UTPB: A Benchmark for Scientific Workflow Provenance Storage and Querying Systems

Artem Chebotko[†], Eugenio De Hoyos, Carlos Gomez, Andrey Kashlev[‡], Xiang Lian, and Christine Reilly
Department of Computer Science
University of Texas - Pan American
1201 West University Drive, Edinburg, TX 78539-2999, USA
[†] *Corresponding author. Email: chebotkoa@utpa.edu*
[‡] *The author is currently with Wayne State University. This work was done while the author was a graduate student in the Department of Computer Science, University of Texas - Pan American.*

***Abstract*—A crucial challenge for scientific workflow management systems is to support the efficient and scalable storage and querying of large provenance datasets that record the history of in silico experiments. As new provenance management systems are being developed, it is important to have benchmarks that can evaluate these systems and provide an unbiased comparison. In this paper, based on the requirements for scientific workflow provenance systems, we design an extensible benchmark that features a collection of techniques and tools for workload generation, query selection, performance measurement, and experimental result interpretation.**

***Keywords*-benchmark; provenance; scientific workflow; performance; scalability; querying; experiment**

I. Introduction

The provenance of data generated by scientific workflows plays a central role in enabling critical eScience functionalities, including experiment reproducibility, result interpretation, and problem diagnosis. Various scientific workflow management systems (SWfMSs) support provenance collection and use their proprietary or third-party systems for provenance storage, reasoning, and querying. Provenance systems differ in a number of important ways, such as provenance models, provenance vocabularies, inference support, and query languages. Therefore, benchmarking of such systems is a challenging task.

In this work, we consider the issue of evaluating and choosing a provenance system that is capable of dealing with large provenance datasets, since scientific workflows are frequently executed multiple times in an automated fashion and can generate a large number of provenance graphs. Generally, to deal with large provenance datasets, provenance systems should comply with two basic requirements. First, such systems should use scalable and efficient techniques to store and query data. Second, provenance systems should provide efficient support for provenance-specific inference. In addition, there can be functional requirements such as supporting a particular provenance vocabulary or query type, as defined by an application context.

With respect to the above two requirements, it is currently difficult to evaluate existing systems. To consistently evaluate a provenance system in terms of scalability, provenance data in a range of sizes should be available. However, there are few such datasets available and they are usually not well-organized or documented. To evaluate a provenance system in terms of inference support, provenance data with predefined inferred results that are known to be correct and complete should be available. We are not aware of any provenance dataset that focuses on the inference aspect of provenance data management. The series of four Provenance Challenges [1], which can be considered as the state-of-the-art in scientific workflow provenance benchmarking, do not provide a testbed for evaluating system scalability and inference but rather target functional requirements, such as the expressiveness of provenance systems, their interoperability, support of the Open Provenance Model (OPM) [2], and various application issues.

As a result, we see a need for a benchmark that can facilitate the evaluation of scientific workflow provenance management systems in a systematic and unbiased manner. In this paper, our main contribution is the design of a novel benchmark that can be used to evaluate scalability and inference support of such systems. The name of our benchmark is the University of Texas Provenance Benchmark (UTPB). To address the challenge of provenance data heterogeneity, we make UTPB extensible via so-called workflow provenance templates that can be used with the benchmark to automatically generate datasets of varying sizes. UTPB 1.0 features 27 predefined provenance templates representing provenance captured for three sample workflows using three vocabularies, namely OPMV, OPMO, and OPMX, that serialize provenance according to the Open Provenance Model in RDF and XML formats. Different templates for a given workflow and vocabulary are defined to capture different workflow execution scenarios, such as successful vs. erroneous workflow runs, and raw provenance vs. provenance with completion and multi-step inferences materialized. The benchmark also supplies a provenance data generator that can generate provenance datasets based on one or more templates and includes 27 test queries organized into 11 categories. Finally, UTPB defines five performance metrics that can be used to empirically evaluate provenance systems.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents the architecture and various components of the University of Texas Provenance Benchmark. Section 4 concludes the paper and lists possible future work directions.

II. Related Work

Provenance management is recognized as an important concept in scientific workflow environments as signified by the series of four Provenance Challenges organized by the community [1]. The first Provenance Challenge started in 2006 and focused on understanding and sharing information about provenance representations and various capabilities of existing provenance systems. The second Provenance Challenge also commenced in 2006 and aimed at testing and establishing interoperability of different provenance systems by allowing them to exchange data. This event triggered an effort of the community to establish a common ground for provenance modelling and representation that later resulted in the Open Provenance Model specification [2]. The third Provenance Challenge launched in 2009 and was dedicated to evaluating various aspects of OPM. Finally, the fourth and last Provenance Challenge started in 2010 and was designed to showcase OPM in the context of novel applications that are enabled by provenance interoperability. While Provenance Challenges feature sample workflows and provenance datasets, their main focus is on benchmarking functional requirements of provenance system expressiveness, interoperability, OPM support, and OPM applications. Therefore, UTPB is complementary to Provenance Challenges and achieves the orthogonal goal of testing non-functional requirements of provenance systems, including performance and scalability of data storage, querying, and inference capabilities.

In the provenance literature, a few works [3], [4] that empirically compare provenance systems rely on either their own, ad-hoc benchmarks or benchmarks developed in other research domains (e.g., [3] uses a semantic web benchmark). To our best knowledge, UTPB is the first formally defined benchmark that targets the scientific workflow provenance domain. Yet, before designing UTPB, we surveyed benchmarks for data management systems in several domains: traditional [5], [6], [7], [8] and XML [9], [10], [11], [12] databases, semantic web and knowledge base systems [13], [14], [15], [16], and description logics systems [17], [18]. These benchmarks are not directly applicable to scientific workflow provenance due to different data models, serialization formats (e.g., provenance can be serialized in XML, RDF and as relations), and reasoning requirements. However, we got a number of insights from existing works on various aspects of benchmark design, such as query selection and performance metrics. As we discuss in the respective sections of the paper, some UTPB performance metrics and benchmarking scenarios also exist in other domain benchmarks, yet they have to be properly customized and new ones have to be introduced to reflect the requirements of the provenance field.

UTPB targets provenance systems that are used in scientific workflow environments (e.g., Taverna, Kepler, View, VisTrails, Swift, RDFProv, OPMProv, Karma, and many others). We omit details on these systems for the brevity of our presentation (e.g., see [4] for a brief survey).

III. University of Texas Provenance Benchmark

In this section, we present the benchmark architecture and provide more details for some of its components. The complete suite of UTPB tools, provenance templates, and test queries can be found at the UTPB website [19].

A. Benchmark Architecture

The UTPB architecture is shown in Fig. 1. It includes a data generator that is capable of generating datasets of varying sizes to test provenance system performance and scalability. Data is generated based on provenance templates, each of which describes the provenance of one workflow execution that is serialized according to some provenance vocabulary. Benchmark data is then fed to a test module that interacts with one or more provenance systems to load data and execute test queries in an automated fashion. Query execution results are compared to reference answers to verify their soundness and completeness. Finally, a benchmarking report is produced by the test module.

While this “ideal” architecture for evaluating different provenance systems may be fully supported by UTPB in the future, UTPB 1.0 (presented in this paper) does not supply a test module. There exist no standard or commonly accepted API for provenance management systems and therefore, interaction with each individual system requires a unique program or script to load data and execute queries.

B. Provenance Vocabularies

Almost every existing scientific workflow management system defines its own proprietary model for provenance, and each model is serialized in some format, such as RDF, XML, or relational data, according to one or more predefined vocabularies or schemas. Supporting all existing provenance vocabularies will be difficult to achieve for any provenance benchmark. However, in addition to numerous proprietary models, many systems also support the community-driven provenance model, called Open Provenance Model (OPM) [2], which was developed to provide a common layer of interoperability among existing systems. While OPM is an abstract model, there exist several vocabularies to serialize OPM provenance. UTPB 1.0 supports three of them:

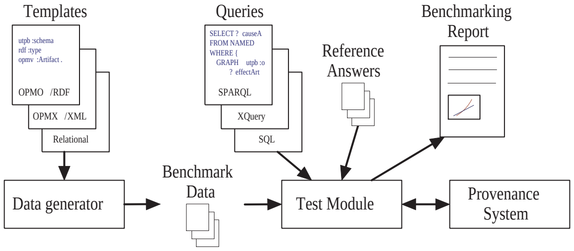


Figure 1. UTPB benchmark architecture.

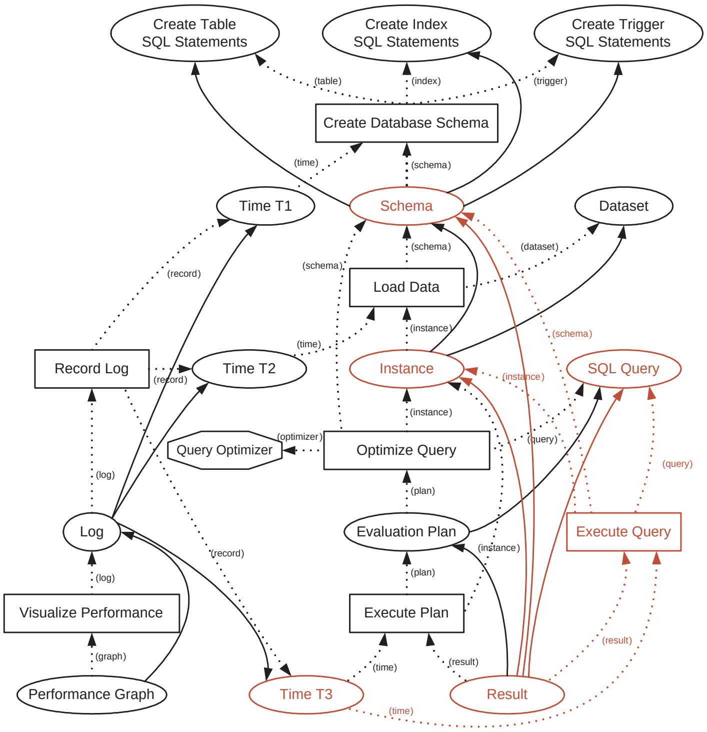


Figure 2. Provenance graph of Database Experiment (successful execution).

- The OPM Vocabulary (OPMV) is a lightweight ontology that allows serialization of most OPM features in RDF.
- The OPM Ontology (OPMO) is an ontology that extends OPMV to provide full-fledged support of OPM provenance serialization in RDF.
- The OPM XML Schema (OPMX) is a schema for XML documents that serialize OPM provenance.

Future versions of UTPB will support additional provenance vocabularies and models as they get developed and become mature, including the most recent W3C's provenance data model (PROV-DM) [20] that is in the working draft status as we write this paper.

C. *Workflow Provenance Templates*

The idea of using provenance templates, which can be cloned or instantiated as many times as needed by a data generator, came forth to overcome two issues. First, in the emerging research field, selecting and hard-coding a single workflow that can be used for provenance generation does not seem to be an adequate long-term solution because such a workflow may not include all possible features that may be of interest at present and in the future. Instead, when a particular feature needs to be tested, new provenance templates should be designed and adopted by a benchmark. Second, provenance data is heterogeneous, meaning that it can be captured using various models and represented using different vocabularies, and even the same vocabulary may provide multiple ways to state the same information. Templates enable the benchmark to overcome this challenge as new templates can be created on demand to accommodate new provenance models and vocabularies. All in all, provenance templates make the benchmark extensible and

Table I
UTPB 1.0 WORKFLOW TYPES AND THEIR CORRESPONDING TEMPLATE CHARACTERISTICS.

Workflow Name	**Template Characteristics**				
	Processes	**Artifacts**	**Accounts**	**Agents**	**Other**
Database Experiment	7	14	2	1	
Jeans Manufacturing	13	18	3	2	several processes use and generate the same artifacts and are executed in parallel
French Press Coffee	15	15	4	0	several branches with multiple processes are executed in parallel; several processes trigger each other without the record of using or generating artifacts

Table II
UTPB 1.0 TEST QUERIES.

Category		**Query**
Graphs	1.	Find all provenance graph identifiers.
	2.	Find a provenance graph with a particular identifier.
Dependencies	3.	Find all artifact derivation dependencies in a particular provenance graph.
	4.	Find all process triggering dependencies in a particular provenance graph.
	5.	Find all artifact use dependencies in a particular provenance graph.
	6.	Find all artifact generation dependencies in a particular provenance graph.
	7.	Find all controlled-by dependencies in a particular provenance graph.
Artifacts	8.	Find all artifacts and their values, if any, in a particular provenance graph.
	9.	Find all artifacts that served as initial inputs or final outputs of a workflow whose execution is described in a particular provenance graph.
Processes	10.	Find all processes and their persistent names, if any, in a particular provenance graph.
	11.	Find all processes that halted with an error in a particular provenance graph.
Accounts	12.	Find all pairs of overlapping accounts in a particular provenance graph.
	13.	Find all pairs of accounts and their refinement accounts in a particular provenance graph.
Agents	14.	Find all agents and their controlled processes, if any, in a particular provenance graph.
	15.	Find all agents that controlled two or more processes in a particular provenance graph.
Roles	16.	Find artifacts that were used with different roles in a particular provenance graph.
	17.	Find artifacts that were used with the same roles by different processes in a particular provenance graph.
Values	18.	Find all artifacts with the largest numeric value in a particular provenance graph.
	19.	Find all pairs of artifacts that were derived from each other and have the same values in a particular provenance graph.
Cross-Graph Queries	20.	Find all provenance graphs that have a common process that used all artifacts with the same values.
	21.	Find all pairs of provenance graphs whose structures match while semantics and exact values of artifacts may be different.
	22.	Find all pairs of provenance graphs that are the same structurally and semantically, such as in a case when provenance graphs were obtained by running a workflow multiple times on the same inputs.
Inferences	23-27.	Queries 3-7 with completion and multi-step inferences applied.
Application-Specific		User-defined queries, specific to an application or template.

thus adaptable to the changing requirements of the field.

UTPB 1.0 has predefined provenance templates that deal with three workflow types, three kinds of workflow execution, and three provenance vocabularies, resulting in $3 \times 3 \times 3 = 27$ templates overall. The three workflow types were selected to be easy-to-understand and effective workflow examples that feature different structural characteristics. The three workflows and their characteristics are listed in Table I.

The three execution types for each virtual workflow are successful execution, incomplete execution with an error, and successful execution with materialized provenance inferences. While the first two scenarios of success and failure represent dataset heterogeneity, the last one can be used to benchmark the inference support of a provenance system.

The graphical representation of a sample template for the successful execution of a database experiment is shown in Fig. 2. The graph follows the conventions used in the OPM Specification [2]: processes are shown as rectangles; artifacts are shown as ellipses; agents are shown as eight-sided polygons; edges represent the dependencies “used”, “wasGeneratedBy”, “wasControlledBy”, and “wasDerivedFrom”, which are easily distinguishable based on the nodes they connect (dependencies “wasTriggeredBy” are not shown as they are inferrable in this case; they explicitly exist in a graph with materialized inferences); roles are shown as edge labels when applicable; and accounts are differentiated by color.

Finally, this sample provenance graph and other graphs for different executions of the three workflows are serialized in vocabularies OPMV, OPMO, and OPMX as discussed in the previous section.

It should be noted that we created all 27 templates manually to have “clean” syntax and meaningful identifiers, however a template can also be easily created from a provenance document for a single workflow run generated by a SWfMS with only minor modifications. The main convention used in UTPB templates is that identifiers that start with an underscore (‘_’) or that do not belong to the UTPB namespace are never modified by the data generator; such identifiers are usually used to define persistent names of the same processes across multiple workflow runs.

D. Provenance Generation

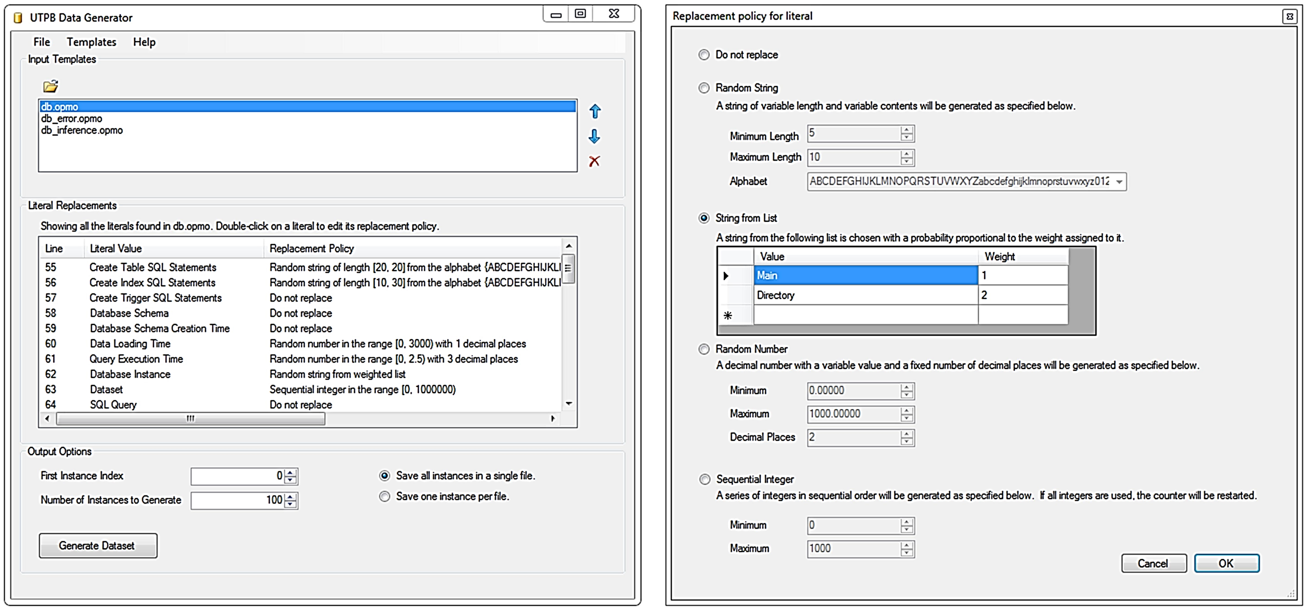
The UTPB data generator takes one or more provenance templates of the same vocabulary as input and generates a provenance dataset of desired size as output. Each template is instantiated a particular number of times as specified by the number-of-instances parameter. The process of instantiation involves cloning a template, appending an ordinal instance number to some identifiers to make them unique across different instances, and replacing some of the literals or values found in the template according to one of the five customizable replacement policies. The data generator places each template instance in a separate file or can combine them together in a single file. In addition, a dictionary file is generated with a list of all instance identifiers in a dataset. For example, OPMO and OPMV instances are named RDF graphs with graph names/identifiers also serving as instance identifiers, and OPMX instances reuse OPM graph identifiers as instance identifiers. Some of the data generator features are illustrated in Fig. 3, where the three Database Experiment OPMO templates are loaded (left screenshot) and some of the literals found in the first template are selected to be replaced using different replacement policies (right screenshot).

It is important to note that under the same parameter settings, data generation is reproducible even when random number or string replacement policies are used, which is accomplished by using the hashes of original literal values found in the input template as seeds for the pseudo-random number and string generators. To simplify dataset generation repeatability, a configuration file with all data generation settings can be saved by the application.

E. Test Queries

Since there is no standard or commonly accepted query language for provenance, we choose to define test queries in English and then provide SPARQL and XQuery versions for the respective vocabularies. To select meaningful and useful queries for UTPB, we surveyed existing provenance literature including Provenance Challenges [1] and various provenance applications. As a result, we designed 27 test queries in 11 categories presented in Table II with the last category being empty to provide extensibility for application or template specific queries. In addition to the usefulness requirement, we used two other requirements when selecting these queries: 1) they should be generic to work with different provenance templates and 2) they should provide different patterns of query complexity. The queries satisfy the first requirement as they only rely on “a particular provenance graph” identifier information. They meet the second requirement since they involve a number of diverse operations, including optional/missing values, data aggregation, operations on sets (i.e., union and difference), type conversion, data combination/joining from multiple sources, and graph pattern extraction and matching.

To illustrate how query complexity may vary, we provide SPARQL versions of queries $Q1$ and $Q8$ for both OPMV and OPMO vocabularies in Fig. 4. While both versions of the first query contain only one triple pattern and are issued over default RDF graphs with all named graph identifiers (aka dictionary), the other query has higher complexity that also varies with the vocabulary (two triple patterns and one optional clause in OPMV and six triple patterns and two optional clauses in OPMO) and is issued over the respective named RDF graphs with provenance of particular workflow executions. In Q8, the optional clauses are aimed at matching



UTPB Data Generator

Input Templates

db.opmo
db_error.opmo
db_inference.opmo

Literal Replacements

Showing all the literals found in db.opmo. Double-click on a literal to edit its replacement policy.

Line	Literal Value	Replacement Policy
55	Create Table SQL Statements	Random string of length [20, 20] from the alphabet {ABCDEFGHIJKLI
56	Create Index SQL Statements	Random string of length [10, 30] from the alphabet {ABCDEFGHIJKLI
57	Create Trigger SQL Statements	Do not replace
58	Database Schema	Do not replace
59	Database Schema Creation Time	Do not replace
60	Data Loading Time	Random number in the range [0, 3000) with 1 decimal places
61	Query Execution Time	Random number in the range [0, 2.5) with 3 decimal places
62	Database Instance	Random string from weighted list
63	Dataset	Sequential integer in the range [0, 1000000)
64	SQL Query	Do not replace

Output Options

First Instance Index: 0
Number of Instances to Generate: 100

● Save all instances in a single file.
○ Save one instance per file.

(a) Choosing provenance templates and customizing output

Replacement policy for literal

○ Do not replace

○ Random String
A string of variable length and variable contents will be generated as specified below.
Minimum Length: 5
Maximum Length: 10
Alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnoprstuvwxyz012

● String from List
A string from the following list is chosen with a probability proportional to the weight assigned to it.

Value	Weight
Main	1
Directory	2

○ Random Number
A decimal number with a variable value and a fixed number of decimal places will be generated as specified below.
Minimum: 0.00000
Maximum: 1000.00000
Decimal Places: 2

○ Sequential Integer
A series of integers in sequential order will be generated as specified below. If all integers are used, the counter will be restarted.
Minimum: 0
Maximum: 1000

(b) Setting a replacement policy for a literal

Figure 3. UTPB data generation.

an artifact value if it exists; two alternative approaches are used in the OPMO query version. For comparison, our SPARQL query for Q9 (not shown in the figure) has 10/18 triple patterns, 8/8 optional clauses, 1/1 union and 2/2 filtering operations with complex conditions when expressed over OPMV/OPMO data, respectively.

Finally, not all test queries are easily (if at all) expressible in languages like SPARQL, XQuery, and SQL as these languages were not designed for provenance querying. Therefore, existing provenance systems may not be able to answer all the queries yet. For scalability benchmarking purposes, we recommend selecting 10-15 UTPB queries with varying complexity from supported categories of interest.

F. Performance Metrics

To provide a foundation for effective provenance system evaluation and experimental results interpretation, UTPB defines five main performance metrics: data loading time, repository size, query response time, query soundness, and query completeness. These metrics are known in databases (e.g., OO1 Benchmark [7] and Wisconsin Benchmark [5], [6]) and knowledge base systems (e.g., Lehigh University Benchmark [21], [13]), however we apply several customizations that are important for the scientific workflow provenance field.

Data loading time refers to the time elapsed from acquiring a raw dataset until the moment when it completely stored into the system. This time includes any preprocessing of the dataset, such as parsing and inference precomputation. In addition to this standard metric, we define its special case – ordinal data loading time – which refers to the time required to store n-th provenance graph (template instance in UTPB) when $n-1$ graphs have already been stored and $n \geq 1$. This metric is important for provenance because SWfMSs generated provenance datasets incrementally, one graph after another, and it is crucial for a provenance system to be able to keep up with incoming storage requests.

Repository size refers to the space taken by a provenance system on a persistent storage device after a dataset has been loaded into the system. Main memory consumption is usually not measured as its accurate measurement is difficult to achieve.

Query response time measures the time elapsed from query issuance until the query result is returned and traversed, where traversal refers to the sequential access of the returned data to ensure that data (and not just a pointer or a cursor) transfer time is included in the measurement. Two special cases of this metric are cold-start time and warm-start time, where the former refers to the first query iteration after the system has been restarted and the latter refers to any subsequent query iteration when the system has a "warm" cache. For accuracy, the warm-start time should be calculated as an average of at least 10 consecutive query iterations.

Last but not least, query soundness and completeness refer to the quality of query results, which must be correct and complete. These metrics are especially useful in the presence of inference, when new data that is not part of the raw dataset is inferred based on provenance-specific

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT *
WHERE { ?graph rdf:type owl:Thing . }
```

(a) Test query Q1, OPMV version.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT *
WHERE { ?graph rdf:type owl:Thing . }
```

(b) Test query Q1, OPMO version.

```
PREFIX opmv: <http://purl.org/net/opmv/ns#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX utpb: <http://cs.panam.edu/utpb#>
SELECT ?artifact ?value
FROM NAMED <http://cs.panam.edu/utpb#opmGraph>
WHERE {
  GRAPH utpb:opmGraph {
    ?artifact rdf:type opmv:Artifact .
      OPTIONAL { ?artifact rdf:label ?value . }
  }
}
```

(c) Test query Q8, OPMV version.

```
PREFIX opmv: <http://purl.org/net/opmv/ns#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX opmo: <http://openprovenance.org/model/opmo#>
PREFIX utpb: <http://cs.panam.edu/utpb#>
SELECT ?artifact ?value
FROM NAMED <http://cs.panam.edu/utpb#opmGraph>
WHERE {
  GRAPH utpb:opmGraph {
    ?artifact rdf:type opmv:Artifact .
    OPTIONAL { ?artifact opmo:annotation ?annotation .
               ?annotation opmo:property ?property .
               ?property opmo:value ?value . }
    OPTIONAL { ?artifact opmo:avalue ?artifactValue .
               ?artifactValue opmo:content ?value . }
  }
}
```

(d) Test query Q8, OPMO version.

Figure 4. Test queries of varying complexity expressed in SPARQL for OPMV and OPMO vocabularies.

inference rules. While knowledge base systems frequently use additional metrics of degree of soundness and degree of completeness (measured in percents), we find them less useful for the provenance benchmark. Since scientific workflow provenance is used to support experiments that lead to scientific discoveries, soundness and completeness are of supreme importance - partially correct or complete query responses may lead to erroneous conclusions. Other related metrics, such as forward-chaining (precomputed) inference data loading and storage overheads and backward-chaining inference (dynamic) query response overhead, can also be defined.

G. Interpretation of Benchmark Results

For UTPB, we adapt two standard scenarios and propose three new scenarios for benchmarking provenance systems with respect to the main performance metrics.

First, different systems are compared across datasets of varying sizes with respect to a single metric. It is helpful to represent experimental results graphically as a curve chart in which the *X* axis shows increasing dataset size and the *Y* axis measures data loading time, repository size, or query response time. The shapes of these curves signify about the scalability of the systems as measurements usually increase or remain nearly constant with the dataset growth. Curves that remain closest to the *X* axis suggest better system scalability.

Second, different systems are compared using a fixed dataset with respect to a single metric. For example, query response time, query soundness and query completeness can be measured across different test queries for the same dataset. Bar diagrams (alternatively curve charts or tabular representations) with queries on the *X* axis and metric values (bar heights) on the *Y* axis can aid in understanding how different system perform on queries with varying complexity. Composite bar diagrams can be helpful to present overhead metrics in relationship to the respective metrics with no overhead and tabular representations can be most useful to visualize query soundness and query completeness.

Third, the same system or different systems are evaluated on provenance datasets that serialize the same information using different vocabularies, such as OPMV and OPMO. Different vocabularies can encode the same provenance data, but may result in different dataset sizes, which may affect data loading times and repository sizes. The same test queries can be expressed in the same querying language, such as SPARQL, according to chosen vocabularies but result in different query complexities, which may affect query response times, query soundness and completeness. As in the previous scenarios, similar curve graphs, bar diagrams and tabular representations can be used to present and interpret the results. Such an evaluation would be helpful to select an appropriate vocabulary to meet application requirements in terms of our metrics.

Fourth, different systems are compared on provenance datasets that serialize the same information using different technologies, such as RDF, XML and relational technologies. While this comparison seems natural for the provenance field in its current state, it may be one of the most difficult scenarios due to different system APIs, serialization formats, query languages, and inference capabilities. Different technologies may provide different advantages and have different drawbacks, which can be revealed through this type of benchmarking.

Last, the same or different systems are evaluated on provenance datasets that serialize the same information using different provenance models. For this type of comparison, we plan to introduce additional query expressiveness metrics to evaluate how different models can cope with different categories and types of queries.

Furthermore, hybrid evaluation approaches resulting from the above five scenarios are possible.

IV. Conclusion and Future Work

We presented the University of Texas Provenance Benchmark, which is the first benchmark for evaluating and comparing scientific workflow provenance management systems with respect to formally defined performance metrics, including data loading time, repository size, query response time, query soundness, and query completeness. We introduced the notion of provenance templates, which make the UTPB bemchmark extensible to address the challenge of provenance heterogeneity in the evolving research field. We designed 27 provenance templates that span over three workflow types, three workflow execution scenarios, and three provenance vocabularies of the Open Provenance Model. We developed a customizable data generation tool and selected 27 test queries and classified them into 11 provenance querying categories. Finally, we described a number of performance metrics and elaborated on the experimental setup and interpretation of benchmark results.

In the future, our primary goal is to further showcase UTPB via benchmarking several existing provenance systems. We will also seek to extend the benchmark with new, emerging provenance vocabularies and additional test queries. Furthermore, we plan to support additional functional metrics, such as querying expressiveness, to make the best use of the large and diverse set of test queries in the benchmark.

References

[1] *Provenance Challenge Wiki*, http://twiki.ipaw.info/bin/view/Challenge/WebHome.

[2] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. T. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. G. Stephan, and J. V. den Bussche, "The Open Provenance Model core specification (v1.1)," *Future Generation Computer Systems*, vol. 27, no. 6, pp. 743–756, 2011.

[3] P. R. Paulson, T. D. Gibson, K. L. Schuchardt, and E. G. Stephan, "Provenance store evaluation," *Technical Report PNNL-17237. Pacific Northwest National Laboratory*, 2008, retrieved from http://www.pnl.gov/main/publications/external/technical_reports/PNNL-17237.pdf.

[4] A. Chebotko, S. Lu, X. Fei, and F. Fotouhi, "RDFProv: A relational RDF store for querying and managing scientific workflow provenance," *Data & Knowledge Engineering (DKE)*, vol. 69, no. 8, pp. 836–865, 2010.

[5] D. Bitton, D. J. DeWitt, and C. Turbyfill, "Benchmarking database systems a systematic approach," in *Proc. of the International Conference on Very Large Data Bases (VLDB)*, 1983, pp. 8–19.

[6] D. Bitton and C. Turbyfill, "Readings in database systems (2nd ed.)," M. Stonebraker, Ed., 1994, ch. A retrospective on the Wisconsin Benchmark, pp. 422–441.

[7] R. G. G. Cattell, "An Engineering Database benchmark," in *The Benchmark Handbook*, 1991, pp. 247–281.

[8] *TPC Benchmarks*, http://www.tpc.org/information/benchmarks.asp.

[9] *XMark: An XML Benchmark Project*, http://www.xml-benchmark.org/.

[10] *XBench: A Family of Benchmarks for XML DBMSs*, http://se.uwaterloo.ca/~ddbms/projects/xbench/.

[11] *XMach-1: A Benchmark for XML Data Management*, http://dbs.uni-leipzig.de/en/projekte/XML/XmlBenchmarking.html.

[12] *The Michigan Benchmark*, http://www.eecs.umich.edu/db/mbench/.

[13] Y. Guo, A. Qasem, Z. Pan, and J. Heflin, "A requirements driven framework for benchmarking Semantic Web knowledge base systems," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 19, no. 2, pp. 297–309, 2007.

[14] *SWAT Projects - the Lehigh University Benchmark (LUBM)*, http://swat.cse.lehigh.edu/projects/lubm/.

[15] S. Liang, P. Fodor, H. Wan, and M. Kifer, "Openrulebench: an analysis of the performance of rule engines," in *Proc. of the International Conference on World Wide Web (WWW)*, 2009, pp. 601–610.

[16] *OpenRuleBench*, http://rulebench.projects.semwebcentral.org/.

[17] Q. Elhaik, M.-C. Rousset, and B. Ycart, "Generating random benchmarks for description logics," in *Proc. of the International Workshop on Description Logics*, 1998.

[18] I. Horrocks and P. F. Patel-Schneider, "Dl systems comparison," in *Proc. of the International Workshop on Description Logics*, 1998.

[19] *University of Texas Provenance Benchmark*, http://faculty.utpa.edu/chebotkoa/utpb.

[20] W3C, "The PROV Data Model and Abstract Syntax Notation. W3C Working Draft, 02 February 2012. L. Moreau and P. Missier (Eds.)," 2012, available from http://www.w3.org/TR/2012/WD-prov-dm-20120202/.

[21] Y. Guo, Z. Pan, and J. Heflin, "LUBM: A benchmark for OWL knowledge base systems," *Journal of Web Semantics*, vol. 3, no. 2-3, pp. 158–182, 2005.