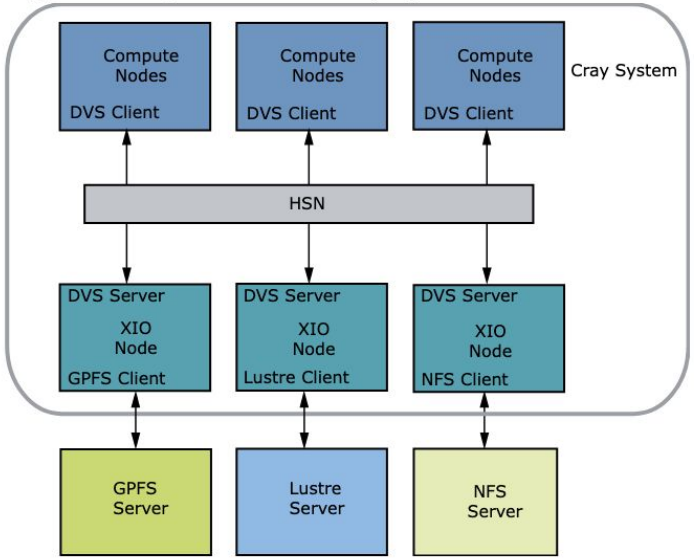


DVS is “new” on Perlmutter. If you’re using these file systems, you may need to change some of the things you’ve been doing

What is DVS?

- DVS is an I/O forwarder developed by Cray
- Designed to deliver file system contents at scale
- Long history of deployment at NERSC, went live on Perlmutter on June 8, 2023
- Only on compute nodes, logins have a native client mount

Figure: Cray DVS In a Cray System



Why DVS?

- Stability!
  - Running Spectrum Scale at a large scale (>5,000 compute clients) with network bottlenecks between the file system and Perlmutter introduced several problems that caused instability in our environment
  - This showed up on the login nodes as long delays reading files, listing files, editing with vi / emacs, etc. and as job failures on the computes

How Does DVS Work?

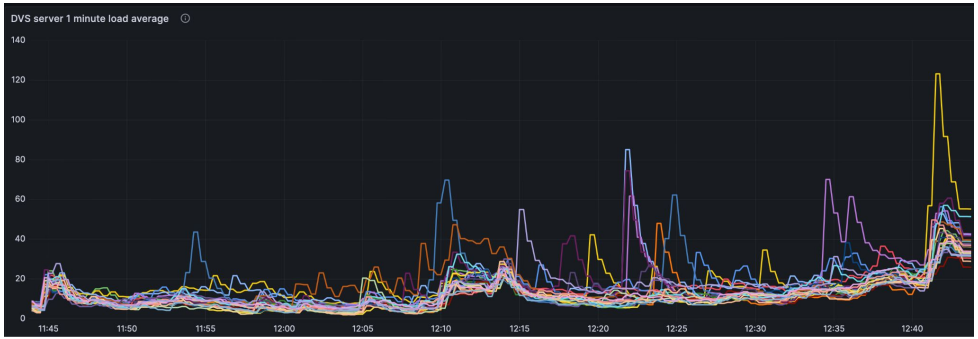
- Perlmutter has 24 gateway servers that serve as DVS servers
- Each server can work 1000 I/O threads at once
- Can cache data to dramatically improve performance at large scales
- Two service modes:
  - Read / Write (RW): gateway server is determined when file is created, stays constant, zero cache
  - Read Only (RO): file can be served by all gateways, stays in cache for 30 seconds
- Global common in mounted RO, all others have default mounts that are RW and alternative mounts that are RO

How Do Users Interact with DVS?

- Intentionally
  - CFS: data stores that are too large for scratch, config files
  - Global Common: SW installs
- Unintentionally
  - Homes: default conda install location, SW installs, scripts, hidden dependencies
  - CFS: SW installs, hidden config files and dependencies

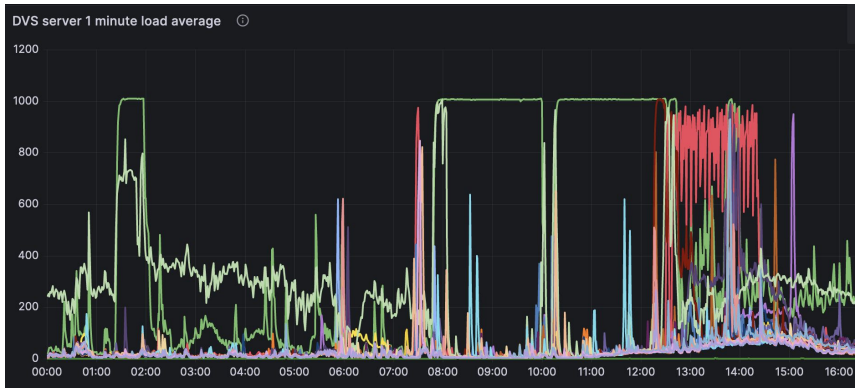
A Tale of Two Loads: The Good

- A job starts on 100 nodes that uses a conda install in Global Common
- All 12,800 processes are spread across 24 DVS servers. The file '/global/common/software/rock/elvis/.conda' is fetched once and then aggressively cached for quick return
- The job starts right away



A Tale of Two Loads: The Bad

- A job starts on 100 nodes that uses a conda install in $HOME
- All 12,800 processes line up on the same DVS server and wait to fetch (and re-fetch) the file '/global/homes/e/elvis/.conda'
- The entire job waits on a single DVS server



Not-so-fun-fact: Up until August 16th, this would have also made **every other user** on this server wait too. Last week we deployed a fairness algorithm on the DVS servers to serve I/O requests equally across users.

What Should I Do?

- If you want to read from CFS, use “/dvs_ro” instead of “/global”
  - “/global/cfs/cdirs/myproject/mega_important_config” -> “/dvs_ro/cfs/cdirs/myproject/mega_important_config”
- If you want a conda environment at scale, use a container or global common: https://docs.nersc.gov/development/languages/python/nersc-python/#moving-your-conda-setup-to-globalcommonsoftware
- Avoid ACLs on files over DVS. These keep the system from using any caching and slow things down

Work is Ongoing

- NERSC is continually working to improve the I/O experience
- If you have any questions or see any unexpected results or performance, please open a ticket

Thank You