# Towards A Practical Provenance Framework for Scientific Data on HPC Systems

Runzhou Han, Suren Byna* and Mai Zheng

*Dept of Electrical and Computer Engineering, Iowa State University*

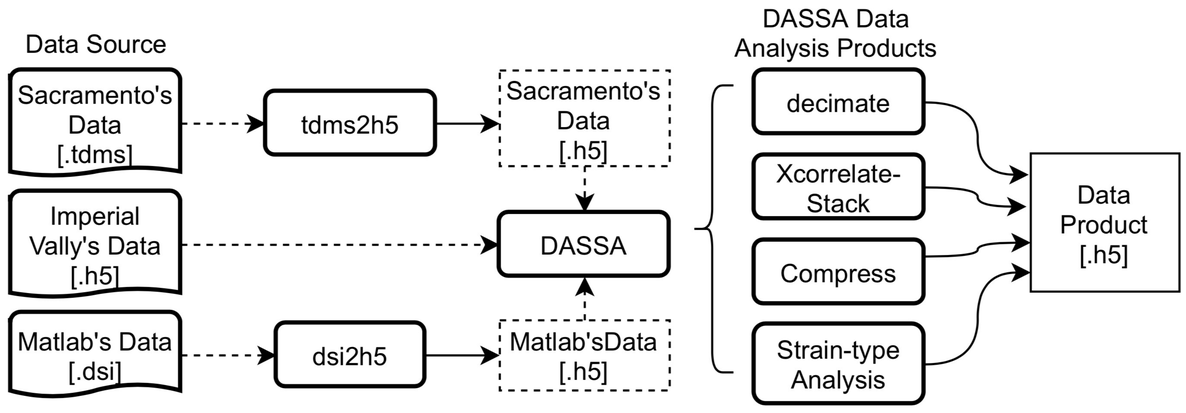
** Lawrence Berkeley National Labs*

FAST'22
20th USENIX Conference on File and Storage Technologies
FEBRUARY 22–24, 2022
SANTA CLARA, CA, USA
www.usenix.org/fast22

## Motivation

- A need for data provenance in HPC scientific workflows
  - Workflow & scientific data reusability
  - Scientific data lineage tracking
  - Workflow optimization
- Challenges
  - Little **quantitative** understanding or specification of scientists' diverse needs
  - The variety of the workflow characteristics (e.g., different I/O interfaces and file formats)

***Example: DASSA, an acoustic sensing workflow [1].*** *DASSA takes input dataset in multiple different formats (i.e., '.tdms', '.h5' and '.dsi') and uniformly converts them to intermediate '.h5' files in its first step. It then processes the intermediate data with four different data analysis products and gives the final data product. In DASSA workflow, the scientists want to know, given a final data product, what's the original input data (backward lineage)*




- A gap between the HPC data provenance and existing solutions
  - Vague provenance model
  - Incompatible provenance software on HPC environment
  - Limited transparency

| Software | Type | Base Model | Environment | Language | Transparent? |
|---|---|---|---|---|---|
| **Karma** | Middleware | OPM | Cloud | Java | Yes |
| **Komadu** | Provenance system | W3C PROV-DM | Cloud | Java | No |
| **Tanerna** | Workflow system | W3C PROV-DM | Desktop | Java | Hybrid |
| **ProvLake** | Provenance system | W3C PROV-DM | Cloud | Python | No |

**Table 1: Existing Provenance Software for Scientific Workflows**

## Methodology

- Case studies on realistic workflows and the end users' provenance needs
- A provenance model for HPC data provenance
- A provenance framework
  - Provenance Tracking for capturing diverse I/O operations
  - Provenance Storage for persisting the captured provenance information as standard RDF triples [2]
  - User Engine for querying and visualizing provenance

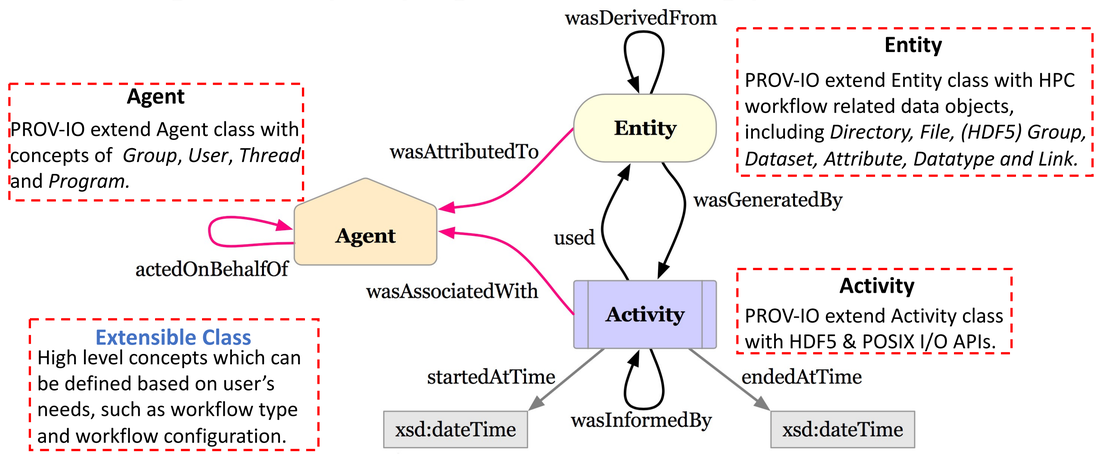



**Figure 1: PROV-IO Model.** An extension of W3C PROV-DM

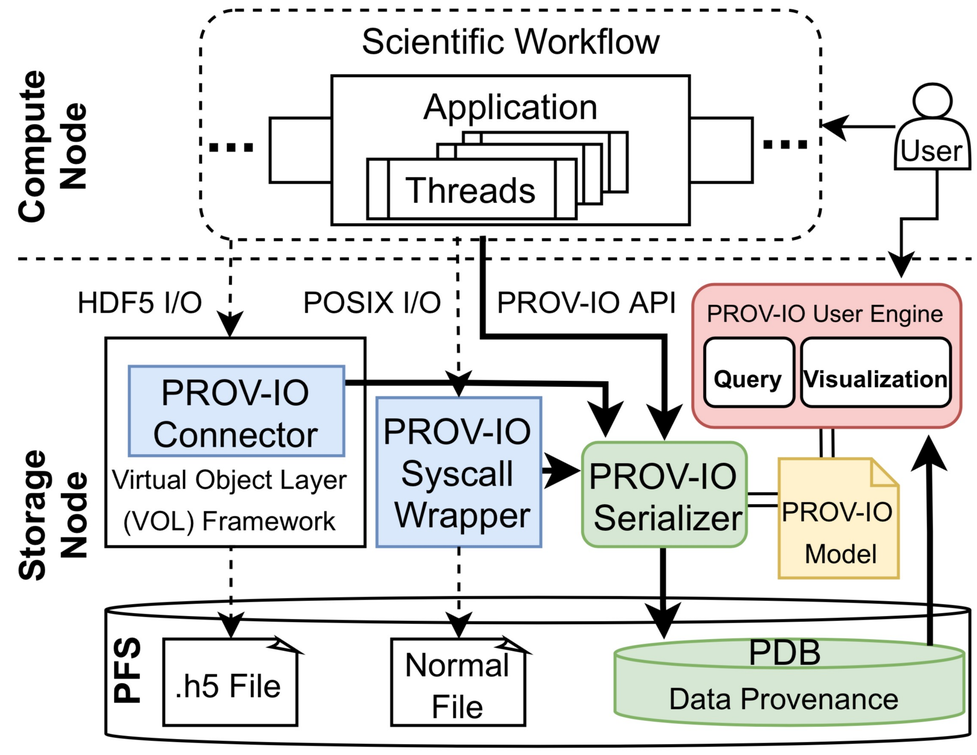



**Figure 2: PROV-IO framework. T**he scientific workflow's I/O operations (e.g. , HDF5 I/O and POSIX I/O) from the compute node are redirected to PROV-IO framework on the storage node. PROV-IO lets I/O calls pass through and only captures provenance information. It then serializes provenance to the provenance database (PDB) which can be queried and visualized

## Preliminary Results

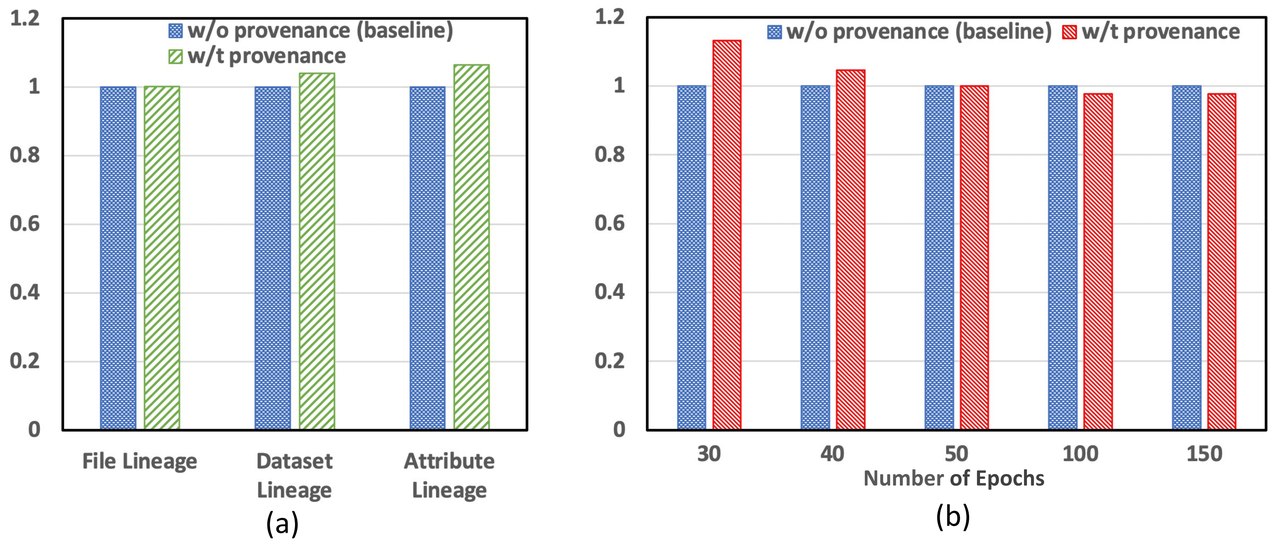



**Figure 3: Performance of Provenance Tracking.** (a) Tracking performance of DASSA. (b) Tracking performance of Top Reco. In the two figures, tracking performance is measured with normalized workflow completion time

- A maximum provenance tracking overhead of 5.2% In DASSA workflow, with the most fine-grained provenance level enabled (HDF5 Attribute Lineage)
- A maximum provenance tracking overhead of 13.2% in Top Reco workflow, with the training epoch of 30. The tracking overhead becomes negligible when applying more training epochs

## Future Work

- Fully implement PROV-IO Syscall Wrapper
- Improve PDB & User engine
  - Optimization existing database solutions and PROV-IO user APIs

## Acknowledgements

- This work was supported in part by NSF under grant CCF-1910747. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of NSF

## References

[1] Bin Dong, Verónica R. Tribaldos, Xin Xing, Suren Byna, Jonathan Ajo-Franklin, Kesheng Wu. DASSA: Parallel DAS Data Storage and Analysis for Subsurface Event Detection. In Proceedings of IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2020.

[2] Resource description framework. https://www.w3.org/RDF/