Braid-DB: Toward AI-driven Science with Machine Learning Provenance

Justin M. Wozniak, Zhengchun Liu, Rafael Vescovi, Ryan Chard, Bogdan Nicolae, and Ian Foster

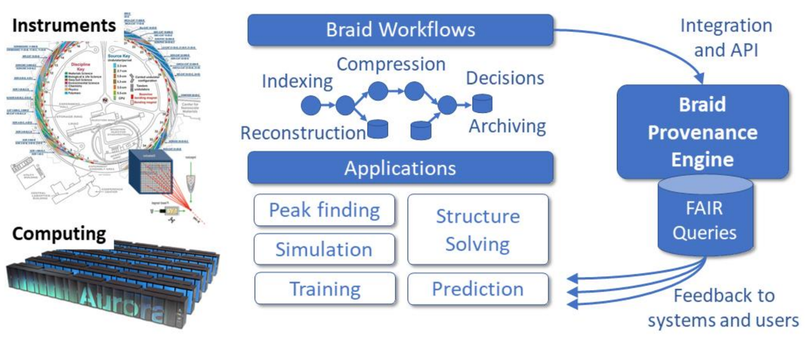
Argonne National Laboratory

https://github.com/ANL-Braid/DB

Overview: Provenance for ML Science

- Next-generation scientific instruments will collect data at unprecedented rates: multiple GB/s and exceeding TB/day.
- Such runs will benefit from automation and steering via machine learning methods, but these approaches require new data management and policy techniques to support the distinct I/O patterns of machine learning and the dynamic and more generalized dataflow pattern inherent in automation.
- We present here the Braid Provenance Engine (Braid-DB), a system that embraces and is designed to support AI-for-science automation in how and when to analyze and retain data, and when to alter experimental configurations.
- Automating such workflows will need provenance recording that is augmented with richer information about model training inputs, including real-world experiments and observations, simulations, and the structures of other learning and analysis activities.
- We must stretch the notion of a provenance database to capture datasets produced by mixing experimental data with ML models.

Braid: Streaming data for experiments



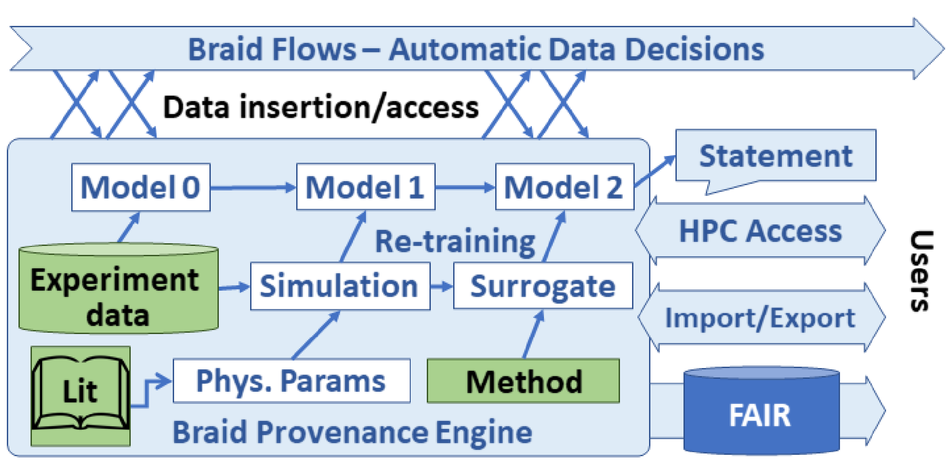
- The over-arching Braid project funnels experimental data streams into Braid Workflows controlled by Globus Flows and funcX invocations
- ML may be applied to control the experiment, simulation, and analysis at multiple points in the workflow
- Braid-DB captures the decisions made and relevant metadata for validation and/or fault diagnosis

Applications

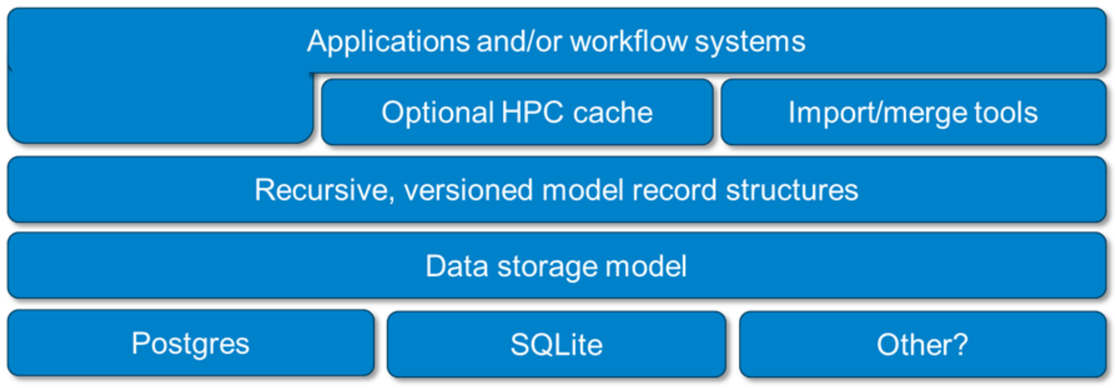
Provenance flow capture for training DNNs in x-ray science:
Data collected in early stages of an experiment, data from past similar experiments, and/or data simulated for upcoming experiments are used to train a deep neural network (DNN) model that, in effect, learns specific characteristics of those data; this model is then used to process subsequent data more efficiently than would general-purpose models that lack knowledge of the specific dataset or data class (Liu 2021). In many cases, the DNN needs to be updated (retrained and fine-tuned) frequently to keep up with changes in experiment setup and sample conditions.

Serial synchrotron x-ray crystallography: Argonne's Structural Biology Center has developed a Braid-compatible pipeline to process raw data, catalog and report interim results, and attempt to refine and solve protein structures (Wilamowskie 2021). This process captures sample information (including protein, preparation technique, exposure, and temperature) and feeds it into the analysis and publication pipeline. It will allow the experiment control algorithms to decide what are the next steps to complete the acquisition.

Braid-DB Architecture



- Braid workflows and policies make automated decisions in support of experimental science
- Braid-DB tracks those decisions in terms of data dependency relationships among static data, experimental data, data derived from simulation or analysis, and ML model-produced data
- The system produces *statements* that are supported by records in the database
- Braid-DB extends traditional provenance capabilities by:
  - Allowing ML models to influence other models
  - Allowing ML models to be updated over time



- The current system is implemented as a Python API and object model wrapped around a SQLite database
- The system is also accessible via a funcX functional API
- An MPI-enabled API will also be developed for use on HPC

Abstractions Stored in Database

BraidRecord: A super-class for Braid-DB provenance records. Each such entity has a unique ID, a Swift PIPS script: Parallel power grid analysis
(possibly not unique) string name, a list of dependencies, and a dictionary of user-specified, string-keyed metadata tags,
BraidFact: A simpler object consisting of static data: for example, pre-existing trusted data or software. BraidFacts may have a provenance outside the system.
BraidData: The Braid-DB representation of traditional provenance-tracked data, with traditional conceptions of its derivation history from other BraidData and/or BraidFacts. A Braid-DB containing only BraidData and BraidFacts would be functionally indistinguishable from a traditional provenance database.
BraidModel: An ML model tracked by Braid-DB. A BraidModel has the additional capability `update()` , which represents model exposure to other BraidRecords, possibly including other models. This includes the possibility of dependency cycles that capture complex interactions among models and data as experiment workflows progress.

Liu 2021: Z. Liu et al. Bridge data center AI systems with edge computing for actionable information retrieval. arXiv preprint arXiv:2105.13967 (2021)
Wilamowski 2021: Wilamowski et al. 2'-o methylation of RNA cap in SARS-CoV-2 captured by serialcrystallography. PNAS 118(21) (2021).

Acknowledgments: This work was supported by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under contract number DE-AC02-06CH11357.