Cross-Platform Provenance

Ashish Gehani Dawood Tariq

SRI International *
Menlo Park, CA 94025

1. BACKGROUND

A number of systems have been developed to track workflows – for example, CMCS helps chemists document combustion research [10], $^{my}Grid$ [14] with Taverna [1] aids biologists, and ESSW is used by earth scientists [5]. Since most infrastructure developed to record the provenance of data has targeted specific fields, the projects were not easily be repurposed for different domains. The systems differed with respect to what data was captured, the types of operations performed, how the data was stored, and the kinds of queries supported. Since 2006, a community of two dozen research groups interested in data annotation, derivation, and provenance have met regularly "to understand the capabilities of different provenance systems and the expressiveness of their provenance representations," and then iteratively created an Open Provenance Model (OPM) aimed at increasing the interoperability of systems [9].

"The Open Provenance Model aims to capture the causal dependencies between the artifacts, processes, and agents" as "a directed acyclic graph, enriched with annotations capturing further information pertaining to execution." It does not "specify the internal representations that systems have to adopt to store and manipulate provenance internally", nor does it "specify protocols to store such provenance information in provenance repositories" or "protocols to query provenance repositories" [9]. Indeed, a recent effort to use MITRE's PLUS system to import, query, and visualize provenance exported in OPM format from Harvard's Provenance-Aware Storage System [11] demonstrated that OPM needed to be augmented to facilitate query interoperability [4].

2. MOTIVATION

As users begin to get access to data sets that are accompanied by provenance records, they will be faced with the challenge of analyzing metadata from external systems. Independent sources are likely to have different levels of completeness, use separate sets of identifiers to refer to the same artifacts, processes, and agents, and introduce dissimilar semantics in the annotations. To facilitate the development of cross-platform query and analysis tools, we have collected data provenance (using SPADE [6, 12]) from the same applications (Apache, BLAST, and PostMark) run on different operating systems (Linux, Mac OS X, and Windows). Since operating-system level provenance has been gathered, the records also include background activity present in the system at the time of collection. This has deliberately been included in the data set to allow query and analysis tools to have contextual provenance as well.

3. CONTRIBUTION

SPADE is the second generation of SRI's provenance collection and management system. The underlying data model used throughout is graph-based, consisting of typed vertices and directed edges, each of which can be labeled with an arbitrary number of annotations (that are key-value pairs). It includes classes for the Open Provenance Model's controlling *Agent*, executing *Process*, and data *Artifact* vertex types, and edge types that relate which process *used* which artifact, which artifact *wasGeneratedBy* which process, which process *wasTriggeredBy* which other process, which artifact *wasDerivedFrom* which other artifact, and which process *wasControlledBy* which agent.

The system completely decouples the production, storage, and utilization of provenance metadata. At its core is a novel **provenance kernel** that mediates between the producers and consumers of provenance information, and handles the persistent storage of the records. The kernel handles buffering, filtering, and multiplexing incoming metadata from multiple provenance **reporters**. It can be configured to commit the elements to multiple provenance **storage** subsystems, and responds to concurrent queries from provenance consumers. The kernel also supports modules that operate on the stream of provenance graph elements, allowing the aggregation, fusion, and composition of provenance elements to be customized with provenance **filters**.

We previously studied the use of provenance for optimizing the re-execution of applications [8]. Our ProvBench traces are from the same workloads (executed with their default settings). Table 1 summarizes the provenance traces (from Linux, Mac OS X, and Windows) collected during the compilation of the Apache Web server [3], the build of a BLAST database [2], and the execution of the PostMark filesystem benchmark [7]. The PROV concepts covered in the provenance traces are summarized in Table 2. The traces were collected over a period from 5th December, 2012 to 23rd January, 2013. They are accessible via a Wiki page [13] at the site where the code is hosted.

*This material is based upon work supported by the National Science Foundation under Grant IIS-1116414. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

EDBT/ICDT '13, Mar 18-22 2013, Genoa, Italy

	Linux	Mac OS X	Windows
Data format	Relational (Provided as compressed SQL script)		
Data model	PROV (Subset restricted to analogs of OPM elements)		
Size Apache	3.4MB	25KB	59.9MB
Size BLAST	11KB	11KB	2.8MB
Size PostMark	19KB	336KB	1.1MB
Tools used	SPADE with H2 SQL Storage and Linux (Audit) Reporter	SPADE with H2 SQL Storage and Mac OS X (OpenBSM) Reporter	SPADE with H2 SQL Storage and Windows (ProcMon) Reporter
Application domain	Apache [3]: Software compilation BLAST [2]: Data set construction PostMark [7]: Filesystem benchmark		
Provenance application	Optimizing re-execution [8]		
Possible queries	• To find all the input and intermediate files involved in compiling the Apache web server, (i) find the vertex v that has the annotation *filename:httpd*, and (ii) compute the ancestors of v that have an annotation of *type:Artifact*. • To find all the files created or modified by the application that creates a BLAST database, (i) find the vertex v that has the annotation *pidname:makeblastdb*, and (ii) compute the descendants of v that have an annotation of *type:Artifact*.		
Submission group	SPADE [12], Computer Science Laboratory, SRI International		
Contact	Ashish Gehani, SRI (ashish.gehani@sri.com)		
License	Creative Commons Attribution-ShareAlike 3.0 Unported License (CC BY-SA 3.0)		

Table 1: Provenance traces were collected from three applications (Apache, BLAST, PostMark) each run on three operating systems (Linux, Mac OS X, and Windows).

Term	Covered
prov:Activity	Y
prov:Agent	Y
prov:Entity	Y
prov:actedOnBehalfOf	N
prov:endedAtTime	N
prov:startedAtTime	N
prov:used	Y
prov:wasAssociatedWith	Y
prov:wasAttributedTo	N
prov:wasDerivedFrom	Y
prov:wasGeneratedBy	Y
prov:wasInformedBy	Y

Table 2: Coverage of PROV concepts.

4. REFERENCES

[1] M. Nedim Alpdemir, Arijit Mukherjee, Norman W. Paton, Alvaro A. A. Fernandes, Paul Watson, Kevin Glover, Chris Greenhalgh, Tom Oinn, and Hannah Tipney, Contextualised workflow execution in myGrid, *European Grid Conference, Springer-Verlag Lecture Notes in Computer Science*, Vol. 3470, 2005.

[2] Stephen Altschul, Thomas Madden, Alejandro Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David Lipman, Gapped BLAST and PSI-BLAST: A new generation of protein database search programs, *Nucleic Acids Research*, Vol. 25(17), 1997.

[3] Apache HTTP Server, `http://httpd.apache.org`

[4] Uri Braun, Margo Seltzer, Adriane Chapman, Barbara Blaustein, M. David Allen, and Len Seligman, Towards query interoperability: PASSing PLUS, *2nd Workshop on the Theory and Practice of Provenance*, 2010.

[5] J. Frew and R. Bose, Earth System Science Workbench: A data management infrastructure for earth science products, *Scientific and Statistical Database Management Conference*, 2001.

[6] Ashish Gehani and Dawood Tariq, SPADE: Support for provenance auditing in distributed environments, *13th ACM/IFIP/USENIX International Conference on Middleware*, 2012.

[7] Jeffrey Katcher, PostMark: A new file system benchmark, Technical Report TR3022, Network Appliance, 1997.

[8] Hasnain Lakhani, Rashid Tahir, Azeem Aqil, Fareed Zaffar, Dawood Tariq, and Ashish Gehani, Optimized rollback and re-computation, *46th IEEE Hawaii International Conference on Systems Science*, IEEE Computer Society, 2013.

[9] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. Van den Bussche, The Open Provenance Model core specification (v1.1), *Future Generation Computer Systems*, 2010.

[10] C. Pancerella, J. Hewson, W. Koegler, D. Leahy, M. Lee, L. Rahn, C. Yang, J. D. Myers, B. Didier, R. McCoy, K. Schuchardt, E. Stephan, T. Windus, K. Amin, S. Bittner, C. Lansing, M. Minkoff, S. Nijsure, G. v. Laszewski, R. Pinzon, B. Ruscic, Al Wagner, B. Wang, W. Pitz, Y. L. Ho, D. Montoya, L. Xu, T. C. Allison, W. H. Green, Jr., and M. Frenklach, Metadata in the collaboratory for multi-scale chemical science, *Dublin Core Conference*, 2003.

[11] Kiran-Kumar Muniswamy-Reddy, David A. Holland, Uri Braun, and Margo Seltzer, Provenance-aware storage systems, *USENIX Annual Technical Conference*, 2006.

[12] Support for Provenance Auditing in Distributed Environments, `http://spade.csl.sri.com/`

[13] ProvBench Traces, `http://code.google.com/p/data-provenance/wiki/Traces`

[14] J. Zhao, C. A. Goble, R. Stevens, and S. Bechhofer, Semantically linking and browsing provenance logs for E-science, *1st IFIP International Conference on Semantics of a Networked World*, 2004.