Rich Operators for Advanced In Situ Processing

Hua Xu Anthony Kougkas Xian-He Sun
hxu40@hawk.iit.edu,akougkas,sun@iit.edu

Gnosis Research Center @ ILLINOIS TECH

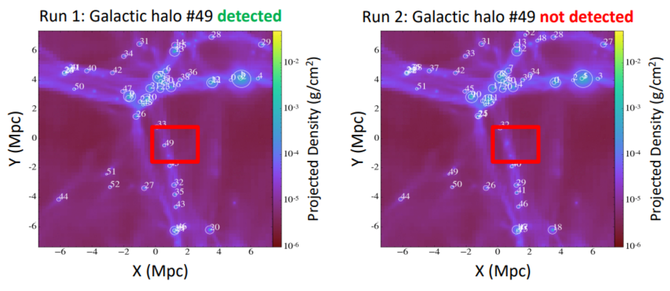
StoreHub

Introduction

- The rapid growth of scientific data has made reading and writing to disk increasingly **expensive**, creating a pressing need for **in-situ processing** to reduce I/O costs.
- While in-situ methods are often used for basic data reduction, their broader potential for integrating complex, traditionally offline operations into the **I/O pipeline** remains underexplored.
- This work reimagines the I/O path as a platform for in-situ processing by enabling **flexible, pluggable operators** capable of performing arbitrary computational tasks, including **derived quantity generation**.

Study case: reproducibility

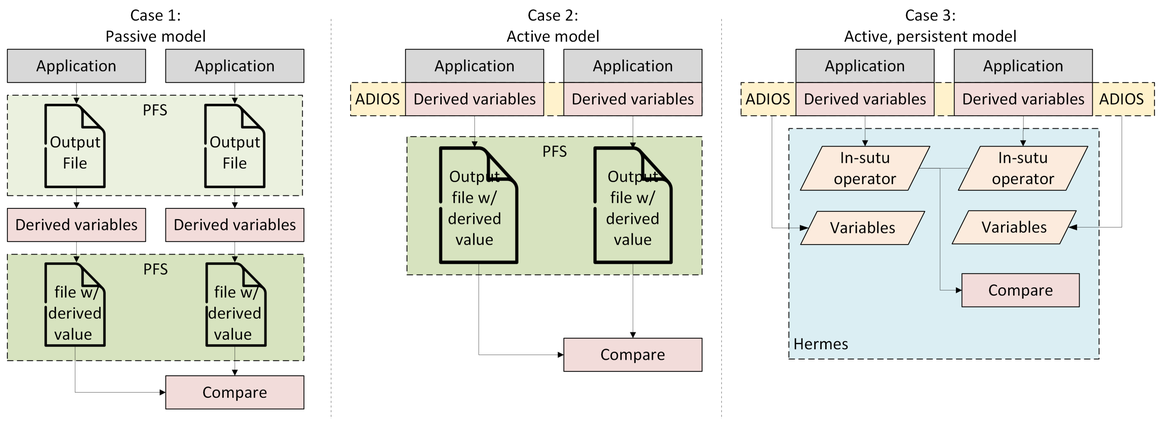
- General simulations can encounter errors stemming from **concurrency** issues, **numerical inaccuracies**, and **hardware-related factors**.
- For example, Stodden, et. al. [1] reported **discrepancies** between two runs of a galactic simulation.
- We implemented a hash operator within ADIOS that can wrap around any existing ADIOS **variable**, **calculate its hash**, and terminate early if it detects **significant** variability compared to a previous run's **hash**.



Design

Our approach addresses the following key areas:

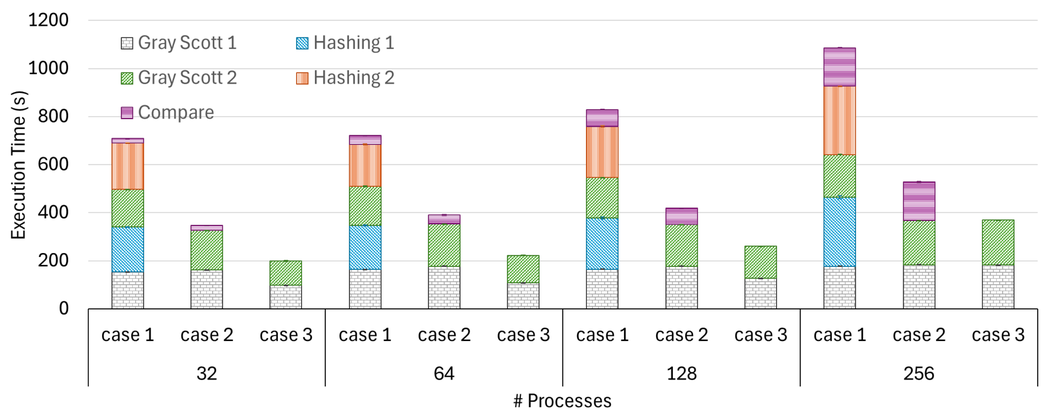
1. The initial handling of simulation data and the activation of the **I/O layer** to define and compute derived quantities.
2. A **user-defined operator** that allows portable, one-line **integration** of custom operators into the storage layer, enabling in-situ computation within the application.
3. **Hierarchical storage management** strategies enabling optimized storage and retrieval of derived data.
4. The **runtime control mechanisms** that orchestrate online hash code comparison for data validation.



Experimental Setups for Three Cases

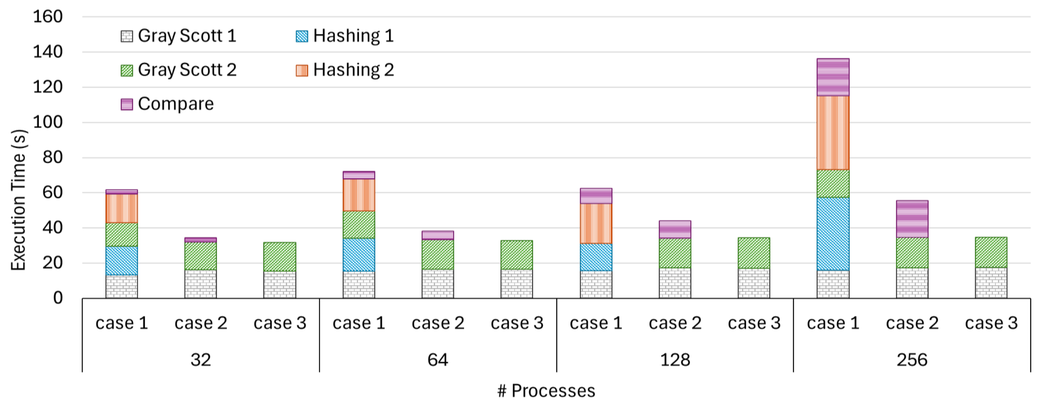
Evaluation

- We evaluate the system using the Gray-Scott application, performing **weak scaling** tests with 32 to 256 processes on three different platforms.



(a) Ares from IIT

(b) Doom from Sandia



(c) Frontier from Oak Ridge

Execution time on three platforms for weak scaling from 32 to 256 processes in Gray-Scott

Observation:

- **Case 1** and **Case 2** suffer from **limited scalability** due to excessive I/O movement and hashing overhead.
- **Case 3** proves **most efficient** at scale by handling data and comparisons asynchronously.
- In-situ operators in **I/O layer** demonstrate strong **portability** across three distinct platforms.

Acknowledge

This material is based upon work supported in part by the National Science Foundation (NSF), under Grant CCRI-CISE 2346504 and U.S. Department of Energy (DOE), under DE-SC0023263.

References

[1] Victoria Stodden and Matthew S. Krafczyk.
Assessing reproducibility: An astrophysical example of computational uncertainty in the hpc context.
In *ResCuE-HPC'18: The 1st Workshop on Reproducible, Customizable and Portable Workflows for HPC at SC'18*, 2018.

Code available on GitHub: github.com/grc-iit/coeus-adapter