Los Alamos NATIONAL LABORATORY

Evolving the I/O Stack: The Case for Computational Storage at LANL

Qing Zheng, Scientist, HPC Storage R&D

6/10/2025

LA-UR-25-25190

HPC at LANL

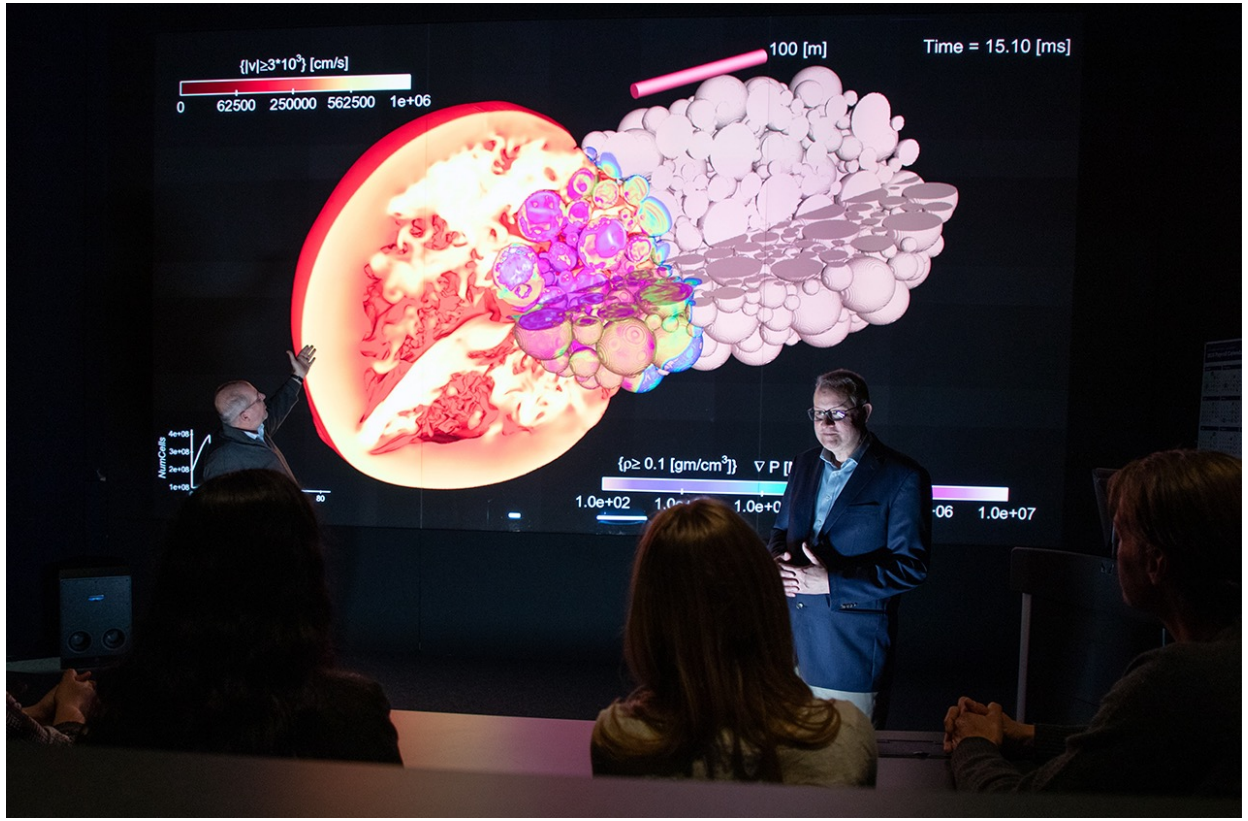
Decades of weapons computing support to keep the nation safe

- LANL responsible for 4 of the 7 types of nuclear weapons in the current stockpile
  - Simulations to analyze weapons aging and performance

Cutting-edge technology enables large, long-running, multi-physics, 3D simulations

- Jobs can last for months running on 80% of the machine
- Drives value across domains
  - Wildfires, rising seas, asteroids, and more

Example: Planetary Defense



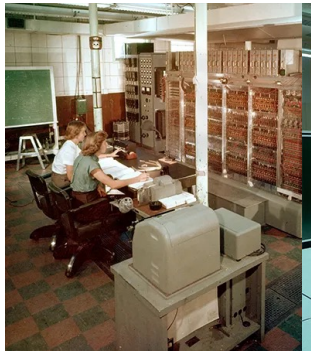
Scientists simulate a megaton blast to nudge an asteroid off its collision course with Earth.

Operation Crossroads Back 1946

We can no longer do things like this since the 1992 moratorium

Platforms Over The Years

1952 1976 1992 1998 2003 2008 2010 2016


MANIAC


Cray-1


CM-5


Blue Mt


Q


Roadrunner


Cielo

Trinity

Crossroads

6144 Nodes
- Intel Sapphire Rapids
- 128 GB (HBM2e)

768 TiB of memory
- No DDR

HPE Cray Slingshot
- Dragonfly

~10PB NVMe storage
- ~1TB/s BW
- Lustre FS

Beyond the Model: Scaling Analysis with Simulation

Unprecedented detail fuels discovery—and explosive data growth

- Per timestep: 10s of TBs to 1 PB
- Per run: Hundreds to 10,000+ timesteps

It's not about how much data you generate

- Big data is powerful—but only if it can be analyzed efficiently

LANL is actively exploring computational storage

- Part of our broader strategy to modernize data workflows on our next-generation HPC systems

What’s Computational Storage

Storage servers doing something for the user/system applications

Accelerators within storage devices

Accelerators alongside storage devices

Accelerators within the network

Why Computational Storage?

Adaptable compute placement—host, network, storage

- Smart devices offer flexibility as costs and technologies evolve

Selective data access

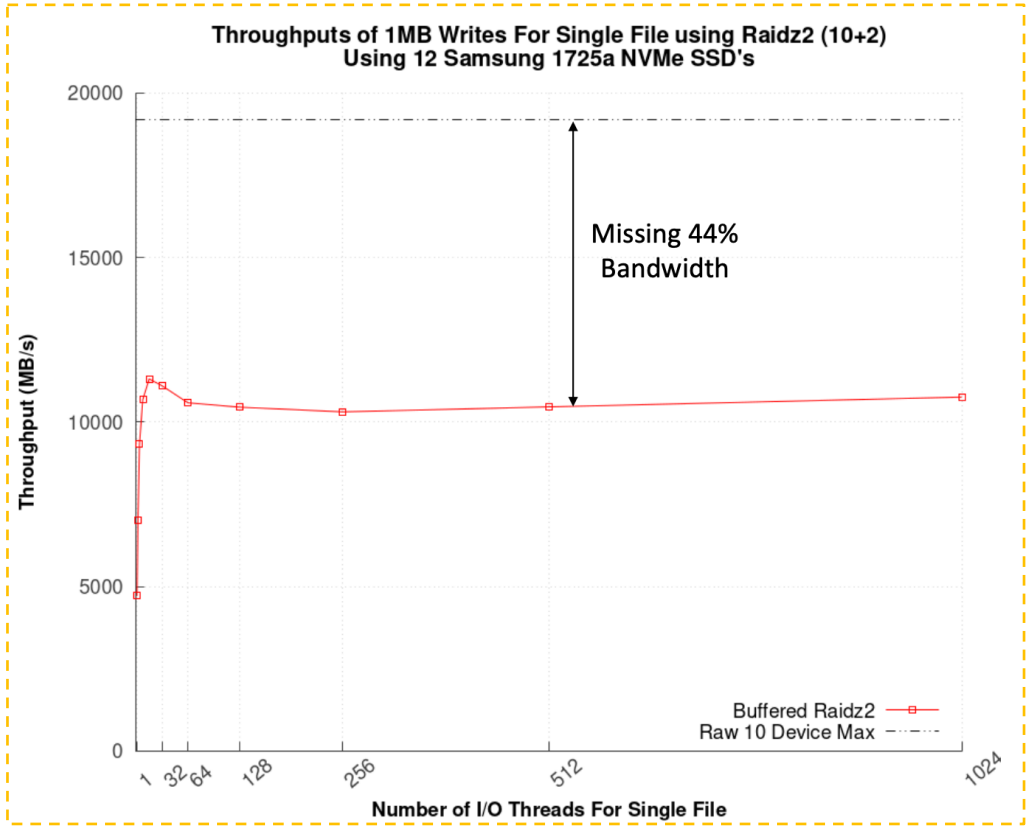
- Analytics often need far less data than was written—if you can find what matters
  - Save 80% of total analysis time
- Relax analysis memory needs
  - Allow large-scale analysis from modest system

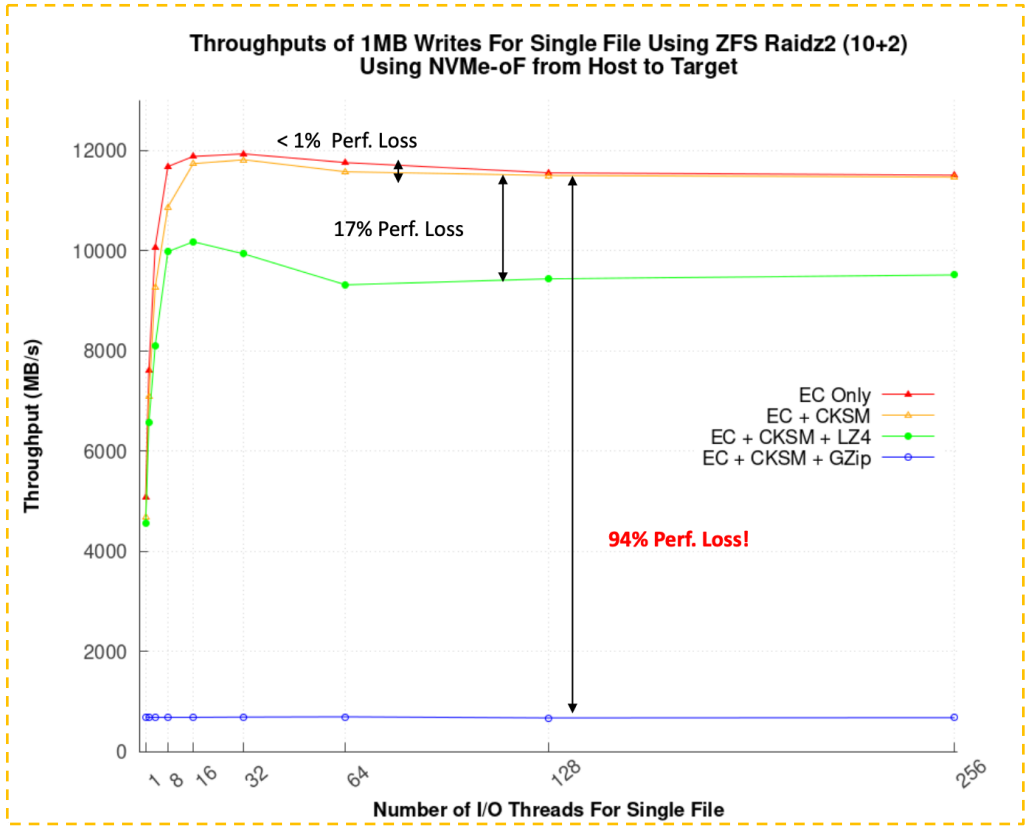
Flexible offload strategy

- Leverage both data-agnostic and data-aware offloads for broad applicability

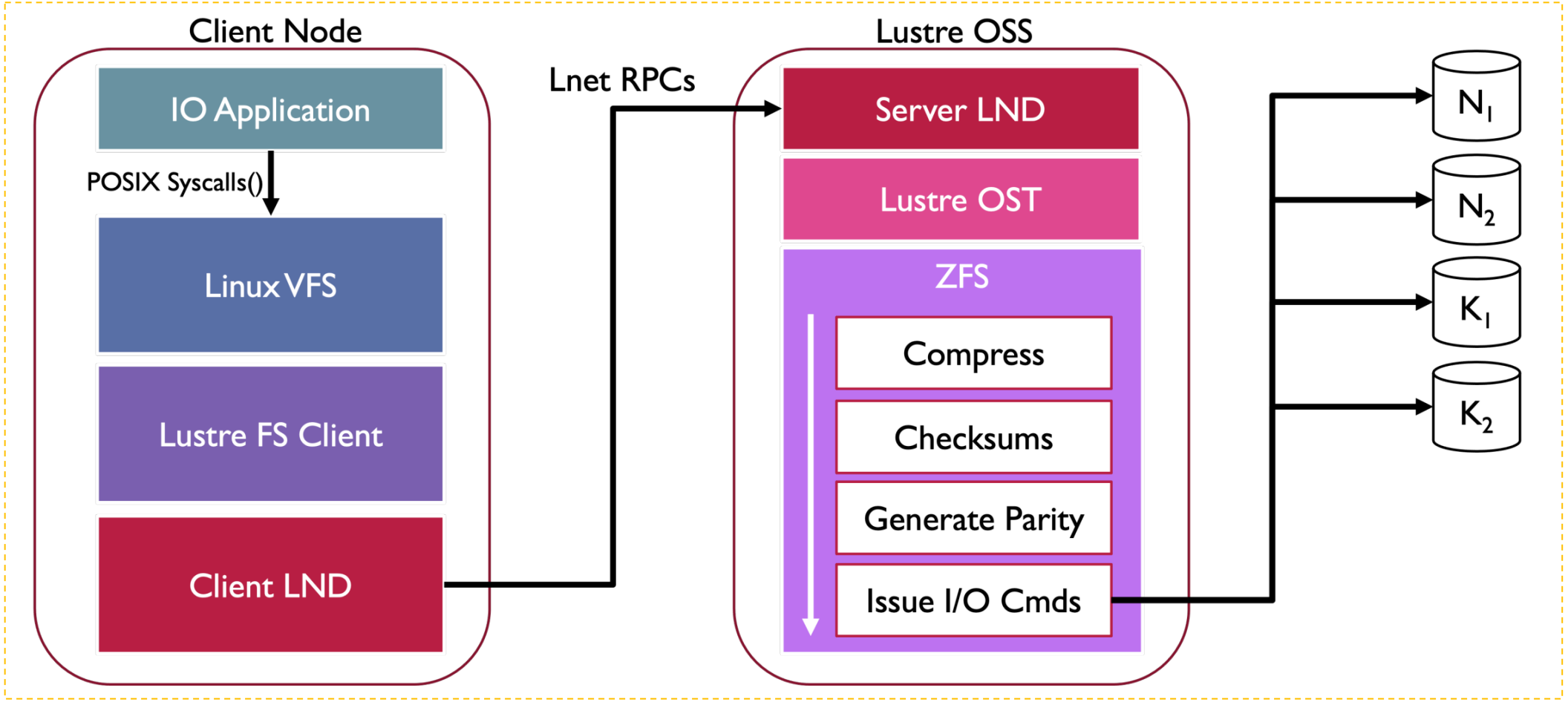
Data Agnostic Offloads

Problem: Limited server memory bandwidth makes multi-pass processing over large data streams inefficient





Lustre/ZFS IO Stack



ABOF: Accelerated Box of Flash

Offload compression to hardware isn't something new but doing it in a very consumable way takes efforts

Co-design:

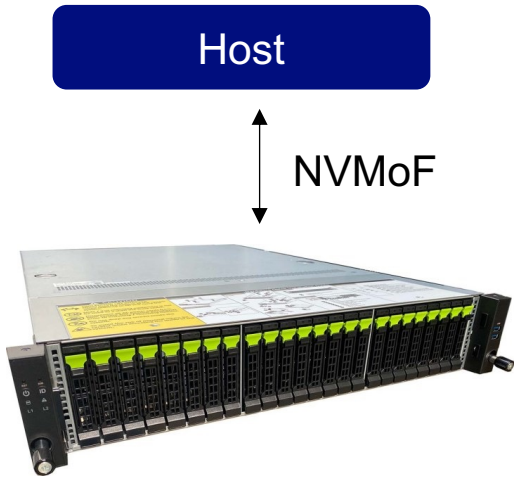
LANL: ZFS Direct I/O + ZIA (ZFS Interface for Acceleration) + DPUSM (Data Processing Unit Service Module) kernel module

Eideticom: Custom NVMe target in Linux kernel Accelerator technology

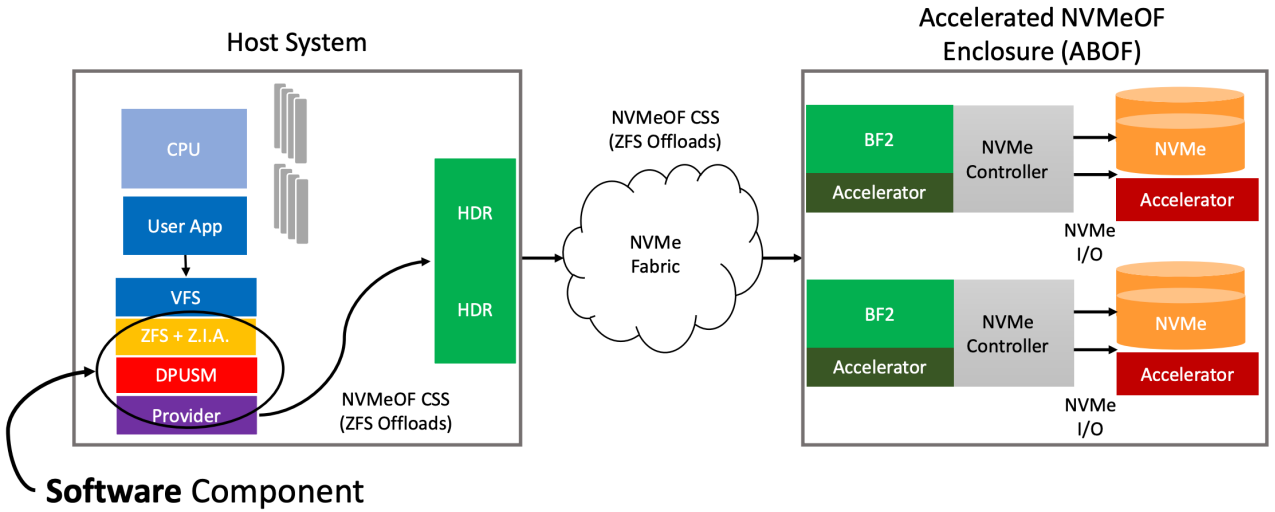
Aeon Computing: Enclosure design

Nvidia: BlueField-2

SK hynix: Flash drives

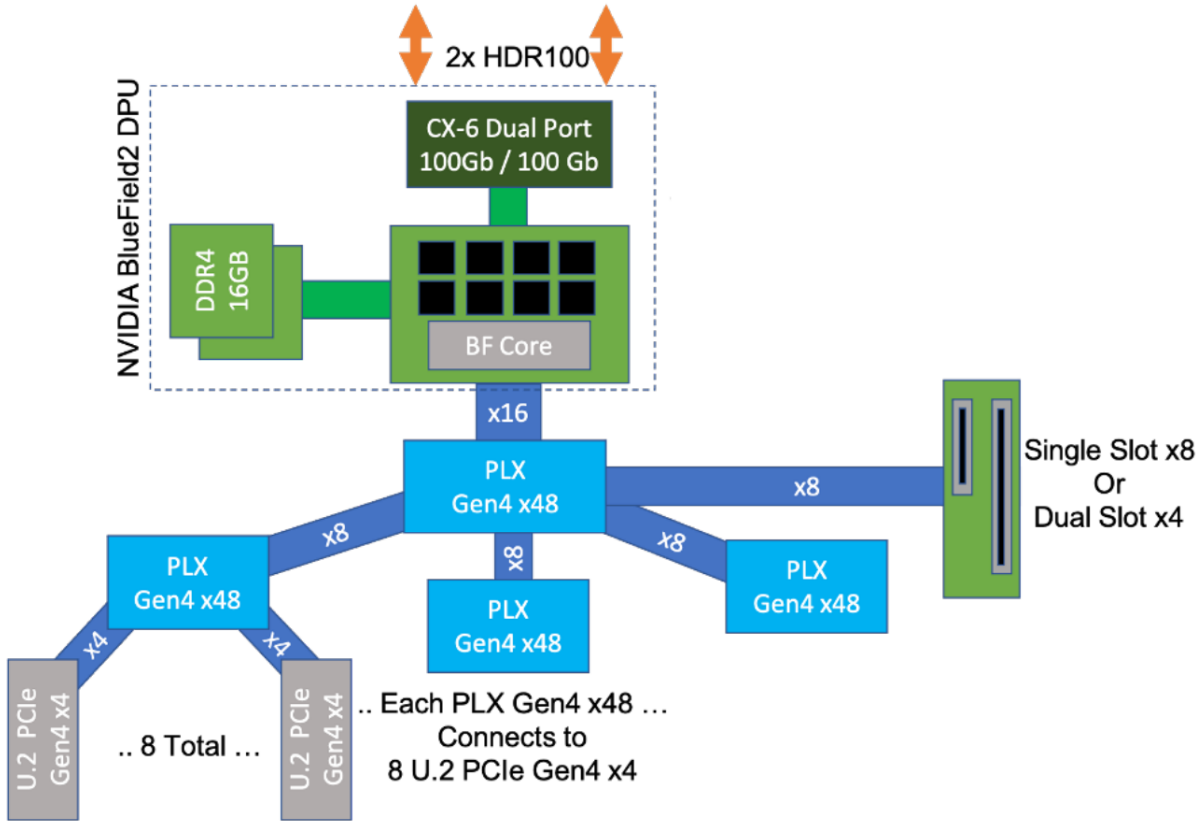


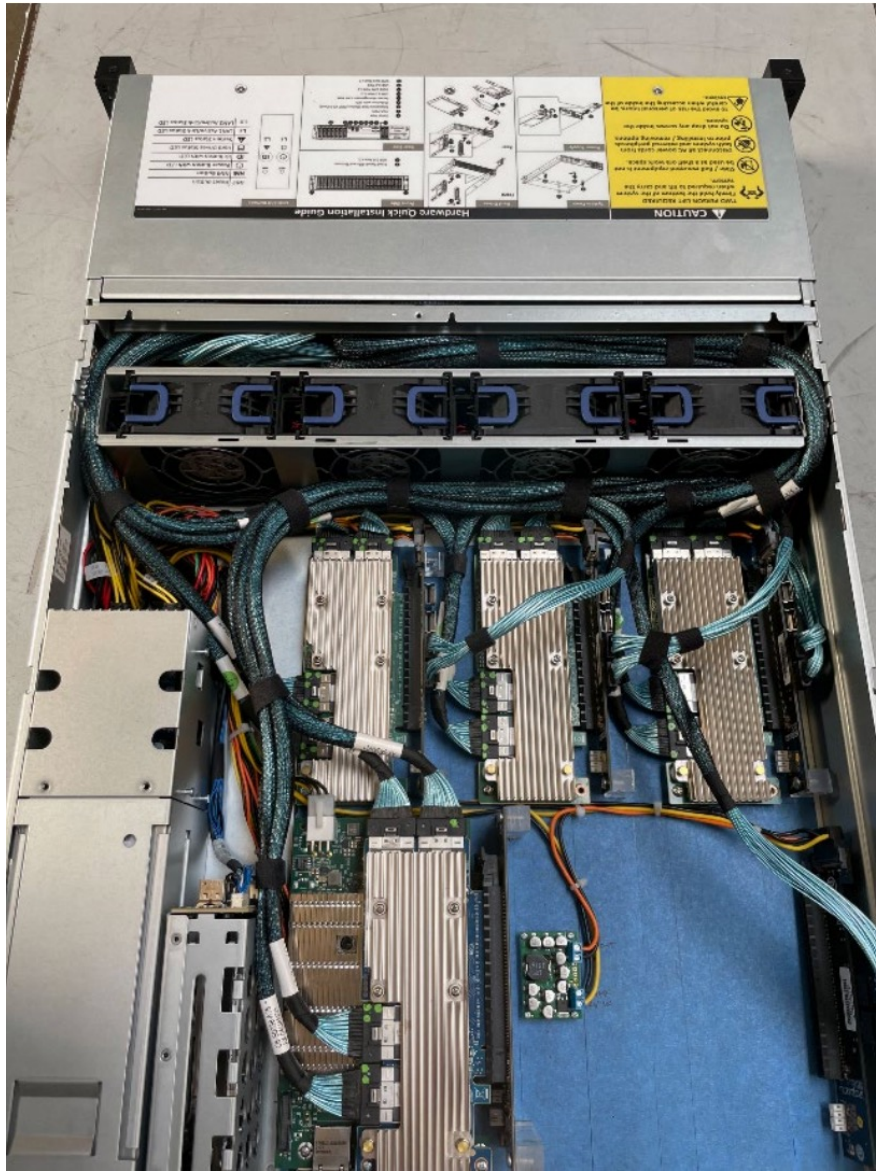
Accelerated Box of Flashes



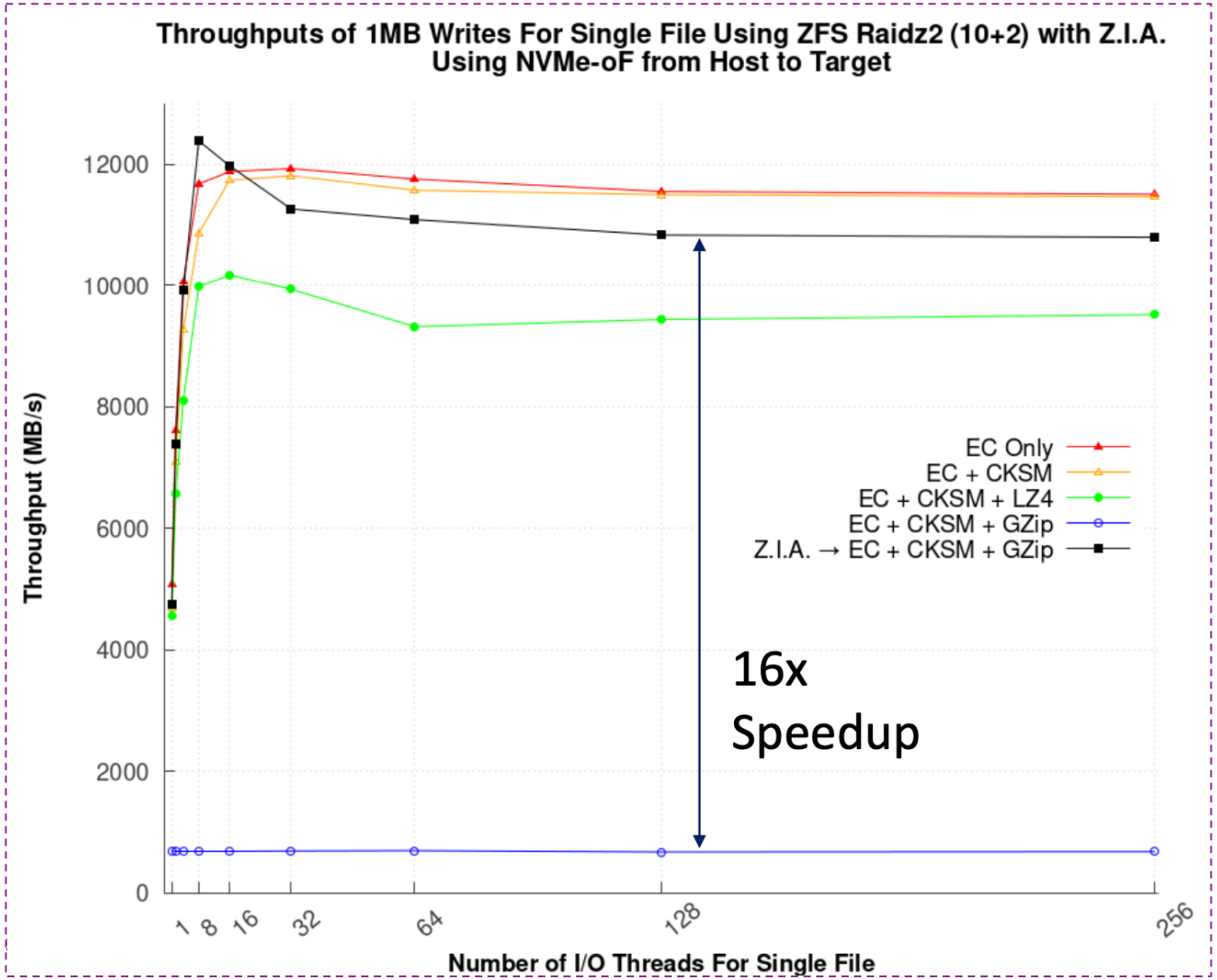
Software fallback

Another Look At It



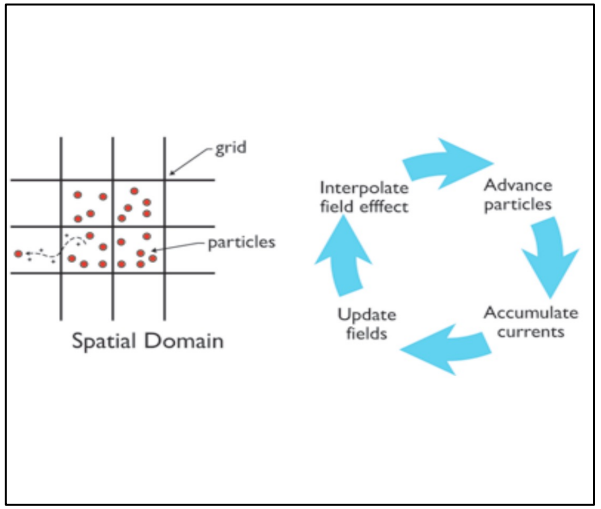


ABOF Results



Data Aware Offload for PIC Code

Can we build indexes as data is written at a very small cost to win big during analysis?



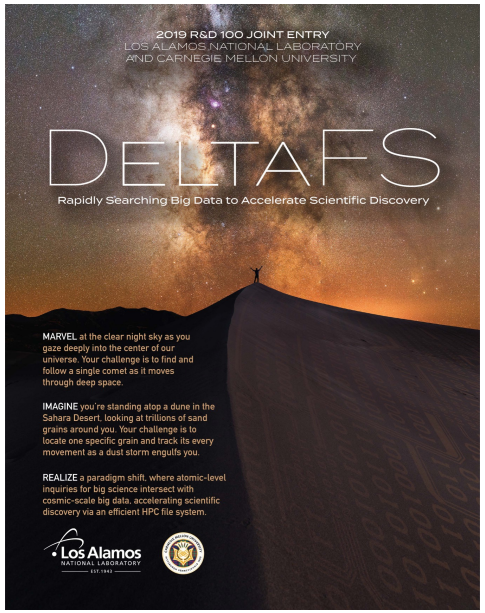
Problem: Identifying outlier particles often requires scientists to sift through trillions of records.

DeltaFS: on-the-fly data indexing for 1 primary index

- Point queries only (5000x faster)

KV-CSD: hardware-accelerated on-the-fly data indexing for 1 primary and many secondary indexes

- Point and range queries
- RocksDB-like interface (very consumable)



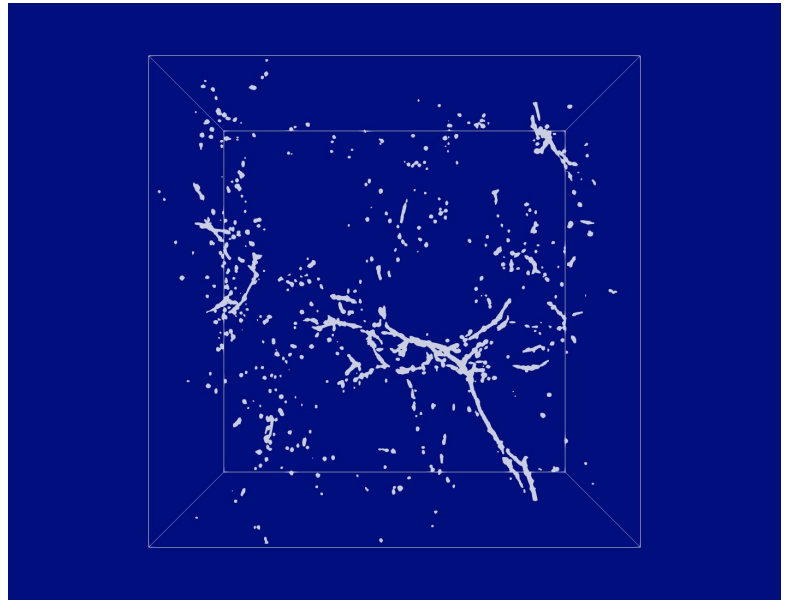
Data Aware Offload for Grid-Based Codes

Grid code features **columnar** data storage and analysis

- Each column represents temperature, pressure, density, …
- **Problem**: Analysis often done with Viz pipelines loading entire columns unnecessarily

Selectively retrieve only relevant data through composing/extending tools from the big data/analytics community

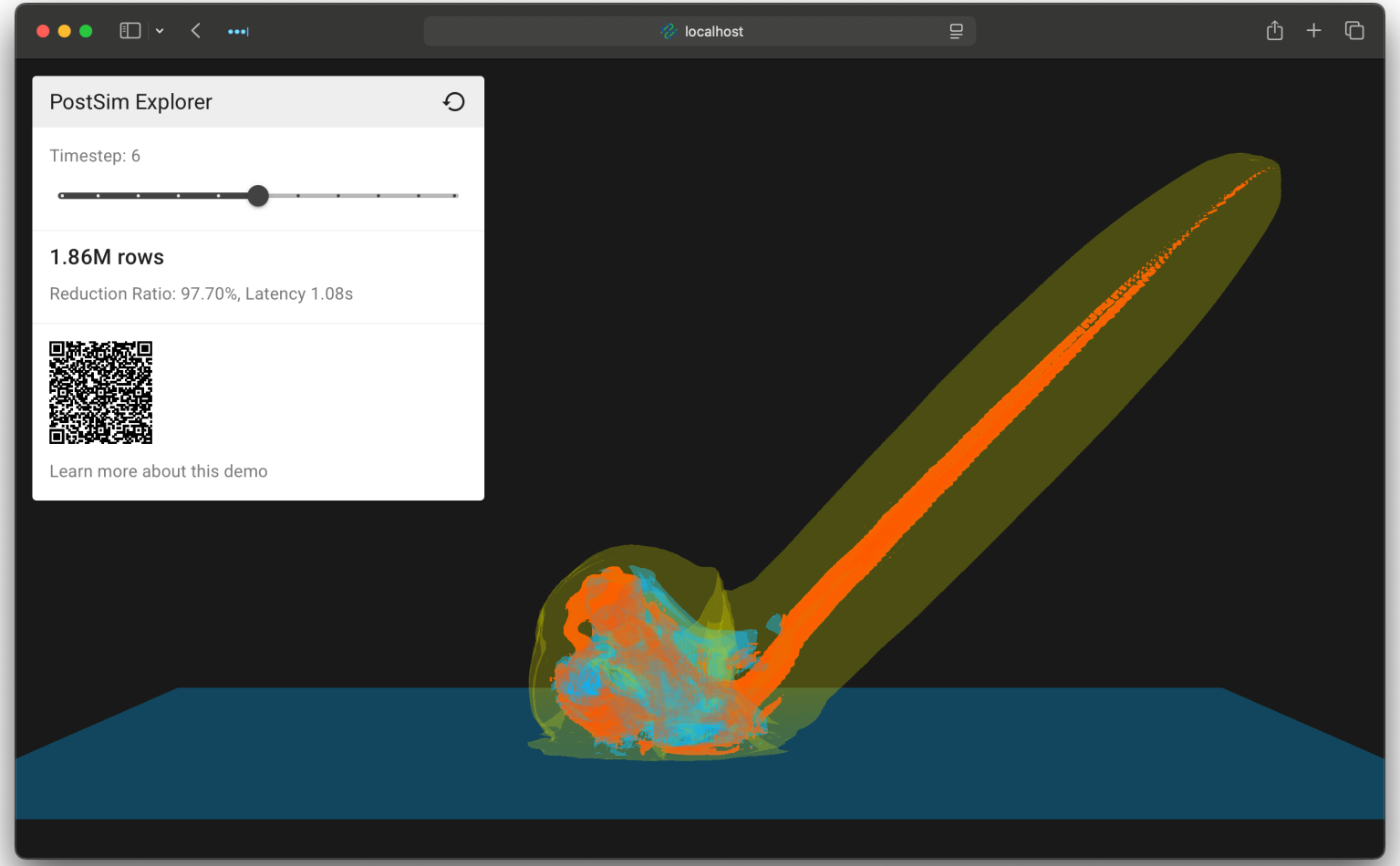
- Distributed SQL engine: Presto, Apache Spark, Apache DataFusion
- SQL IR (Intermediate Representation): Substrait
- Embedded SQL engine: DuckDB
- Popular data formats: Apache Parquet, Apache Arrow
- Open storage : pNFS



A contour over baryon density for regions of candidate halos in a cosmological hydrodynamics simulation (**0.06% data needed**)

Outcome: a standard data reduction function in-storage

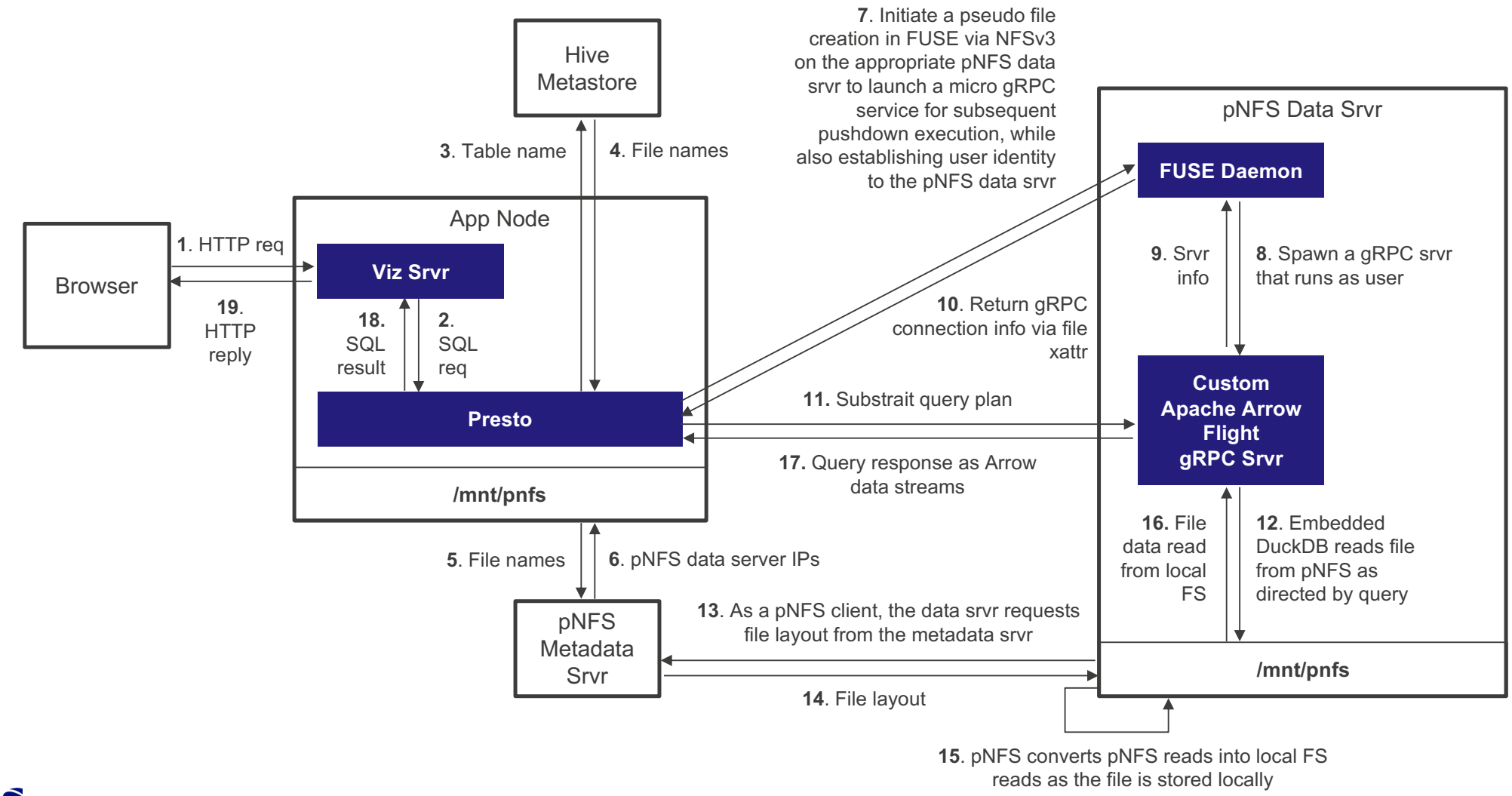
ISC-HPC 2025 Demo



The QR code links to a writeup that dives into the demo, if only photos are allowed in this conference

Keep an eye on LANL/Hammerspace/SK hynix's social media channels

Open Pushdown Architecture



Lessons and Future Direction

Compute-near-storage works best with open, structured formats

- LANL is looking at transitioning from legacy formats to modern analysis-friendly formats (Parquet, Arrow)

Open, consumable APIs are critical (e.g., ZFS, RocksDB-like, Apache ecosystem)

Future work:

- Push erasure coding to clients (new NFS standard in progress)
- Full demo planned for Flash Memory Summit/Supercomputing
- Continue working with partners: SK hynix, Hammerspace, AirMettle, Pometry, CMU, etc.

Thank you!