SDC | StorageAI™

A SNIA Event

April 29, 2026 • Denver, Colorado

PNFS Past, Present, and Future: What's all the excitement about?

Gary Grider, LANL

LA-UR-26-20882

PNFS Background: Building on PNFS beginnings and NFS improvements over decades

U of Mich NFS V4/PNFS (2002-09) - eliminate FS client for the PFS's (Lustre, Panasas, GPFS, PVFS) DOE/DOD/NSF funded

- § File maps (where is file, is it spread and how, is it replicated, is there locality involved in where it is, etc.)
- § Statefulness/recallable maps (moving data transparently/etc.)
- § Compound RPC (less chat/more intents)
- § Server redirects
- § Data servers can be just NFSV3 servers (standard part of Linux), as well as object and block
- § PNFS metadata server becomes NFSv4.2-> metadata server to enable parallel/etc.

- Some things we didn't think of have been added to NFS as well
  - § Re-export – nfs server can reexport files from another nfs server – cached copies/etc.
  - § Multi stream from one client node – eat all BW from client to server
  - § NFS RDMA using either OFED or TCP or other – efficient movement
  - § NFS Local-IO, NFS-Direct, etc. (more on this later)

- Several PNFS solutions exist today
  - Hammerspace, Pure, Peak:AIO
  - Panasas->Vdura, NetApp
- Two other solutions that use PNFS technology in development

From Open/Proprietary/non standard/HPC community to Proprietary HPC and AI solutions and back to open (but this time open/standard with broad community (NFS, AI, HPC, Cloud, etc.)

HPC driven solutions

lustre®

IBM Spectrum Scale

IA grows the market – enter many Proprietary Solutions

PNFS: Standards Based

PNFS: More than just a PFS client, an open/standard ecosystem for scalable/parallel storage

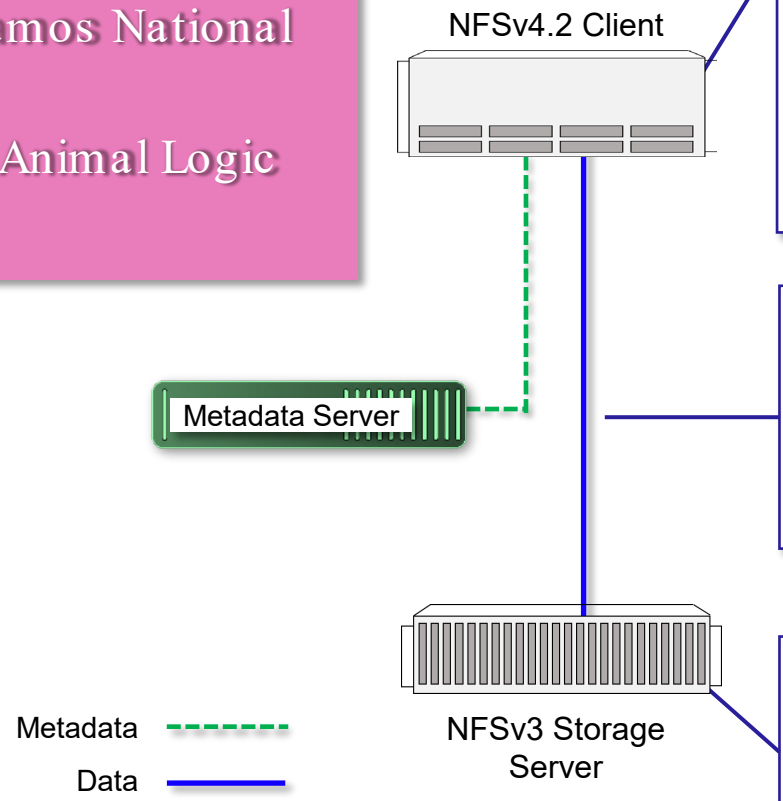
- File System Type consolidation (flexible and powerful enough to do what many file/storage system types do)
  - The best of shared, scaleout NAS, HPC
  - /home /projects /scratch /archive /remotesite/…
  - Distance (compound RPC, caching, reexporting, transparent relocation, ..)
  - HPC N to N ~= Scale out NAS (file mapping, transparent relocation
  - HPC launch/AI training (massive read storms of small files or parts of files) (reexport, caching, local-IO, transparent relocation)
  - Efficient movement, thin stacks for low latency (iops and bw delivered) (RDMA, DirectIO, separate data and metadata paths and separate caching mechanisms)
  - Metadata scaling (referrals, use of shared memory concepts being worked)
  - HPC N to 1 (being worked on now)
  - Client-network/2 tier erasure (being worked on now)

Standards-Based Parallel File System Architecture

-- Parallel NFS v4.2 with Flex files --

Powering High-Performance:

- AI → Meta's Llama 2, 3, &4
- HPC → Los Alamos National Lab
- VFX → Netflix/Animal Logic
- Web → PayPal



Hardening pNFS for HPC/AI Workloads

- 2018: Enhanced Parallel NFS spec with pNFSv4.2 with Flex Files
  - Eliminated NFS GETATTR chattiness
  - Added telemetry feedback
  - Added N-Connect
- Part of every modern Linux distribution

Separate Metadata and Data Paths

- Direct data path using TCP or RDMA
- Provides for multiple parallel connections
- Metadata layer acts as global control plane

Use Any NFSv3 Storage System

- Add any NFS storage volume, from any vendor
- Including GPU server local NVMe

Courtesy Hammerspace

Recent/Important additions

- Local-IO
  - Efficiency (move processing to data – via transparent migration)
  - Enable GPU-Direct in PNFS
  - Local compute node storage becomes managed and MUCH more useful
  - Enable access controlled pushdown in a trustworthy way (something Hadoop never had), and enables Apache Analytics ecosystem pushdown processing (example later)
- Client Direct-IO (lower cpu for BW and lower latency)
- Server Direct-IO (lower cpu for BW and lower latency)
  - Enables near serverless storage Open Flash Platform (OCP) efficiency gains (more later)
  - Massive performance wins, especially IOPS – desperately needed by AI industry
  - Efficiency gains, network expense/tier 1 storage costs lowered, reduced load on tier 1 storage tier, leverageable for any temporal wins – even cloud/distance/etc.
  - Enables named, access controlled, sharing of memory objects (via CXL/other shared memory mechanisms and Micron FAM-FS
- Referrals which allows for scaling metadata service in clumps (like simple Lustre distributed metadata feature)

Near future capabilities

- Multi-writers into one parallel file (heavy use in HPC) (N to 1)
  - Write cache controlled on a per file/open basis to eliminate false sharing
  - Most people don't know you can't write to an NFS file from multiple clients and expect to get expected results, even some solution providers ☹
  - There are few real fully featured Lustre alternatives for true at scale HPC, this is the primary feature that allows PNFS to cover that space
- Client erasure
  - Enables network style erasure eliminating server-side high availability fail over pair deployment need
  - Provides same style durability as cloud objects transparently in files
  - Enables 2 tier erasure, client/network and server side erasure which is only available in a very few settings (one object server, a couple cloud offerings, and LANL Campaign Store (10 years)
  - Enables scalable erasure on removable media – eliminates need for 2 copies on tape or any other media
- Combination of N to 1 and Client erasure
  - This has been the Achilles heal of client erasure – read-update-write ops from the client is expensive
  - Novel function ship of unaligned to block/erasure boundaries to server side produces a first time fundamental solution to this problem

Parallel NFS File Query Pushdown (Use of Local IO)

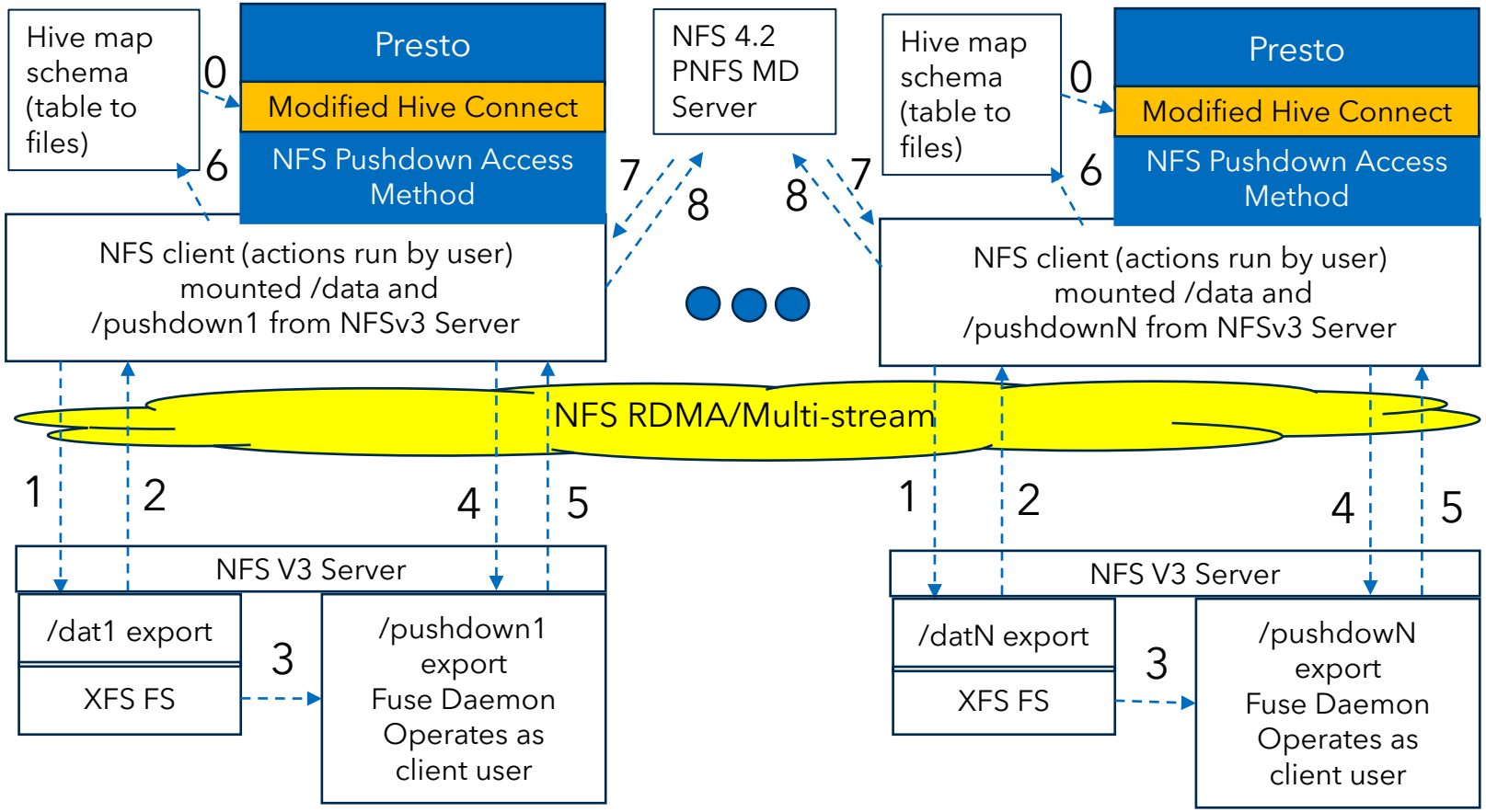
Mkdir /data/d1
7 mkdir to MD server
8 response

Read /data/d1/parq1
7 open to MD server 8 response
1 read to data server (via map)
2 results stream on handle
2 responses

Prep for analytics /data/d1
7 readdir to MD server
8 results stream on handle
8 responses
6 load hive table to file map and schema (include map location data server 1-N)

Query
0 read table to files from hive
4 open all /pushdownN/specialfile(s)
4 Write pushdown to handle(s)
(query on files from hive list) to each data server where there are files from hive
3 Fuse reads files locally (but through 4.2 local copy read so it an happen as user)
5 query results stream on handle from each
5 responses from each

Write /data/d1/parq1 to parqN
7 create to MD server 8 response
1 write data to data server directed by map from md server
(MDS will spread files to different data servers (no in file striping)



SDC | StorageAI™

PNFS Community building activities

- Testbeds – available and in use by PNFS industry partners
  - LANL scale up testbed (400-800-1600 Gbit networks, 160 clients each with local flash storage (for tier zero), Nearly 1 Tbyte/sec server side flash storage
  - CMU Parallel Data Lab (implemented using hand-me-downs (LANL, DOD, Jane Street) 64 node ethernet and 4-500 node Infiniband cluster with storage in every node, to try scaling out clients, data servers, metadata servers, tier zero
- Open source
  - PNFS file based solutions are all open source/linux distro provided for clients and data servers, but the metadata servers were all Proprietary (standards based)
  - Working with Peak:AIO on open source PNFS metadata server, based on BSD kernel NFSD
  - Launching effort to develop an example Linux Open Source User Space metadata and data service, to provide to community to enable innovation
  - Open Source Linux user space because (linux ubiquitous, kv/db etc. tools in user space, innovation easier, shared memory capabilities evolving are user space (see below)
- Enabling metadata scalability (a multi-decadal problem)
  - Working with Micron and SK hynix on shared memory (CXL/Celestica-Marvell/UALink) based technology
    - Provide common virtual address space across nodes sharing common memory
    - Provide coherence mechanisms, MPI/Shmem/other running on shared memory
    - Launching a distributed threads/mutex library (open source) to hopefully allow threads apps to become distributed threads with little or no work.
    - Working with Pometry (graph db/analytics/AI) on distributed tree traversal on shared memory

OFP Initiative Industry Momentum

"Power efficiency isn't optional; it's the only way to scale AI. Period. The Open Flash Platform removes the shackles of legacy storage, making it possible to store exabytes using less than 50 kilowatts, vs yesterday's megawatts. That's not incremental, it's radical."

- Hao Zhong, CEO & Co-Founder of ScaleFlux

"Flash will be the next driving force for the AI era. To unleash its full potential, storage systems must evolve. We believe that open and standards-based architectures like OFP can maximize the full potential of flash-based storage systems by significantly improving power efficiency and removing barriers to scale."

- Hoshik Kim, SVP, Head of Memory Systems Research, SK Hynix

"In AI, moving data fast is just as critical as storing it. Open architectures powered by advanced DPUs turn the network into an engine for agility and efficiency."

- Eric Vallone, VP Business Development, XSIGHT Labs

"Agility is everything for AI — and only open, standards-based storage keeps you free to adapt fast, control costs, and lower power use."

- Gary Grider, Director of HPC, Los Alamos National Lab (LANL)

"Open, standards-based solutions inevitably prevail. By delivering 10x greater capacity density and a 50% lower TCO, OFP accelerates that inevitability."

- David Flynn, Founder and CEO, Hammerspace

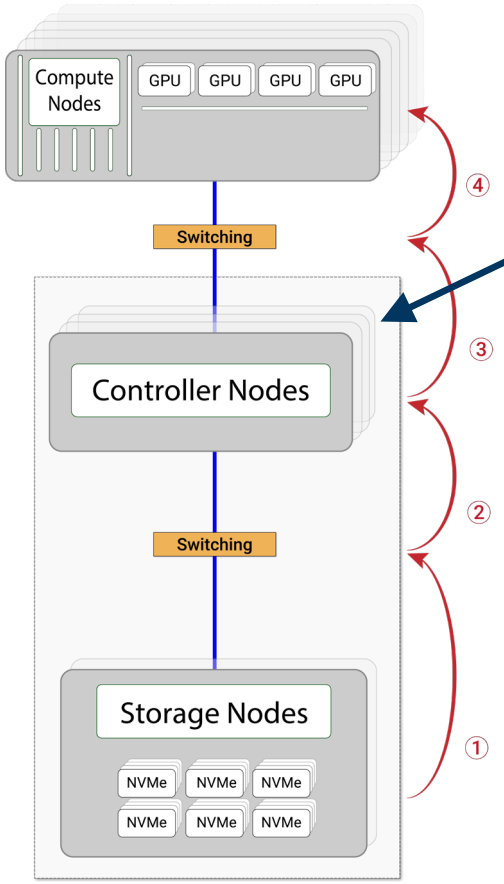
"pNFS inside a DPU makes efficient AI storage a reality: fully Open Flash Platform, Linux-native file storage that scales without adding power-hungry servers or complex network fabric."

- Trond Myklebust, NFS Client Kernel Maintainer

Courtesy OFP Consortium

OFP Introduces Standardized Data Storage for AI

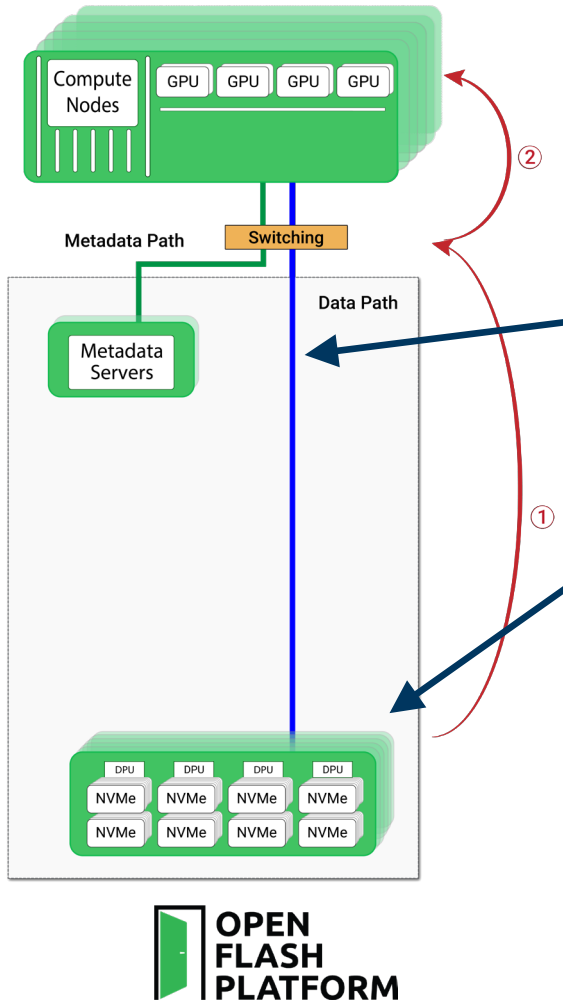
Closed Scale-Out NAS Architecture



Closed Proprietary Systems

- Data funnels through Controller Nodes.
- Puts data within a proprietary silo
  - Adds latency & degrades performance
  - Limits scalability
  - Adds power, cost and complexity.

Open Architecture



pNFS v4.2

- Separates Data Path from Control / Metadata
- Standard Linux – No proprietary client software or storage servers needed in the data path
- Ensures Extreme Linear Scalability

OFP

>10x Capacity Increase

60% Cost/Capacity Reduction

90% Power Savings

OPEN FLASH PLATFORM

Courtesy OFP Consortium

SDC | StorageAI™

The Goals and Leverage of the OFP Initiative

OCP V3 Rack

252	OFP Sleds
1	EB Capacity
200	Tbps Bandwidth
25	kW Idle Power
40	PB per kW
8	Tbps per kW

10X
Or Greater Increase in Density

1 Exabyte / Rack

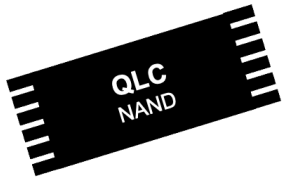
90%
Decrease in Power

60%
Longer Operating Life

66%
Lower TCO

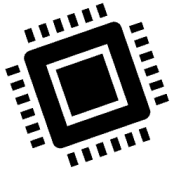
Large Capacity Flash

The wide adoption of QLC as a capacity-centric tier has further driven down the cost of capacity on flash.



Powerful DPUs

DPUs are now powerful enough to run a standard Linux OS and handle storage I/O with a much smaller power and cooling footprint than CPUs.

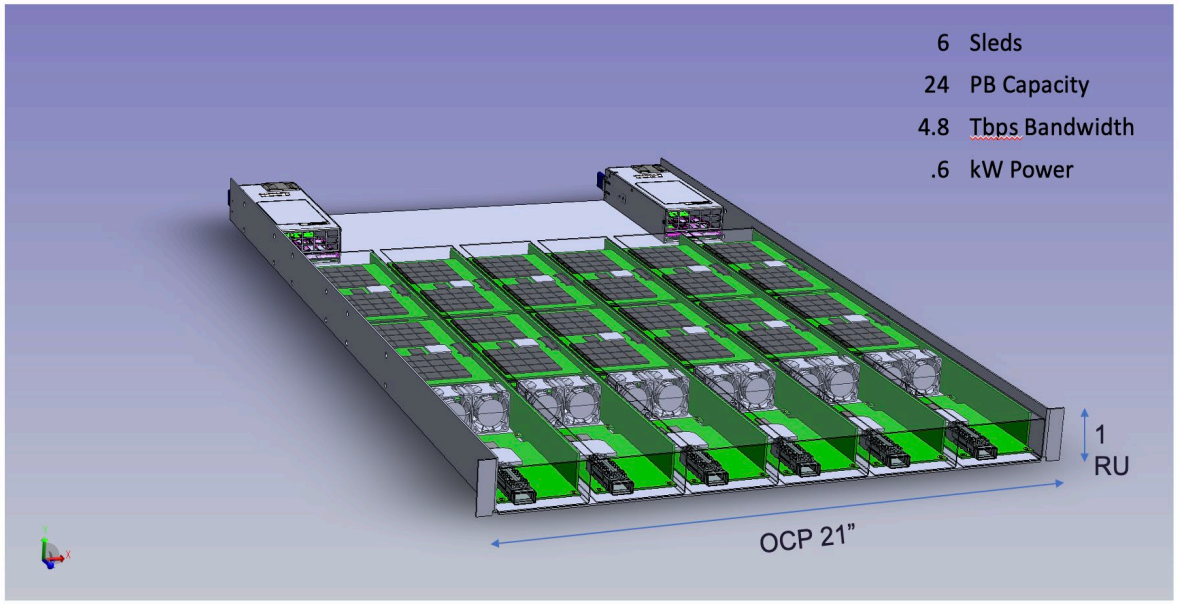


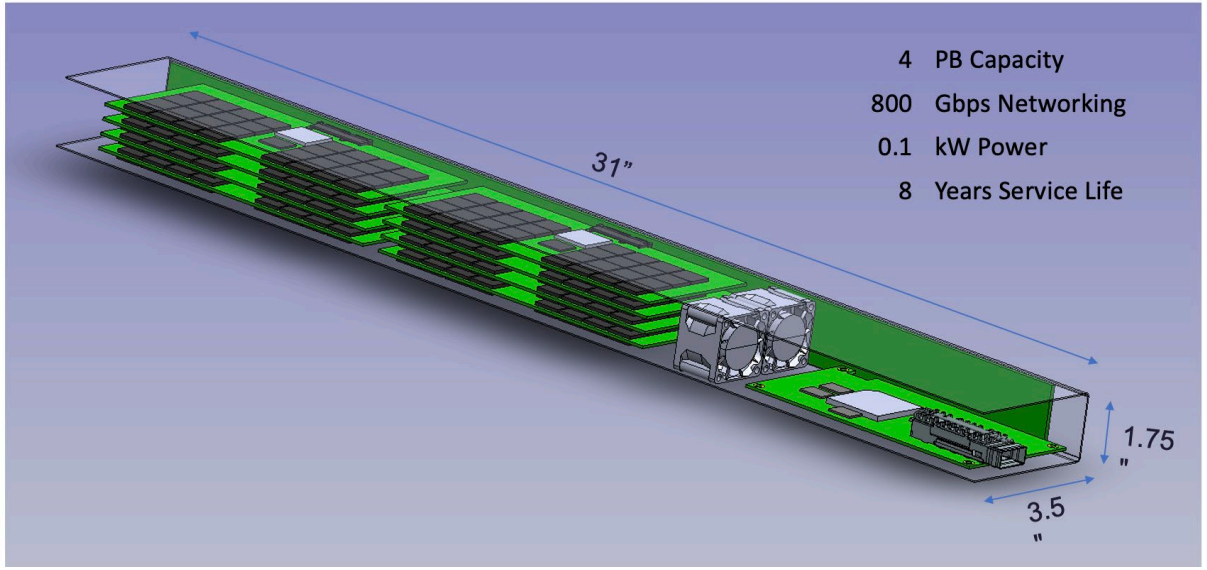
Recent Linux Enhancements

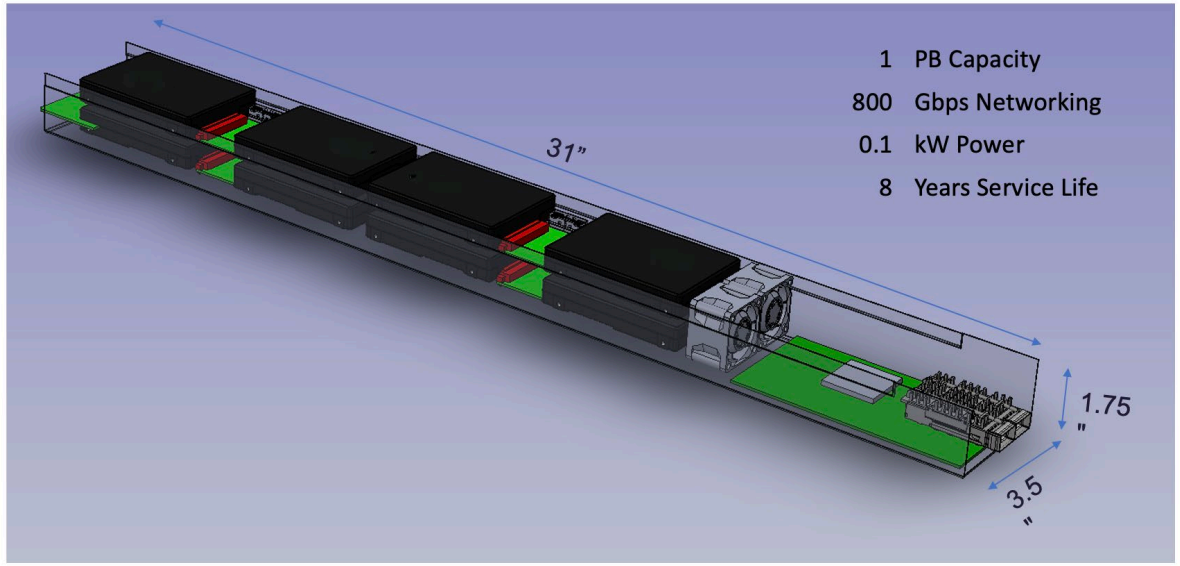
The updates to NFSv4.2 within Linux include the pNFS client and Flex Files, which means a powerful parallel filesystem is already available in data centers worldwide.

Courtesy OFP Consortium

U2/U2E OCP OFP







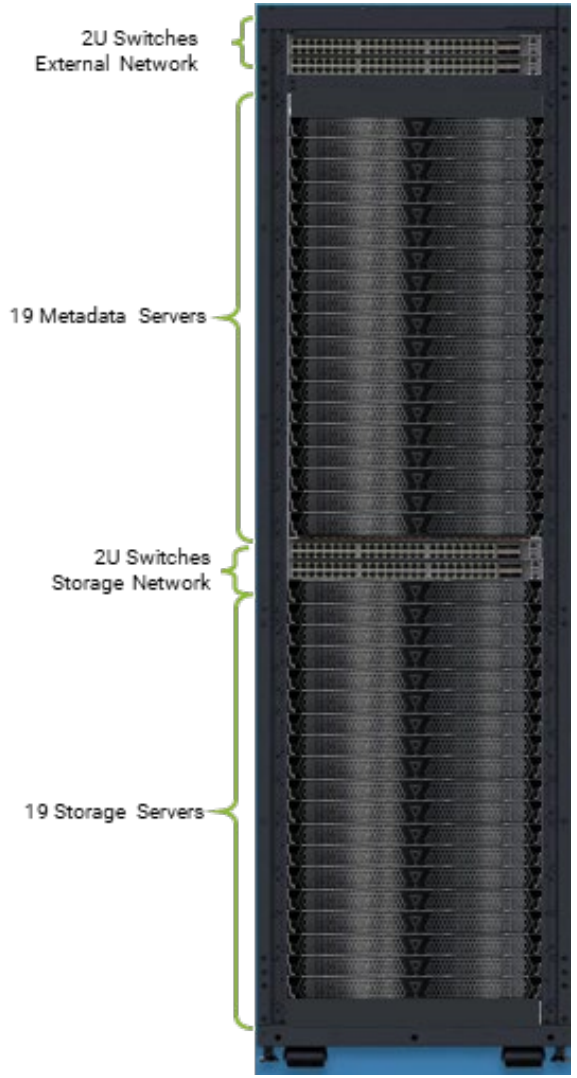
Courtesy OFP Consortium

Open Flash Platform vs All-Flash Scale-Out NAS

50.16PB
capacity

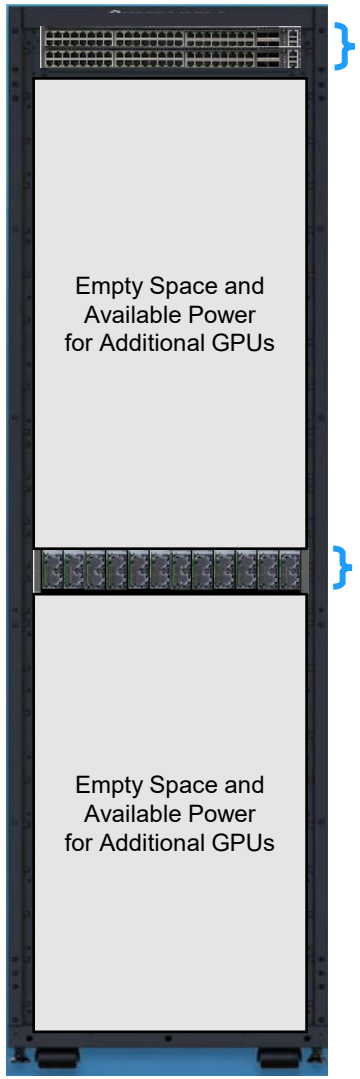
40
Rack Units

29.1kW
power



Greater than
10x
density

96%
power savings



48PB
capacity

2
Rack Units

1.2kW
power

Courtesy OFP Consortium

Removing the Storage Server Improves Efficiency and Reliability

Storage servers lock data into silos and consume needless resources

OFP uses flash as a commodity, with standard, open Linux features and a streamlined data path.

1 | No Dedicated Server Layer

With OFP, intelligence (a DPU/IPU running standard Linux and NFS) is embedded directly within the flash storage sled. There's no separate, bulky, proprietary server box to buy, power, cool, or manage for the basic task of serving data.

2 | Standard, Open Software

Each OFP sled serves data using industry-standard NFS running on Linux. You're not paying a "proprietary software tax" for basic data access at the sled level.

3 | Direct Data Path

Leveraging standard pNFSv4.2 with Flex Files, the path from a client to an individual OFP sled is streamlined, bypassing the heavy processing of a traditional NAS head. This enables extreme scale-out for AI workloads.

Courtesy OFP Consortium

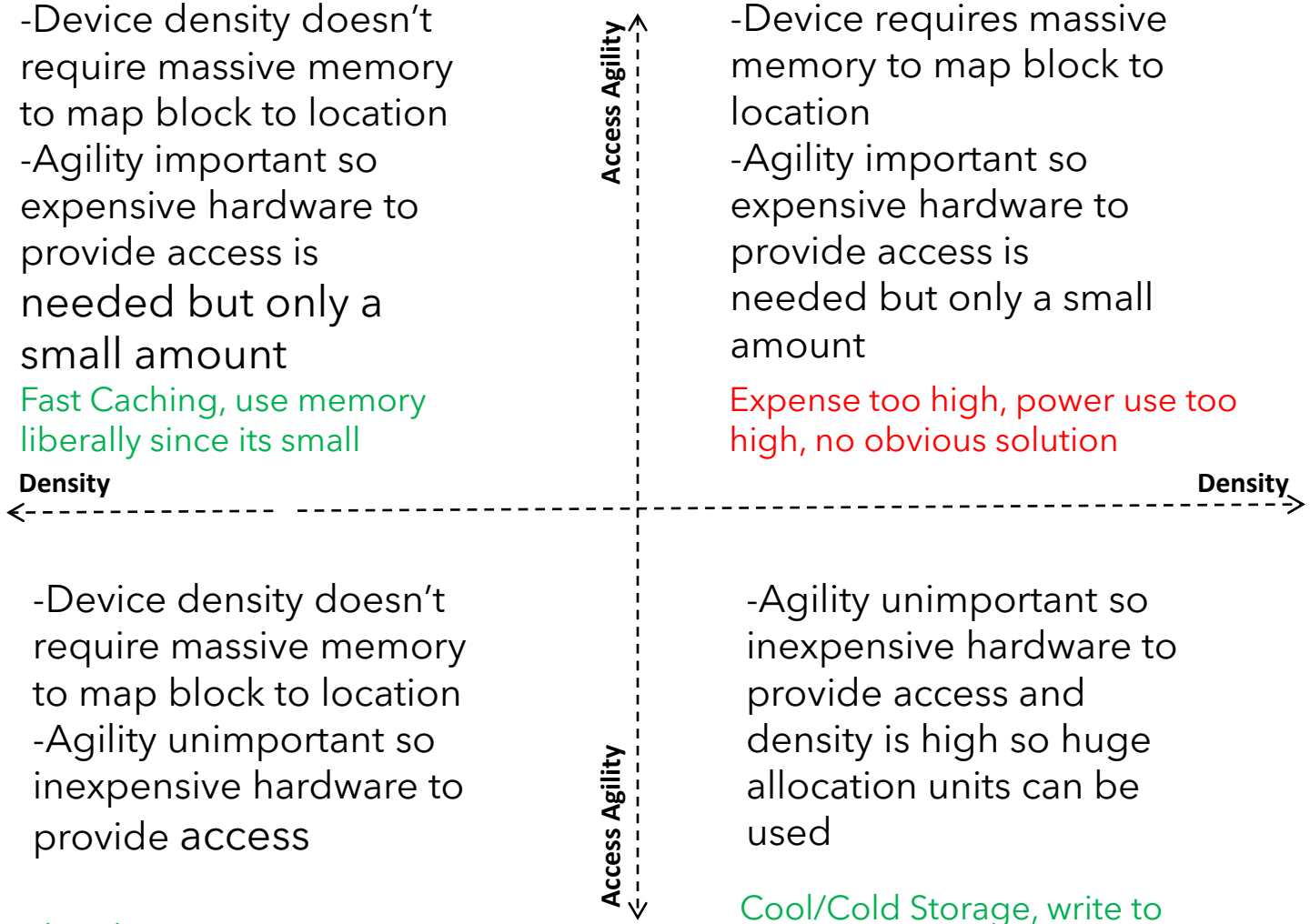
The Age-Old Storage Conundrum

-Device density doesn't require massive memory to map block to location
-Agility important so expensive hardware to provide access is needed but only a small amount

Fast Caching, use memory liberally since its small

-Device requires massive memory to map block to location
-Agility important so expensive hardware to provide access is needed but only a small amount

Expense too high, power use too high, no obvious solution



-Device density doesn't require massive memory to map block to location
-Agility unimportant so inexpensive hardware to provide access

Thumb Drive, etc.

-Agility unimportant so inexpensive hardware to provide access and density is high so huge allocation units can be used

Cool/Cold Storage, write to slc or cmr, move to qlc/smr/hamr

Classical HPC Benefitted from Upper Left and Lower Right due to most accesses being large 100k->
Classical HPC Benefitted from high BW but got that through many device parallelism

AI has needs for high BW, small/agile access in mass parallel, and extreme amounts of randomly accessed data (density)

Will we have to borrow from our memory management experience:

- **Fine grained access to huge pages**
- **Put more in the hands of the application**
- **Better living through Data Structures (Quad trees/Oct trees, Hilbert, etc.?)**


SDC | StorageAI™
A SNIA Event
Thank You