LA-UR-25-26338

Bringing Analytics to the Data: In-Storage Computing for pNFS

Qing Zheng, Scientist, Los Alamos National Lab (LANL)

qzheng@lanl.gov

FMS the Future of Memory and Storage

The Challenge of Scientific Data at Scale

LANL simulations (wildfires, rising seas, high-energy particles) are among the world's most complex

- PBs of data per timestep with 1000s of timesteps

Insight comes from analysis, not just simulation

- Analyzing these massive datasets is becoming a bottleneck

LANL is exploring computational storage

- Part of our broader push to modernize I/O and storage at scale

Why Computational Storage

Selective data access

- Many queries need <1% of data (e.g., wildfire front)
  - Today's tools often read the entire dataset—this doesn't scale
- Loading full datasets demands massive memory on compute nodes, limiting where analysis can run

Adaptable compute placement—host, network, storage

- Computational storage lets us assign compute tasks where they run best, as costs and technologies evolve

LANL’s Compute-Near-Storage Journey

ABOF (Accelerated Box of Flash)—our prototype for data-agnostic acceleration

- Use ZFS plugins for in-line compression at device speed
  - Introduce ZFS Interface for Accelerators (ZIA)
    - Allow ZFS integration with ABOF and other techs like Intel QAT, MaxLinear, …
    - Offload compression, checksumming, parity, and more

Data-aware offloads (the lab’s more recent focus)

- Leverage pNFS and the Apache big data ecosystem for an open, deployable analysis pushdown architecture (this effort)
  - Enable selectively reading only what’s necessary

A Standards-Based Architecture

Data & Query Layers

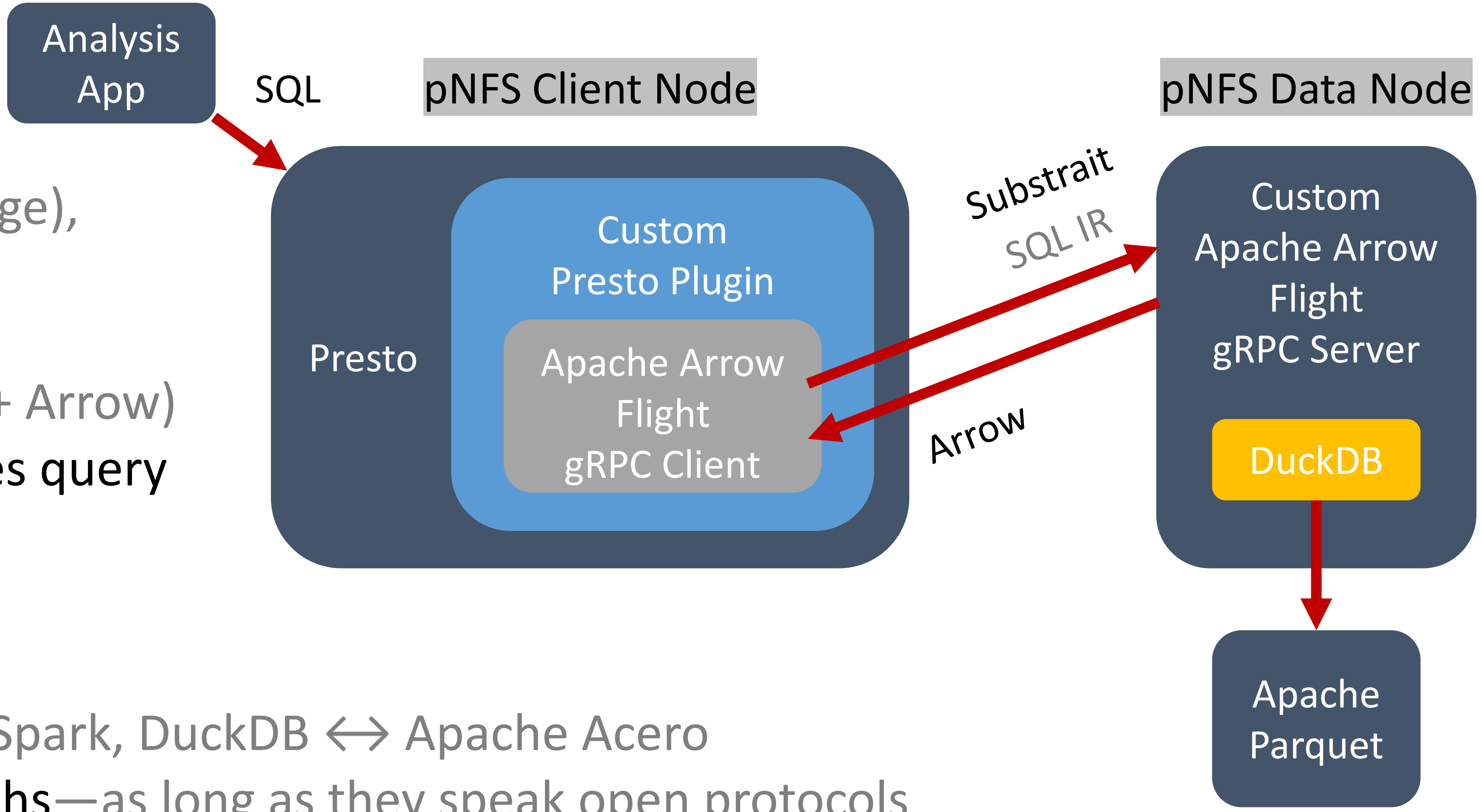
- Storage: Parquet
- Query: Substrait
- Execution: DuckDB (in-storage), Presto (aggregation)

Communication

- Apache Arrow Flight (gRPC + Arrow)
- pNFS layout metadata guides query routing

Modular Design

- Components are swappable
  - e.g., Presto ↔ Apache Spark, DuckDB ↔ Apache Acero
- Easy to plug in emerging techs—as long as they speak open protocols



Secure, Transparent Access to Data

Standard permission checks

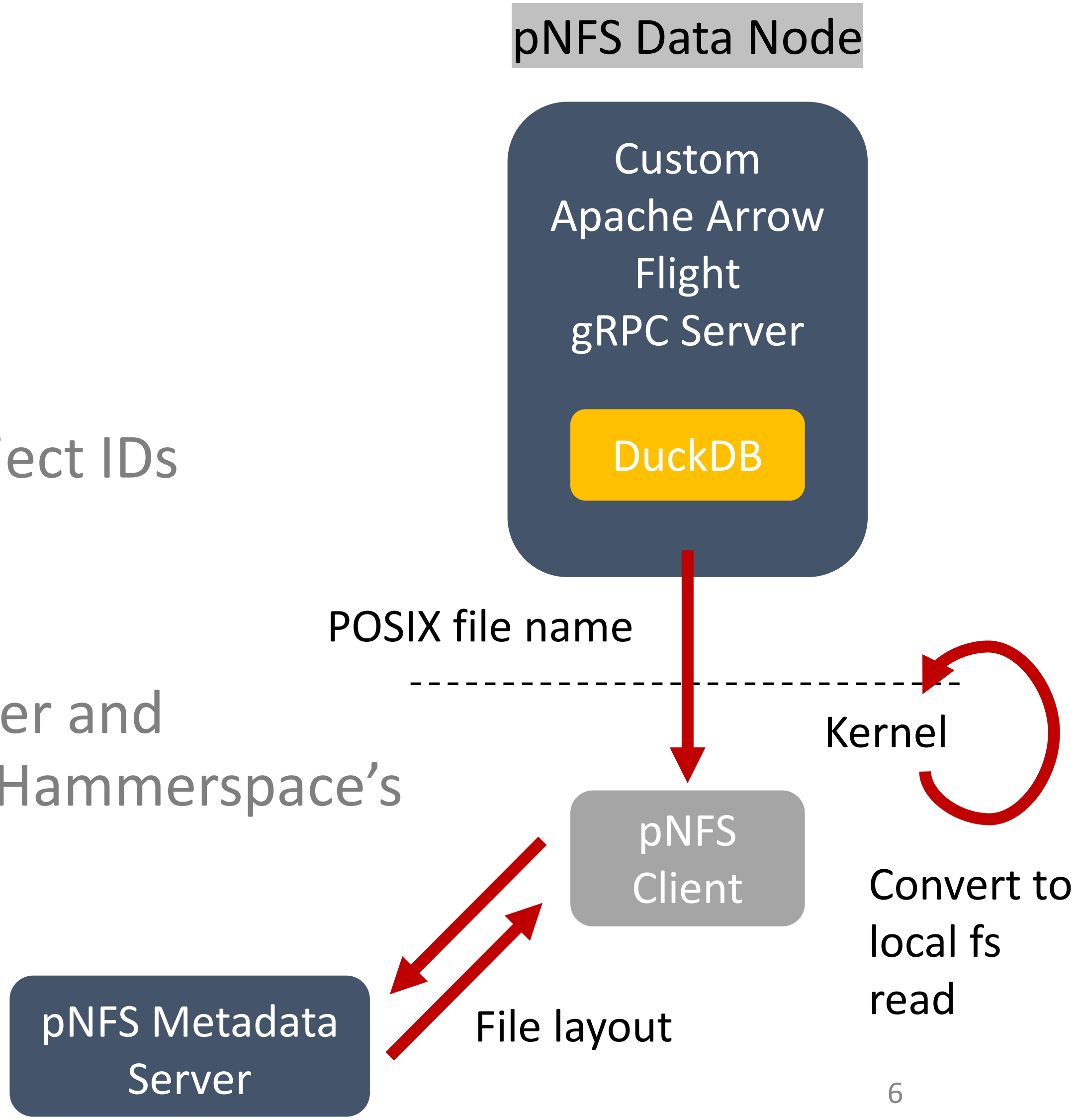
- gRPC server runs as the end user, not root

No exposure of internal mappings

- Queries use POSIX file names—not internal object IDs

Efficient data access

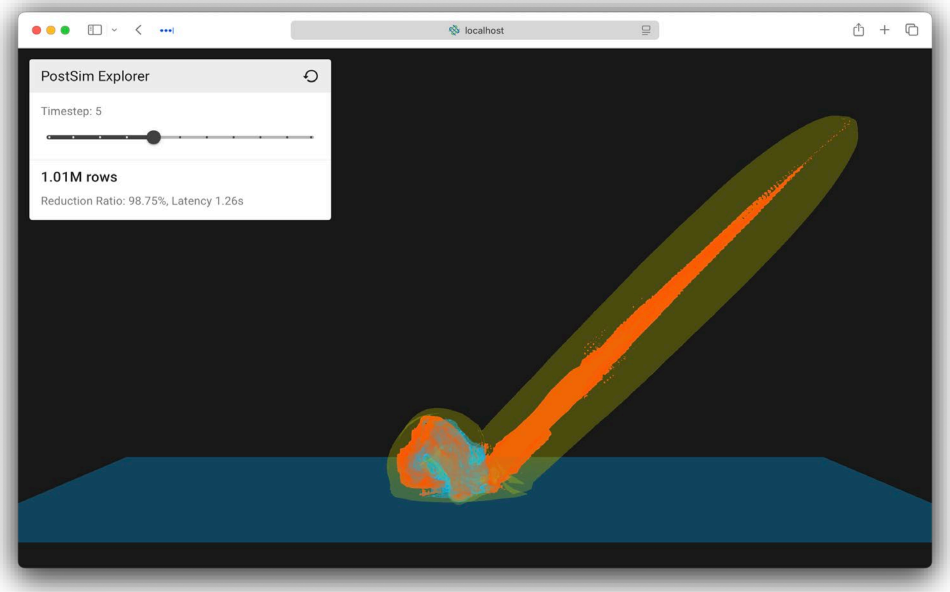
- pNFS data server self-identifies as storage holder and transparently performs local reads—thanks to Hammerspace's recent Linux kernel update



Real-World Impact and Demos

HAMMERSPACE

ISC-HPC 2025



PostSim Explorer

Timestep: 5

1.01M rows

Reduction Ratio: 98.75%, Latency 1.26s

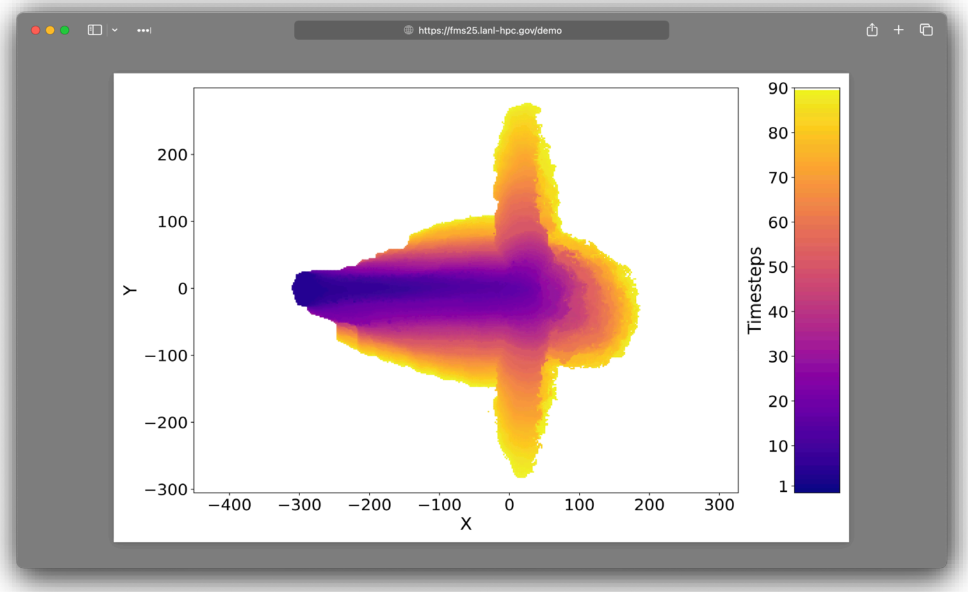
Asteroid-ocean impact analysis

Integration with standard HPC analysis tools

Up to 99% less data movement

SK hynix

FMS 2025



Wildfire spread analysis

Object-based computational storage

Multi-layer query processing

Conclusions

In-storage analysis is most effective with

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

Open, standards-based stacks enable real deployment

Future work

- Continue working with our great partners: Hammerspace, SK, …
- Deeper integration with scientific software
  - E.g., viz contour offload
- Client-driven erasure coding and N-1 writing in pNFS