Revealing the Detailed History of Script Outputs with Hybrid Provenance Queries

Yang Cao¹, Duc Vu², Qiwen Wang¹, Qian Zhang¹, Priyaa Thavasimani³, Timothy McPhillips¹, Paolo Missier³, Bertram Ludäscher¹

¹University of Illinois, Urbana-Champaign, ²Department of Electrical and Computer Engineering, University of Illinois at Chicago, ³School of Computing Science, Newcastle University, UK

Motivation

• Data- and Workflow-Provenance are crucial for transparency and reproducibility in computational and data-driven science.

• Scientific workflow systems provide both prospective provenance (workflow graphs) and retrospective provenance (runtime observables).

Challenges

• Most computational analyses and workflows are conducted using scripts (Python, R, MATLAB, ...) rather than workflow systems.

• Retrospective Provenance Observables, from DataONE RunManagers (file-level), ReproZip (OS-level), or noWorkflow (Python code-level) only yield isolated fragments of the overall data lineage and processing history.

• Prospective Provenance could be used to link and contextualize fragments into a meaningful and comprehensible workflow, but scripts alone do not reveal the underlying workflow graph.

• Provenance (like other metadata) appears to be rarely actionable or immediately useful for those who are expected to provide it ("provenance for others").

Approach

• Simple YesWorkflow (YW) annotations allow users to reveal the workflow (prospective provenance) implicit in scripts.

• Prospective provenance queries expose and test data dependencies at the workflow level.

• Hybrid provenance queries situate runtime observables (reproducible provenance) in the overall workflow, yielding meaningful knowledge artifacts.

→ Comprehensible workflow graphs & customizable provenance reports for script runs, along with data and code in scientific studies ("provenance for self").

Run Reconstruction

• YW recon

• YW recon facts

Coarse and Fine-Grained Observations of Runs

• Matlab Run Manager → list of files input or output

• YesWorkflow → list of files matching @URI annotations

• noWorkflow → values assigned to variables

Hybrid Queries for Fine-Grained Retrospective Provenance

• Q1/Q3 (prospective): Render prospective upstream (downstream) subgraph of the YW model of the script for a given output data product D.

• Q2/Q4 (prospective): List the script inputs that are upstream (downstream) of a given data product D.

• Q5 (hybrid): Render retrospective graph with concrete filenames for a given output data product D.

Conclusions and Future Work

• Provenance from script runs can be revealed graphically and made actionable (e.g., to yield customizable data lineage reports) via (1) simple YW user annotations, (2) linking runtime observables (e.g., DataONE RunManager, ReproZip, noWorkflow), and (3) sharing provenance artifacts and executable queries.

• Extend YW to facilitate querying log files for hybrid provenance at data level.

• Extend YW toolkit to support other (optional) workflow modeling constructs (e.g., control-flow to complement dataflow); to support graph pattern queries; to support project-level provenance.

• Evolve ProvONE to support project-level provenance and graph queries.

References


• YesWorkflow Project and Tools. https://github.com/vanworkflow.org


Acknowledgments

Supported by NSF awards ACI-1430508 and ABI-1262458.