Run Queries Where the Data Lives With pNFS

ISC-HPC 2025

Introduction

From rising sea levels and raging wildfires to asteroid impacts as devastating as the one that ended the age of dinosaurs, Los Alamos National Laboratory runs some of the world's largest and most sophisticated simulations. These advanced, multi-physics models can run for months at a time, harnessing hundreds of thousands of processor cores to help scientists explore extreme scenarios, measure global risks, and strengthen our resilience against future threats.

When the Data's Too Big to Use

Thanks to today's supercomputers, researchers can create simulations that are not only incredibly detailed but also extraordinarily demanding. The scale of data they produce is staggering: a single timestep can generate anywhere from tens of terabytes to a full petabyte, with total simulations spanning hundreds—or even tens of thousands—of timesteps. But while these models are powerful tools for understanding complex systems, their true value is only realized when the data they generate can be efficiently managed, analyzed, and understood.

With current systems, turning raw simulation data into usable insight can be extremely expensive. It requires moving the entire dataset and analyzing it on platforms with enough memory to hold at least one full timestep—which can be as much as a petabyte. That's more than many systems can handle, which makes it tough to do analysis outside of the biggest machines.

But bigger isn't always necessary. Depending on the analysis, scientists may be interested in only a small portion of a much larger dataset—such as the forefront of a wildfire or the leading edge of a shockwave. In these cases, the most valuable insights are often buried within just a tiny fraction of the total simulation output—sometimes several orders of magnitude smaller than the full dataset.

Toward Selective Data Access in Scientific Workflows

Los Alamos has been keenly interested in smarter ways to navigate simulation output—approaches that reduce how much data needs to be moved, processed, or even read at all. In an earlier effort, Los Alamos and SK hynix co-developed KV-CSD—a hardware-accelerated key-value store that enables fast, indexed access in Particle-in-Cell simulations, where scientists often search for particles with unusual feature vectors—a task that previously required scanning billions or even trillions of data records.

Now, in collaboration with Hammerspace and SK hynix, the lab is extending its data-reduction efforts to its grid-based simulations. These codes organize data in a columnar fashion, with each column representing a specific attribute—such as temperature, density, or pressure. Scientists typically analyze this data using visualization pipelines—built from readers, filters, and renderers—that convert raw simulation output into interactive 3D visuals to support result interpretation and insight into the underlying physical phenomena.

Today, the entire pipeline runs on compute nodes—separate from storage. As a result, analysis begins by transferring the full dataset from storage to compute node memory, where filters discard irrelevant portions and transform the rest for on-screen rendering. This creates bottlenecks—not only due to unnecessary data movement, but also from high memory demand, as compute nodes must hold the entire dataset even though much of it may be discarded by filters later.

Query Pushdown Architecture for pNFS-based Storage

Built on standard pNFS protocols and open-source tools from the big data and analytics ecosystem, this demo showcases a pushdown analysis architecture that distributes complex query execution—including those used in scientific visualization—across

An Equal Opportunity Employer / Managed by Triad National Security, LLC, for the U.S. Department of Energy's NNSA
LA-UR-25-25395

application and storage layers. Client applications issue SQL queries to distributed engines such as Presto, Apache Spark, and Apache DataFusion, which use custom plugins to decompose and translate the required work into query plans represented in Substrait for offloading to an underlying pNFS system. pNFS metadata servers provide file layout information to route queries appropriately, while custom Apache Arrow Flight gRPC servers—dynamically launched as user processes on pNFS data servers—execute offloaded queries as close to the data as possible. These servers leverage DuckDB to parse Substrait plans, process data stored in popular formats such as Parquet, and return results as Apache Arrow data streams.

A detailed breakdown of this pushdown process is shown on Page 3. In Steps 5 and 6, we leverage a key capability of Hammerspace's pNFS metadata server: the ability to resolve file names to the specific data servers where the files reside. For simplicity, we currently assume that each file is stored entirely on a single pNFS data server, with no striping. In Step 7, a FUSE mount acts as a secure mechanism for users to launch gRPC servers on pNFS data nodes to handle offloaded query processing. This mount does not store any actual data itself.

In Steps 13 through 16, the pNFS data server plays a dual role—both as a pNFS client and as a data server. As a client, it communicates with the metadata server to open the file and recognizes that the data resides locally. The pNFS read operation is then transparently converted into a local read, enabled by a recent Linux kernel patch from Hammerspace, allowing the read operation to run efficiently.

Results and Future Directions

By offloading queries closer to the data, this pushdown architecture enables selective retrieval of only the information that matters. This not only reduces data movement and lightens the load on downstream systems, but also empowers large-scale analysis from modest platforms—even a scientist's laptop—while speeding up discovery.

To show how this capability impacts real-world sciences, this demo analyzes an asteroid's descent into Earth ocean water, using contouring—a common filtering technique akin to elevation lines on a map—to trace how the asteroid and water deform upon impact. Contours help isolate meaningful regions, like surfaces and boundaries. In this visualization, red represents the asteroid, blue depicts the ocean, and yellow indicates temperature. A screenshot is shown on page 4.

Traditional visualization pipelines require loading the entire dataset from storage before applying filters such as contouring. By enabling storage to actively participate in data retrieval and filtering, we observe up to a 99% reduction in data movement and client memory usage, as storage preselects and transfers only the relevant data. While the "X-fold speedup in analysis time" is less apparent in this demo—since data movement isn't the primary bottleneck at this scale—we expect the performance benefits to be substantial in production environments.

Looking ahead, we have several areas of future work in mind. First, pNFS currently lacks efficient support for client-side erasure coding and concurrent writes from multiple processes to a single file—both critical features for broader adoption in production environments. Second, while open-source analytics tools like Presto and DuckDB offer powerful capabilities, they make it difficult to offload complex filters commonly used in scientific visualization pipelines to storage. To address this, we plan to implement these filters as custom SQL functions that can be invoked by analysis programs, offloaded by distributed SQL engines like Presto, and executed by embedded engines such as DuckDB. LANL, Hammerspace, and SK hynix are actively collaborating to make these visions a reality. Stay tuned!

This demo is supported by Supermicro, Hammerspace, and SK hynix.

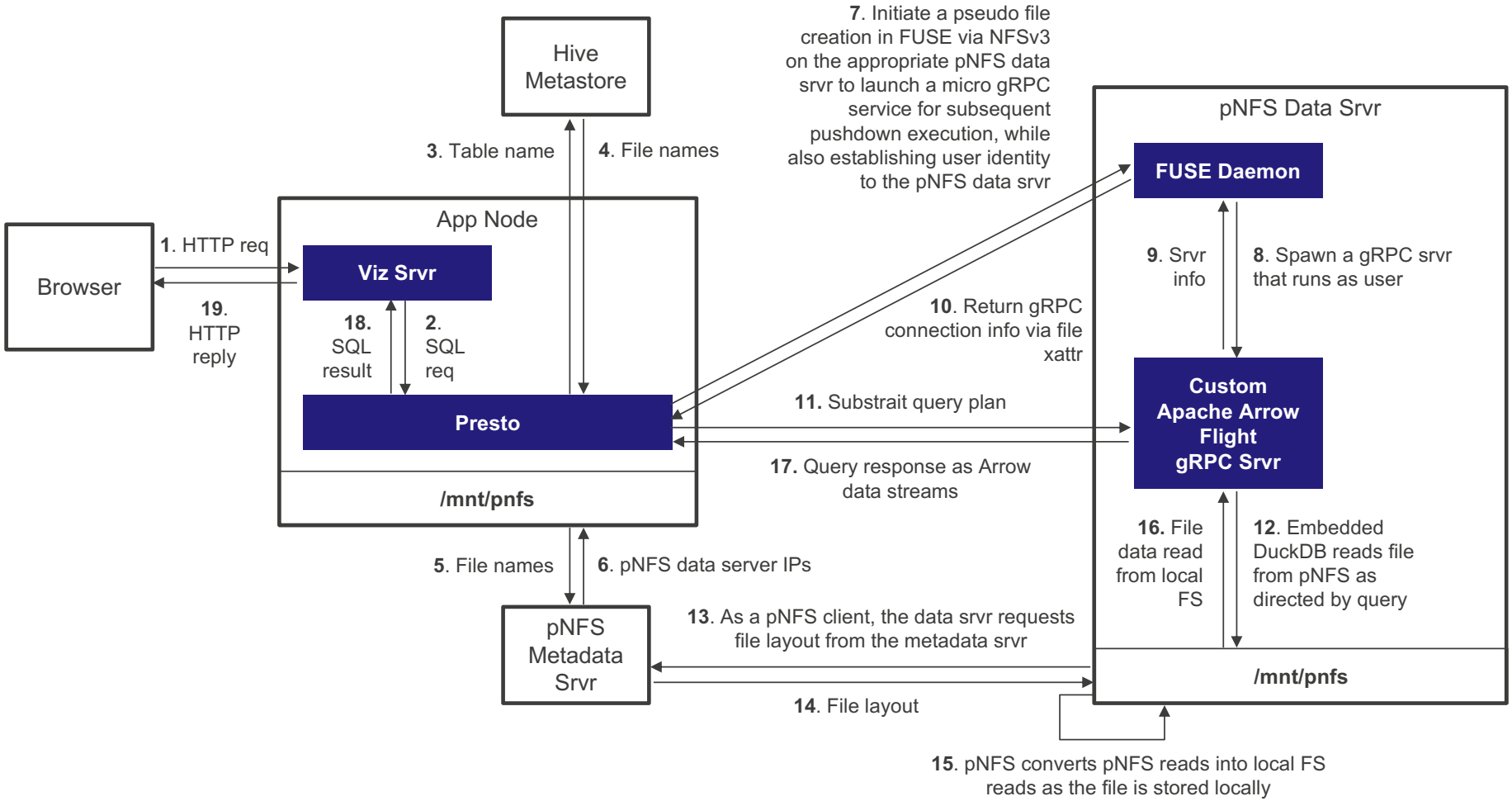

Figure 1. This demo showcases an open pushdown architecture built on standard pNFS protocols and tools from the big data and analytics ecosystem including Persto, Substrait, Apache Arrow, and DuckDB. SQL queries are decomposed at Presto into Substrait plans and offloaded to pNFS data servers, where dynamically launched Apache Arrow Flight gRPC servers execute them close to the data using DuckDB. A key feature of Hammerspace's pNFS metadata server enables file-to-server resolution, while the FUSE mount on the pNFS data server allows users to securely launch gRPC servers. In later steps, the pNFS data server acts as both client and server, converting pNFS reads into efficient local reads via a recent Hammerspace kernel patch. This architecture enables selective, near-data query execution—reducing data movement by up to 99% and supporting large-scale analysis even from modest platforms.

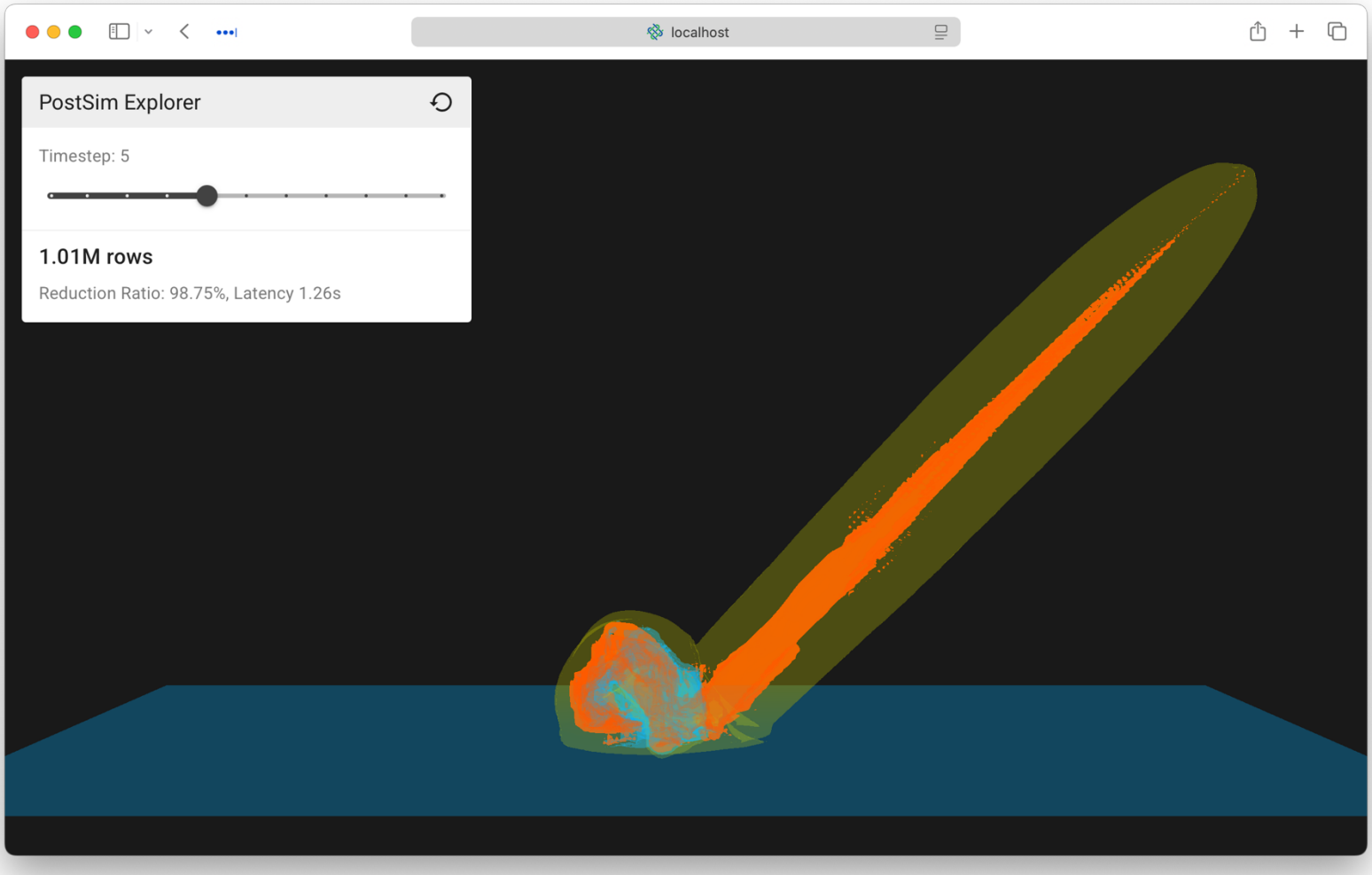

Figure 2. Asteroid impact visualization using contouring to trace how the asteroid and ocean water deform upon impact. In this visualization, red represents the asteroid, blue depicts the ocean, and yellow indicates temperature. Offloading filtering to storage reduces data movement and client memory usage by up to 99% by transferring only relevant data.