Provenance Datasets Highlighting Capture Disparities

G. Blake Coe,[+] R. Christopher Doty,[*] M. David Allen,[+] Adriane P. Chapman[+]
[+]*(gcoe, dmallen, achapman)@mitre.org* [*]*chris.doty@library.gatech.edu*
The MITRE Corporation *Georgia Institute of Technology*

Abstract

Provenance information is inherently affected by the method of its capture. Different capture mechanisms create very different provenance graphs. In this work, we describe an academic use case that has corollaries in offices everywhere. We also describe two distinct possibilities for provenance capture methods within this domain. We generate three data sets using these two capture methods: the capture methods run individually and a trace of what an omniscient capture agent would see. We describe how the different capture methods lead to different graphs and release the graphs for others to use via the ProvBench effort.

1. Introduction

There have been previous efforts to create provenance flows for testing. The First Provenance Challenge [1, 17] created a common workflow that each team ran to create provenance in their system's model. The Second and Third Challenges aimed at interoperability and querying provenance from other systems' provenance records. These efforts produced real provenance graphs that, especially in later challenges, focused on interoperability and could be exchanged and run across many systems. Additionally, the ProvBench effort [7, 8, 13] aims to collect provenance traces that can be distributed in a common model, PROV [16], for ease in testing different scenarios and styles of graphs. Finally, [3] describe a utility to create synthetic provenance graphs with specific and varying graph properties for scalability testing.

For both the Provenance Challenge and previous ProvBench efforts, the traces available for consumption are the output of only one style of capture: workflow execution traces. In other words, the traces themselves are the complete provenance graph *as seen by a particular type of capture agent.* We notice that properties of the graph (e.g., bushiness versus sparseness, number/density of agents and hand-offs involved, overall time span), type of information (i.e., attributes within a provenance node), and what nodes and edges are present vary greatly. Consider the difference between what is *capturable* in a workflow system like Vistrails [20], Taverna [21] or Kepler [6], and an OS-observing system like PASS [18]. In less granular *workflow* systems, the data files, scripts run, etc. are capturable as long as they are executed within the workflow system. In more granular *OS-observing* systems, the actual reads, writes, file opens, etc.—whether directly related to the current execution or other system maintenance—are captured. While the provenance graphs may document the same set of tasks, they are remarkably different. All other graph and data properties aside, the level of granularity of capture profoundly impacts the size and shape of the result. Yet, due to limitations in what these systems can see, equivalence often cannot be achieved by simply "rolling up" very granular information to less granular information. OS-level capture knows that a socket was opened and that data was sent to a foreign host but does not know that port 3306 on that foreign host has a database service behind it or that the data sent was an SQL query. Less granular workflow collection methods would know that a database was involved but often wouldn't be able to observe minute details such as port numbers.

These problems will be exacerbated as we try to capture provenance in more places. Efforts such as [4, 11, 14] have described mechanisms for provenance enabling many different types of applications. In general, what is required is a *capture agent* that observes and monitors a given application. However, the information available to the capture agent varies based on the application and how the agent is written, thus affecting what is actually produced and stored in the provenance graph.

In addition to the actual provenance traces, the ProvBench effort is attempting to have provenance trace

The Homeland Security Systems Engineering and Development Institute (hereafter "HS SEDI" or "SEDI") is a federally funded research and development center (FFRDC) established by the Secretary of Homeland Security under Section 305 of the Homeland Security Act of 2002. The MITRE Corporation operates SEDI under the Department of Homeland Security (DHS) contract number HSHQDC-09-D-00001.

SEDI's mission is to assist the Secretary of Homeland Security, the Under Secretary for Science and Technology, and the DHS operating elements in addressing national homeland security system development issues where enterprise, life cycle, and/or acquisition systems engineering expertise is required. SEDI also consults with other government agencies, nongovernmental organizations, institutions of higher education, and nonprofit organizations. SEDI delivers independent and objective analyses and advice to support systems development, decision-making, alternative approaches, and new insight into significant acquisition issues. SEDI's research is undertaken by mutual consent with DHS and is organized by Tasks in the annual SEDI Research Plan.

This abstract does not necessarily reflect official DHS opinion or policy.

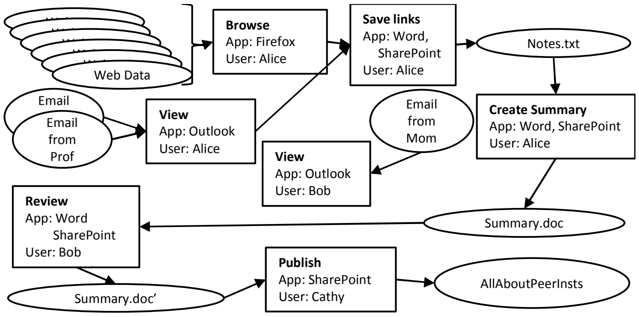

Figure 1: Sample provenance graph of the librarians preparing the requested report, from the "Complete" dataset.

contributors annotate the graphs so that they can be more useful to other researchers. ProvBench aims to distribute annotated provenance flows so that both the provenance and the actual actions from the workflow are understood within the dataset. This work attempts to add to the ProvBench effort by providing a use case and datasets that offer different views of the same set of scenarios, as seen via different capture agents. This trio of datasets is unique in that it is the same trace, stored by the same system, but under very different capture mechanisms.

2. Use Case

We have chosen a simple but common use case that allows for some variation within instances of execution; with workflows that include human actors, this will be the norm. The following explains the scenario. On behalf of the Vice Provost for Graduate Education, the Dean of Libraries at the Georgia Institute of Technology asks the Faculty Engagement Department to investigate the services offered by the Graduate Schools at Georgia Tech's peer institutions. Figure 1 shows a reduced example of what actually happened during the process. The five members of the department split the 20 peer institutions among themselves and scour the Web pages of the universities and institutes.[1] Notes are made and links pasted in a variety of formats. Files are saved on a shared drive. When everyone is done, Bob aggregates all of the information into one document and writes a summary of what was discovered. This document is shared with the team to review and make changes. The final document is then made available to the Dean via internal SharePoint to send to her fellow Vice Provost. Note that the sample may also include things like Bob's email from his mother; as provenance is a record of what happened. In some cases it may include "chaff" of marginal relevance to the workflow.

3. Capture Methods

There are several capture methods that are available for use [4]:

- Manual capture.
- Scraping of logs or wrappers for legacy systems.
- Embedded within the application. The workflow systems [6, 20, 21] provenance capture creates graphs with detailed knowledge of all processes used *within the scope of the application*. The application-based provenance capture systems, e.g., [11, 19], can only see provenance within a specific application.
- Coordination points [4]. A system like PASS [18] can see everything within the coordination point but the level of detail may not be applicable to the actual usage of provenance.

Of these, we chose to implement two: application modification on SharePoint and Firefox, and a coordination point. Using these two capture points gives very different provenance graphs. It highlights the difference between capture mechanisms and the ability to query those provenance graphs for a particular use. The PLUS system is a provenance management system that provides a basic application programming interface (API) for capture agents to publish provenance information. It then provides storage, administration, and queries over the provenance for end users. The capture methods described below merely use the PLUS reporting API to store the appropriate information.

3.1. Application-Based

Every application that is used, as well as every touch point between applications, must be provenance enabled in order to obtain a *complete* provenance graph via the application-based method. It is impractical to achieve 100 percent completeness in most non-trivial cases. In this case, we have provenance-enabled Firefox and SharePoint (which can help track the changes in Word as well). Notice in Figure 1 that because Outlook is not provenance enabled, the information coming from emails will not be a part of the provenance graph in the application-based capture scenario.

3.2. Coordination Point

A coordination point is merely an application through which a large volume of activities, applications, or data communicate. Enterprise Service Buses [4], HTTP proxies, and OSs [18] are examples. We have enabled an additional coordination point, the high-level user desktop.

[1] http://www.irp.gatech.edu/peer-institutions/

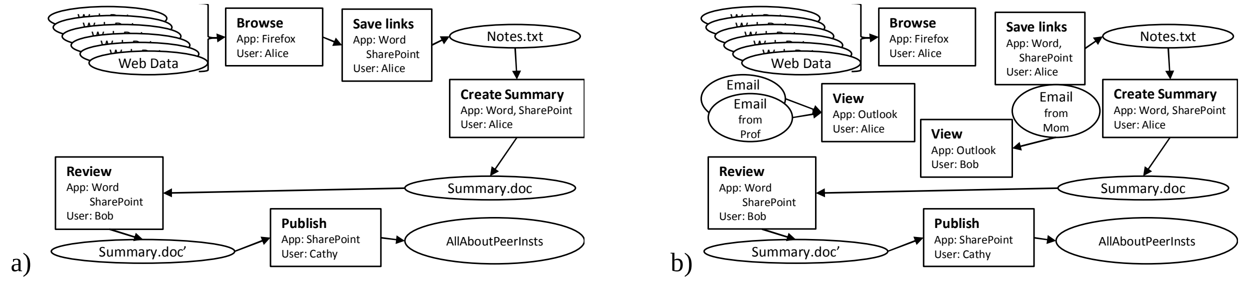

Figure 2: (a) The provenance graph relating to the provenance trace in Figure 1, from the "App-Based" dataset. (b) The provenance graph relating to the provenance trace in Figure 1, from the "Coordination" dataset.

The tool, SpectorSoft,[2] was originally created to monitor user activity on a machine, e.g., to watch children's usage or analyze employee activity. We have modified it to report provenance to our PLUS system. Unfortunately, while the coordination tool can see all of the applications and data files used on the system, creating linkages between them is particularly difficult. Because it does not have hooks deep inside the application, it is much more difficult to establish that data from one application was copied and moved to another with this one single capture approach.

4. Datasets

We have produced these datasets within the PLUS system [9], a provenance manager developed at The MITRE Corporation to address the previously unmet requirements shared by most of our U.S. government customers. There are three datasets, each containing 100 provenance traces. The three datasets are: "Complete," "App-Based," and "Coordination." The 100 graphs of each are related across datasets. That is, the first graph in "Complete" is the same scenario in App-Based and Coordination. The only difference is the information present in the provenance trace as determined by the available capture agents. Figure 2 shows the graphs from the App-Based and Coordination datasets that relate to the one depicted in Figure 1.

Notice that the same events occur in the Complete, App-Based, and Coordination dataset traces. However, because of the difference in capture methods, some nodes are absent (Outlook and emails in Figure 2(a)) as are some edges (between unrelated apps in Figure 2(b)). We attempt to provide the annotations that ProvBench seeks; instead of annotating the use case and scenario information, we provide the complete scenario as a provenance trace and then the related traces based on what is capturable given each method. The datasets are available in PROV-XML and will be released with ProvBench 2014.

The datasets have the following types of variation: number of websites used, number of websites reused, number of emails viewed, types of email viewed (work versus personal), and number of revision cycles to produce the final product. This leads to very diverse graphs in terms of density and length.

Query Workload: The intention behind these datasets is to highlight the disparities in provenance when captured across different agents. As such, while there are any number of queries that could be performed, we have chosen a query workload that highlights these disparities. Each of the queries should be run three times, once for each dataset (Complete, App-Based, and Coordination):

1. Return all websites/emails/revisions used in the creation of a final product.
2. Return average number of nodes/edges in a provenance graph.
3. Return the average provenance graph length.
4. Return the number of emails from Aunt Reba received during a work period.

5. Related Work

All provenance systems to this point have been applied to "closed world" systems and therefore are less useful for integration systems. A closed world system contains at least one of the following properties:

- The underlying application or systems are known in advance and provenance enabled.
- A provenance administrator has administrative privileges for the applications and systems in use.
- Full knowledge of either the data or processes is known in advance.

These assumptions work very well for scientific applications [5, 11, 12, 15, 20] within relational databases [10] and for specific applications [18]. However, the world of large-scale enterprises is much messier. We

[2] http://www.spectorsoft.com/

Table 1: Axis to consider for rigorous testing with a provenance benchmark

Creation	Graph Properties	Provenance Usage
·Granularity ·Number of Human Users ·Timespan ·Method of Capture ·Convergent/ Divergent Workflow	·Node Size ·Average ·Connectivity ·Data: Process Ratio ·Distance from Ideal	·Fit for Use (Single Graph) ·Workflow Compare (Multi Graph) ·Protect Graph

describe current provenance systems and then highlight the area in which their use is infeasible below.

The provenance community has two styles of testing: actual generated provenance [1, 7, 8, 13, 17] and the scalable but less empirical style presented in this work. In the database world, testing is done very differently, with a benchmarking standard that tests query workload, use cases, and scalability [2].

6. Future Work and Conclusions

The choice of capture agent(s) defines the nature and structure of the provenance graph. Because graph uses are profoundly impacted by what the graph provides, using diverse capture agents is essential for best coverage. To this end, we have generated three interrelated sets of provenance traces: Complete, App-Based capture, and Coordination Point capture. The same set of scenarios exists in each set, but with different views of the provenance information. We have released these datasets through ProvBench‘14 to facilitate future analysis on how to mitigate the effects of capture agents on the resulting graphs. Additionally, we have released the PLUS system, containing tools necessary for building capture agents at https://github.com/plus-provenance/plus. Going forward, we advocate creation of a benchmark similar to [2] for the provenance community. Just as the TPC Benchmarks are carefully crafted to test over specific loads in varying axis, such as DB query type and DB content, Table 1 shows the axis to consider while creating a benchmark specific to provenance.

7. Bibliography

[1] "Provenance Challenge" http://twiki.ipaw.info/bin/view/Challenge/, 2010.

[2] "Transaction Processing Performance Council" http://www.tpc.org/, 2013.

[3] M. D. Allen, A. Chapman, and B. Blaustein, "Engineering Choices for Open World Provenance," *Submitted to IPAW*, 2014.

[4] M. D. Allen, A. Chapman, B. Blaustein, and L. Seligman, "Provenance Capture in the Wild," *IPAW*, 2010.

[5] I. Altintas, O. Barney, and E. Jaeger-Frank, "Provenance Collection Support in the Kepler Scientific Workflow System," *IPAW*, pp. 118-132, 2006.

[6] M. K. Anand, S. Bowers, T. McPhillips, and B. Ludascher, "Efficient provenance storage over nested data collections," *EDBT*, pp. 958-969, 2009.

[7] K. Belhajjame, J. M. Gomez-Perez, and S. Sahoo, "ProvBench," 2013.

[8] K. Belhajjame, J. Zhao, D. Garijo, A. Garrido, S. Soiland-Reyes, P. Alper, and O. Corcho, "A Workflow PROV-Corpus based on Taverna and Wings," *ProvBench*, J. M. G.-P. Khalid Belhajjame, Satya Sahoo, Ed., 2013.

[9] A. Chapman, M. D. Allen, B. Blaustein, and L. Seligman, "PLUS: A Provenance Manager for Integrated Information," *IEEE International Conference on Information Reuse and Integration (IRI ‘11)*, 2011.

[10] J. N. Foster, T. J. Green, and V. Tannen, "Annotated XML: Queries and Provenance," *PODS*, pp. 271-280, 2008.

[11] J. Frew, D. Metzger, and P. Slaughter, "Automatic capture and reconstruction of computational provenance," *Concurrency and Computataion: Practice and Experience*, vol. 20, pp. 485-496, 2008.

[12] P. Groth, S. Miles, and L. Moreau, "PReServ: Provenance Recording for Services," *UK OST e-Science second AHM*, 2005.

[13] L. M. R. G. Jr., M. Wilde, M. Mattoso, and I. Foster, "Provenance Traces of the Swift Parallel Scripting System," in *ProvBench*, J. Z. Khalid Belhajjame, Jose Manuel Gomez-Perez, Satya Sahoo, Ed., 2013.

[14] P. Macko and M. Seltzer, "A general-purpose provenance library," *Theory and Practice of Provenance*, 2012.

[15] P. Missier, K. Belhajjame, J. Zhao, and C. Goble, "Data lineage model for Taverna workflows with lightweight anotation requirements," *IPAW*, 2008.

[16] L. Moreau and P. Groth, *Provenance An Introduction to PROV*, Morgan & Claypool Publishers, 2013.

[17] L. Moreau, B. Ludäscher, and et al., "Special Issue: The First Provenance Challenge," *Concurrency and Computation: Practice and Experience*, vol. 20, pp. 409-418, 2008.

[18] K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. I. Seltzer, "Provenance-Aware Storage Systems," *USENIX*, pp. 43-56, 2006.

[19] H. Park, R. Ikeda, and J. Widom, "RAMP: A System for Capturing and Tracing Provenance in MapReduce Workflows," *VLDB*, 2011.

[20] C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, and C. Silva, "Querying and Re-Using Workflows with VisTrails," *SIGMOD*, 2008.

[21] K. Wolstencroft, R. Haines, and et al., "The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud," *Nucleic Acids Research*, vol. 41, pp. w557-w561, 2013.