Towards A Practical Provenance Framework for Scientific Data on HPC Systems

Runzhou Han
Iowa State University

Suren Byna
Lawrence Berkeley National Laboratory

Mai Zheng
Iowa State University

1 Motivation

Data-driven scientific discovery has been well acknowledged as a new fourth paradigm of scientific innovation [17]. The shift toward the data-driven paradigm imposes new challenges in data reproducibility, explainability, trustworthiness, etc [7], all of which demand innovative solutions for modeling and capturing provenance, i.e., the lineage of data life cycle.

While the importance of provenance has been well recognized across communities (e.g., databases [1] [10] [12] [6] [19], operating systems [9, 8], eScience [5] [13] [2] [16] [15]), we find that there are three major gaps between the available solutions and the real needs of domain scientists.

First, in terms of provenance modeling, we find that existing standards (e.g., W3C PROV [18]) are too vague to describe the specific needs of scientific data and scientists precisely. Scientists often seek a variety of information from scientific workflows on high performance computing (HPC) systems, including the origins of data products, the configurations used for deriving results, the usage patterns of datasets, and so on, which cannot be described effectively using any existing provenance models. Such ambiguity fundamentally limits the usability of existing provenance products for scientific data.

Second, in terms of provenance systems, existing products are largely incompatible with the scientific workflows or HPC systems. For example, it is common for HPC workflows to use HDF5 library [4] or a combination of HDF5 and POSIX system calls [11] to perform I/O operations and access data in different granularity (e.g., file, dataset, attribute). However, to the best of our knowledge, none of existing products can handle HDF5-related I/O operations effectively or efficiently. Moreover, existing systems lack the flexibility to handle high-level semantic information that is much needed by domain scientist (e.g., high-level metrics of a workflow).

Last but not the least, existing solutions often require manual efforts to identify relevant source code locations for instrumentation [14, 15], which is inconvenient for domain scientists and impractical for large-scale scientific workflows.

2 Our Approach

To address the challenge, we first analyze three representative scientific workflows in collaboration with domain scientists. In doing so, we identify the unique characteristics of the workflows (e.g., I/O interfaces, data formats) as well as the specific needs for provenance (e.g., lineage at file, dataset, or attribute granularity).

Based on the first-hand investigation, we propose a practical provenance framework for scientific data on HPC systems. Our key observation is that I/O operations are critically important in affecting the state of data that form the lineage. So we derive an I/O-centric provenance model, which enriches the W3C PROV standard [18] with a variety of concrete sub-classes to describe both the data and the associated I/O operations and execution environments precisely with extensibility.

Moreover, based on the unique provenance model, we are building a practical prototype which includes three major components: (1) Provenance Tracking for capturing diverse I/O operations; (2) Provenance Storage for persisting the captured provenance information as standard RDF triples [3] ; (3) User Engine for querying and visualizing provenance. The prototype can provide end-to-end provenance support with little manual effort, which is essential for addressing the provenance problem in complex HPC environments.

Our preliminary evaluation on a state-of-the-art HPC system with realistic scientific workflows show that the provenance framework incurs reasonable tracking and storage overhead for the use cases evaluated. More importantly, through the query and visualization support, the prototype can address the data provenance needs of the end users effectively.

References

[1] Buneman, P., Khanna, S., and Tan, W. C. Why and where: A characterization of data provenance. In *Proceedings of the 8th International Conference on Database Theory* (2001).

[2] Davidson, S. B., and Freire, J. Provenance and scientific workflows: Challenges and opportunities. In *ACM SIGMOD* (2008).

[3] Framework, R. D. https://www.w3.org/rdf/.

[4] HDF5. https://github.com/hpc-io/h5bench.

[5] Jandre, E., Diirr, B., and Braganholo, V. Provenance in collaborative in silico scientific research: A survey. *SIGMOD Rec.* (2020).

[6] Miao, Z., Zeng, Q., Glavic, B., and Roy, S. Going beyond provenance: Explaining query answers with pattern-based counterbalances. In *Proceedings of the 2019 International Conference on Management of Data* (2019).

[7] Milton, M. J., and Possolo, A. Trustworthy data underpin reproducible research. *Nature Physics* (2020).

[8] Muniswamy-Reddy, K.-K., Braun, U., Holland, D. A., Macko, P., Maclean, D., Margo, D., Seltzer, M., and Smogor, R. Layering in provenance systems. In *USENIX ATC* (2009).

[9] Muniswamy-Reddy, K.-K., Holland, D. A., Braun, U., and Seltzer, M. Provenance-aware storage systems. In *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference* (2006).

[10] Niu, X., Glavic, B., Gawlick, D., Liu, Z. H., Krishnaswamy, V., and Radhakrishnan, V. Interoperability for provenance-aware databases using PROV and JSON. In *7th USENIX Workshop on the Theory and Practice of Provenance (TaPP 15)* (2015).

[11] POSIX. https://en.wikipedia.org/wiki/posix.

[12] Senellart, P. Provenance and probabilities in relational databases. *SIGMOD Rec.* (2018).

[13] Simmhan, Y. L., Plale, B., and Gannon, D. A survey of data provenance in e-science. *SIGMOD Rec.* (2005).

[14] Simmhan, Y. L., Plale, B., and Gannon, D. A framework for collecting provenance in data-centric scientific workflows. In *2006 IEEE International Conference on Web Services (ICWS'06)* (2006).

[15] Souza, R., Azevedo, L., Thiago, R., Soares, E., Nery, M., Netto, M. A. S., Vital, E., Cerqueira, R., Valduriez, P., and Mattoso, M. Efficient runtime capture of multiworkflow data using provenance. In *e-Science* (2019).

[16] Suriarachchi, I., Withana, S., and Plale, B. Big provenance stream processing for data intensive computations. In *e-Science* (2018).

[17] Tolle, K., Tansley, S., and Hey, T. The fourth paradigm: Data-intensive scientific discovery [point of view]. *Proceedings of the IEEE* (2011).

[18] W3C, T. P. D. M. https://www.w3.org/tr/prov-overview/.

[19] Wang, Z., Xu, Y., Suo, B., Wang, Z., Xu, J., Chen, Q., and Li, Z. A provenance storage method based on parallel database. In *2015 2nd International Conference on Information Science and Control Engineering* (2015).

Towards A Practical Provenance Framework for Scientific Data on HPC Systems

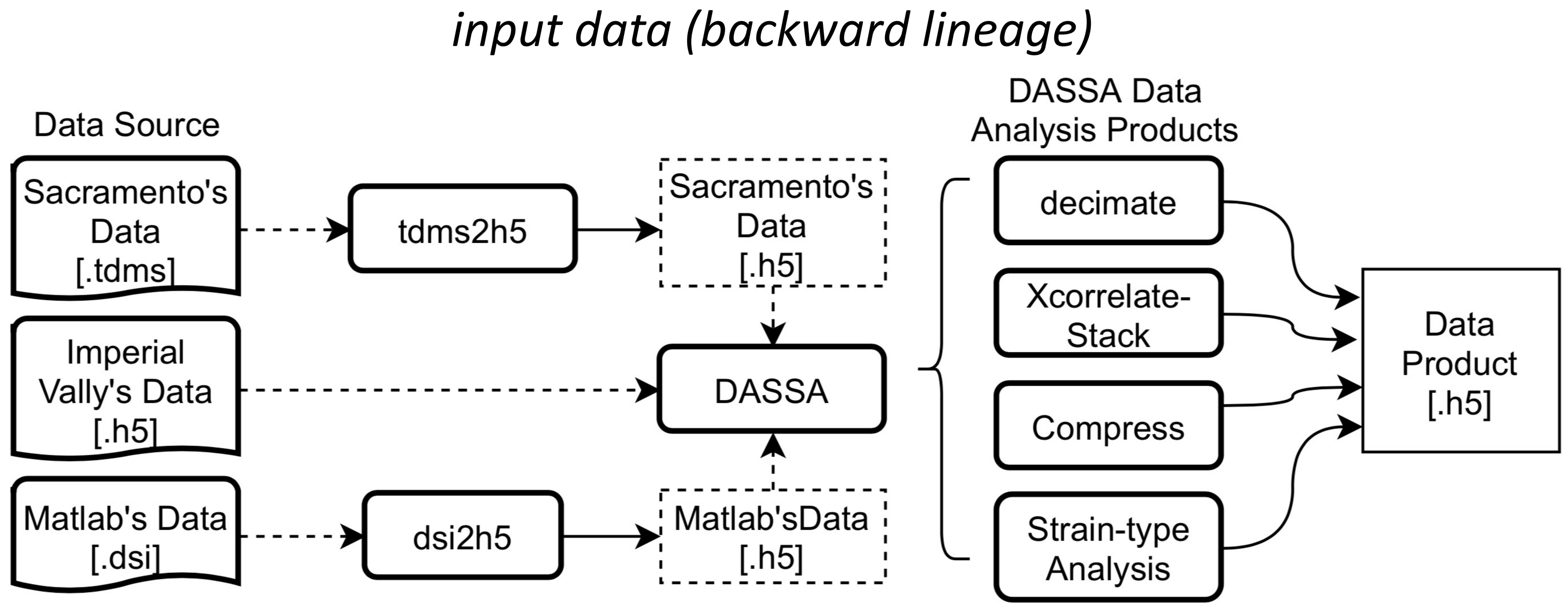
Runzhou Han, Suren Byna* and Mai Zheng
Dept of Electrical and Computer Engineering, Iowa State University
** Lawrence Berkeley National Labs*

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

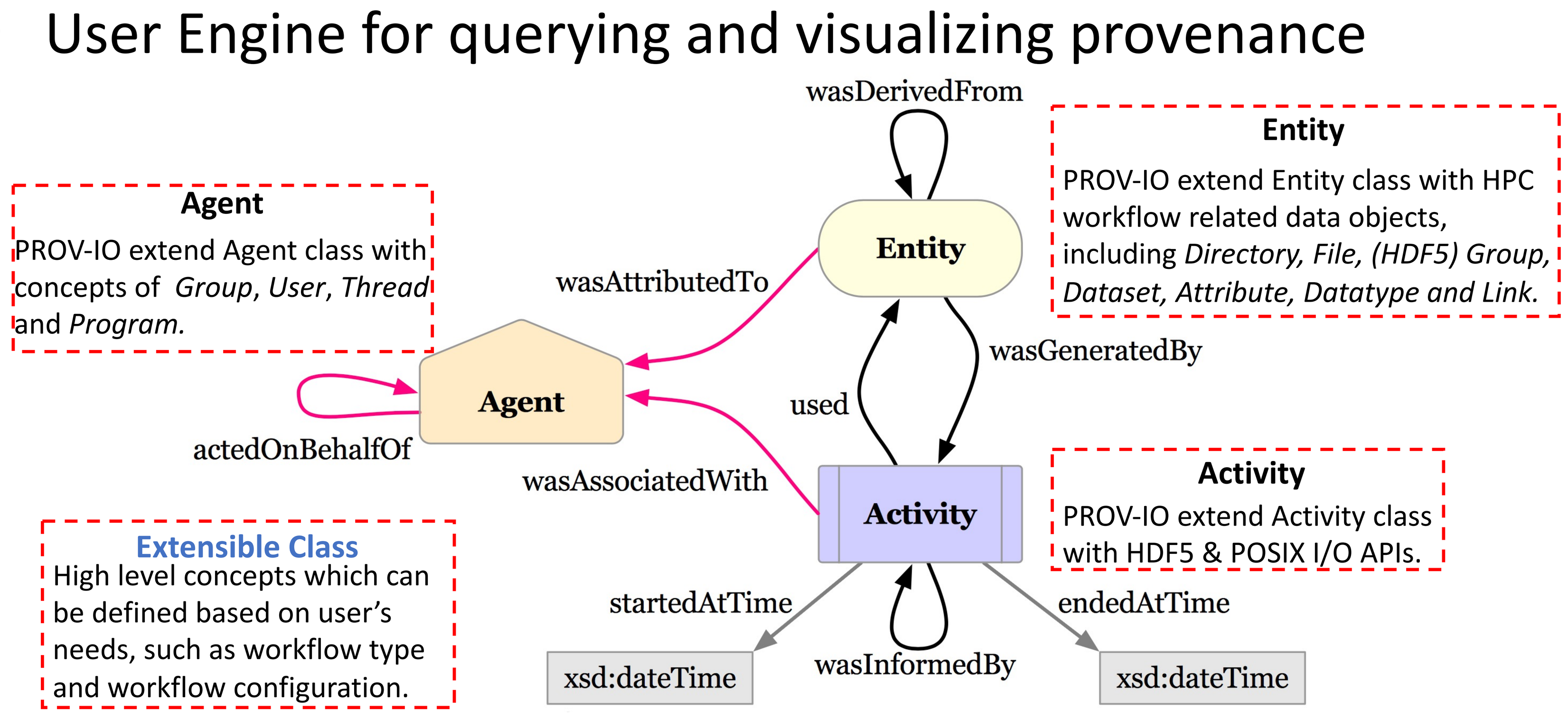


Figure 1: PROV-IO Model. An extension of W3C PROV-DM

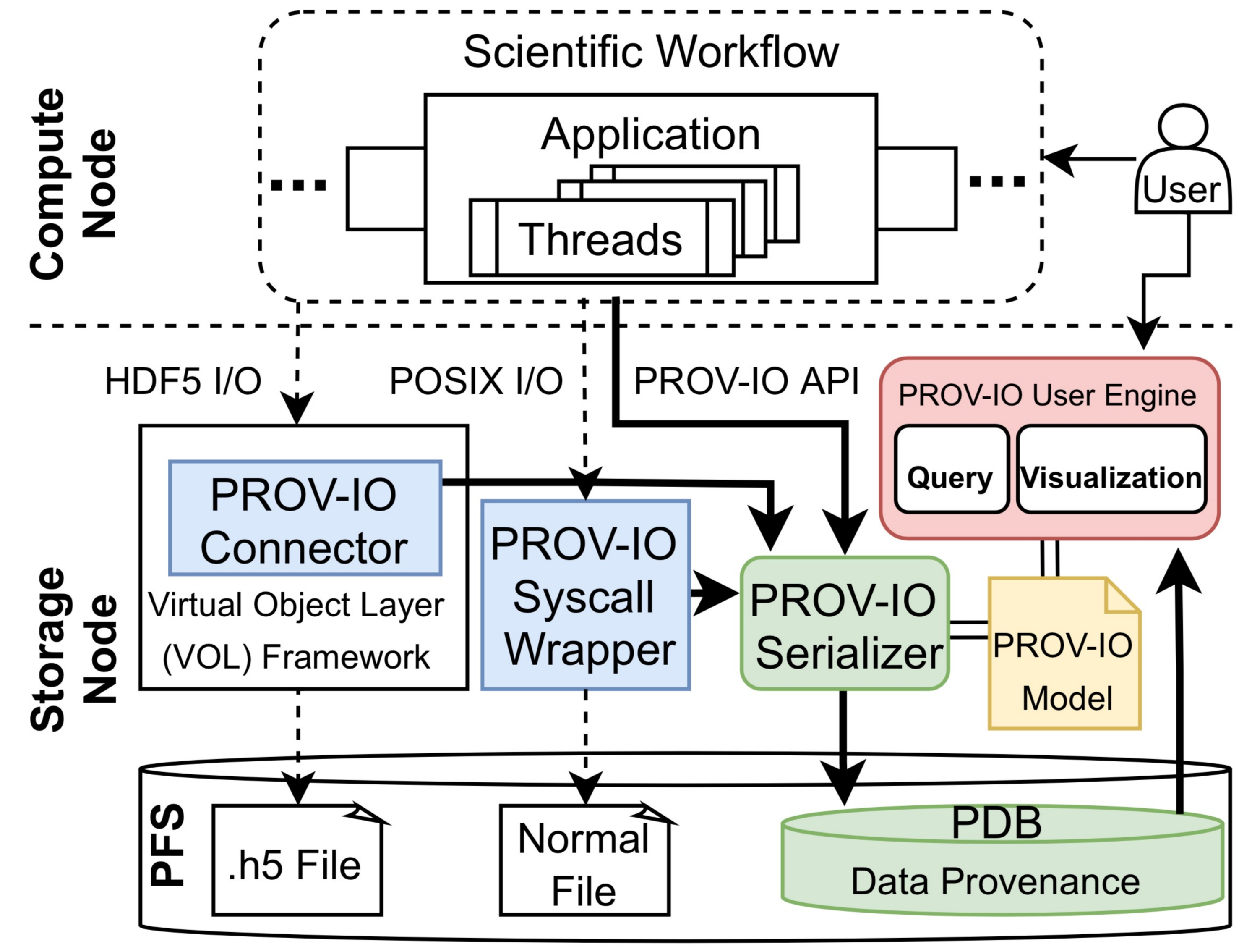


Figure 2: PROV-IO framework. The scientific workflow's I/O operations (e.g. , HDF5 I/O and POSIX I/O) from the compute node are redirected to PROV-IO framework on the storage node. PROV-IO lets I/O calls pass through and only captures provenance information. It then serializes provenance to the provenance database (PDB) which can be queried and visualized

Preliminary Results

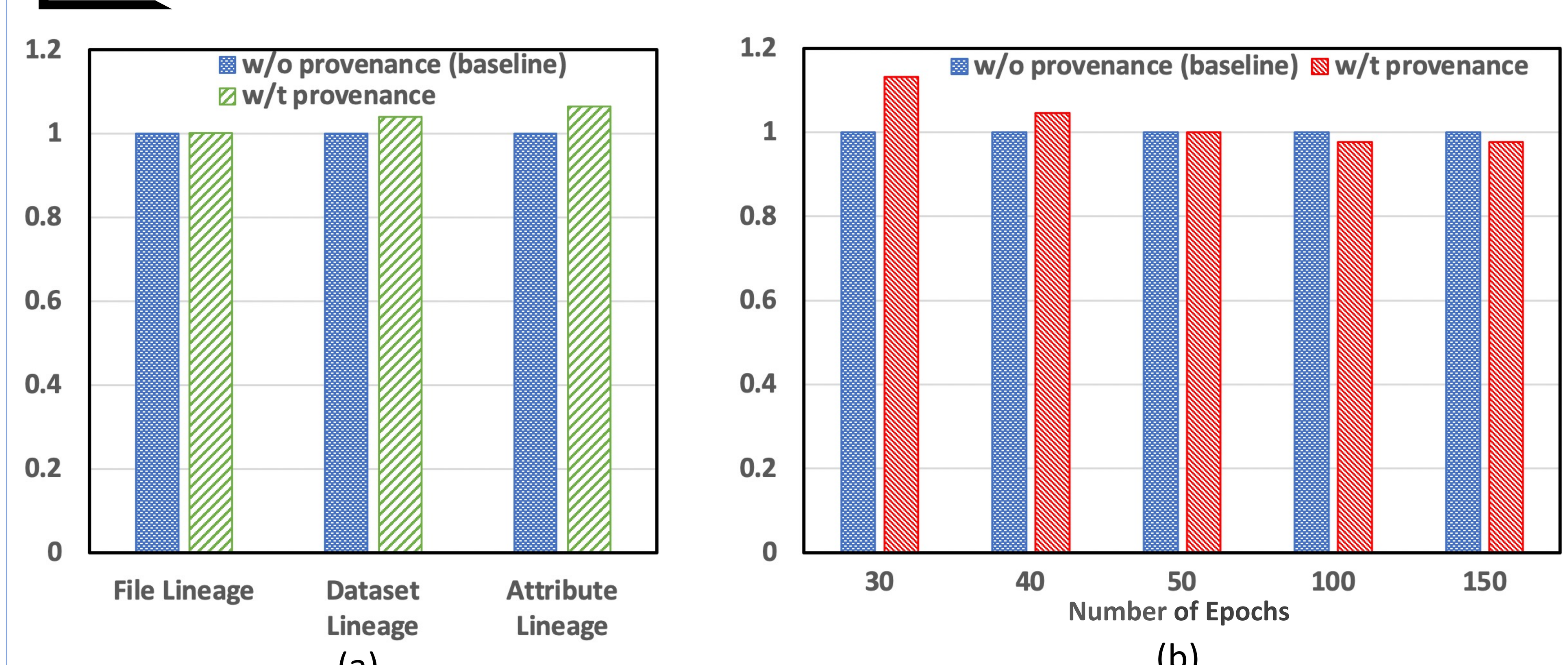


Figure 3: Performance of Provenance Tracking. (a) Tracking performance of DASSA. (b) Tracking performance of Top Reco. In the two figures, tracking performance is measured with normalized workflow completion time

- 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/