

The Present and the Future for Usable In Situ Analysis in HPC: Solving the Problems with Active Storage

Niclas Schroeter

Deutsches Klimarechenzentrum (DKRZ) / Otto-von-Guericke Universität
Magdeburg (OVGU)

September 2026



Post-Processing in Scientific Workflows

Current State of the Art

- 1 Run simulation on compute nodes, with output at predefined time steps
- 2 Read the data and do post-processing tasks (statistics, visualization etc) on compute or post-processing nodes

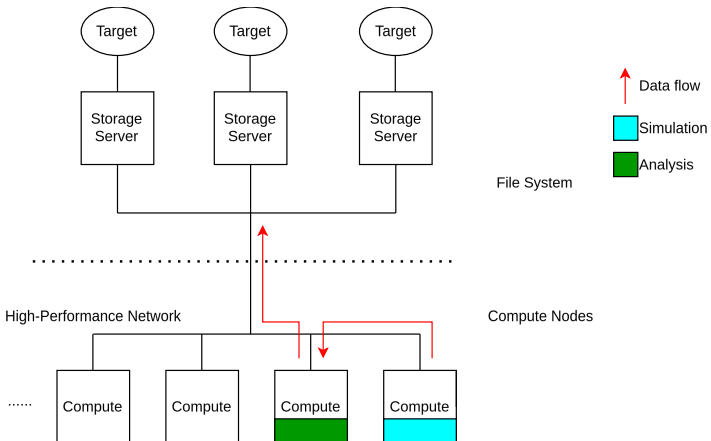
Problems with this approach

- Stresses network, file system; not energy-efficient
- "Time to insight" is longer due to sequential nature of the steps
- Problems at Exascale: amount of storage to store data is too large, storage systems improved at much slower rate

Potential Solution: In Situ Analysis

- Combine simulation and analysis, run them simultaneously
- Either on simulation nodes or on different compute nodes
- Store only the analysis results

Architecture Example: In Situ Workflow



Problems with Current In Situ Solutions

- Often requires invasive changes to the code
- Large barrier to entry
 - ⇒ Tighter integration with I/O libraries is promising
- Mostly targeted at the largest simulations (Exascale), leaving the majority of users behind
- Targeted hardware (computes nodes of the HPC system) already heavily contested

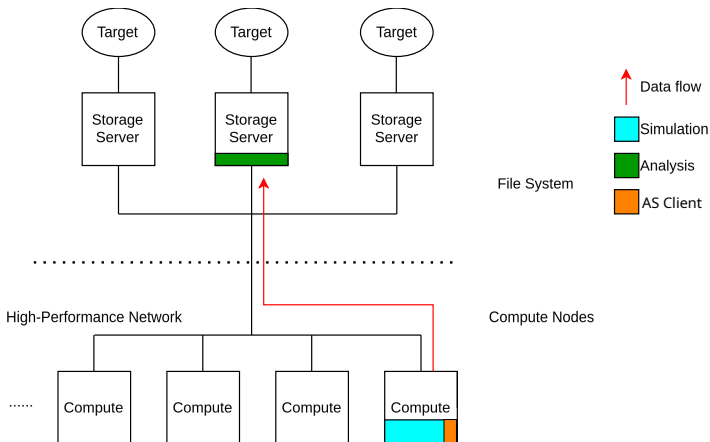
Active Storage

- Extend file system to allow user-driven calculations on FS hardware
- Explored ~ 20 years ago, did not go beyond research prototypes
- Basic idea becoming more relevant again: "Bring computation to the data, not the other way around"
- Modern FS hardware is underutilized, esp. the CPUs
- Care: FS performance must not degrade!

Active Storage

- Client/Server architecture based on RPCs, FS agnostic for now
- Client is integrated into HDF5 plugin
⇒ Intercept write/flush calls to communicate with server
- Predefined data analysis operations instead of user-defined
- Minimal assumptions regarding input data (i.e., time is first dimension)

Architecture Example: Active Storage



Data Flow

Mode 1: Direct Processing

- Client sends raw data to server
- Server only writes analysis results
- Less flexible, but stresses the FS less

Mode 2: Files as Staging Area

- Client writes data to file, sends file name to server
- Server reads data from file
- Increased flexibility at the cost of FS stress
- Could delete staging files afterwards

Initial Evaluation

- Tested with dummy application: write 20 GB over 128 time steps, post-processing is mean calculation per time step
- Two test setups:
 - 1 Local workstation with SSD
 - 2 Single node of Levante supercomputer with Lustre and HDDs
- Setups more akin to traditional *in situ* workflows, not making use of Active Storage

	Local	Levante
Traditional	11s	33s
<i>In situ</i>	10s	23s

Future Work

- Test cluster: One storage server, 8 client nodes
- Determine baseline FS/hardware performance
- Then introduce Active Storage:
 - Performance?
 - Task scheduling?
 - Limitations?

Summary

- Current data analysis workflows will not scale
- Combine in-situ data analysis with Active Storage
- Ensure transparency and usability
- Provided prototype shows promising results
- Could bridge gap between *in situ* and HPC communities
- Potential to improve hardware utilization of FS hardware