Provenance in Dynamic Linked Data

Marcin Wylot
Linking Everything: Dynamic Graphs

➢ Integrated and summarized uncertain graph data
➢ Dynamic physical and logical network of “things”
➢ Necessity to established transparency
“Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness.”

What pieces of data and how they were combined to produce the results?
Outline

➢ Storing and tracking provenance in Linked Data [DONE]
➢ Restricting query execution with provenance data [DONE]
➢ Provenance in dynamic data [FUTUR]
➢ Provenance for performance [FUTUR]
How to store and track provenance in Linked Data processing?

➢ a new way to express the provenance of query results at two different granularity levels by leveraging the concept of provenance polynomials

➢ two new storage models to represent provenance data in a native RDF data store compactly

➢ query execution strategies to derive the provenance polynomials while executing the queries

Provenance Polynomials

"Algebraic structures for capturing the provenance of sparql queries."
Geerts, Floris, et al.

➢ Ability to characterize ways each source contributed
➢ Pinpoint the exact source to each result
➢ Trace back the list of sources the way they were combined to deliver a result
Polynomials Operators

➢ Union ($\oplus$)
- constraint or projection satisfied with multiple sources
  \[ l_1 \oplus l_2 \oplus l_3 \]
- multiple entities satisfy a set of constraints or projections

➢ Join ($\otimes$)
- sources joined to handle a constraint or a projection
- OS and OO joins between few sets of constraints
  \[ (l_1 \oplus l_2) \otimes (l_3 \oplus l_4) \]
select ?lat ?long where {
?a ?p "Eiffel Tower".
?a inCountry FR .
?a lat ?lat .
?a long ?long .
}

(\[ l_1 \oplus l_2 \oplus l_3 \] \otimes (\[ l_4 \oplus l_5 \] \otimes (\[ l_6 \oplus l_7 \] \otimes (\[ l_8 \oplus l_9 \] ))))
How can we efficiently support queries tailored with provenance information?

➢ a characterization of **provenance-enabled queries** (RDF queries tailored with provenance data)

➢ **storage model** and **indexing techniques** extensions to handle provenance-aware query execution strategies

➢ **five provenance-oriented query execution strategies**

A **Workload Query** is a query producing results a user is interested in. These results are referred to as workload query results.

A **Provenance Query** is a query that selects a set of data from which the workload query results should originate.

A **Provenance-Enabled Query** is a pair consisting of a **Workload Query** and a **Provenance Query**, producing results a user is interested in (as specified by the Workload Query) and originating only from data pre-selected by the Provenance Query.
Provenance-Enabled Query: Example

SELECT ?title WHERE {  
?a <type> <article> .  
?a <tag> <Obama> .  
?a <title> ?title . }

➢ ensure that the articles come from sources attributed to the government
SELECT ?ctx WHERE {  
?ctx prov:wasAttributedTo <government> . }

➢ ensure that the data used to produce the answer was associated a “SeniorEditor” and a “Manager”
SELECT ?ctx WHERE {  
?ctx prov:wasGeneratedBy <articleProd>.  
<articleProd> prov:wasAssociatedWith ?ed .  
?ed rdf:type <SeniorEditor> .  
<articleProd> prov:wasAssociatedWith ?m .  
?m rdf:type <Manager> . }
TripleProv: Query Execution Pipeline
Lessons Learnt

• Provenance overhead does not have to be high.

• We can leverage provenance information to improve performance.
Dynamic Linked Data

➢ Velocity

➢ Dynamic structure of the graph

➢ Incomplete data

➢ Heterogeneous environment
Continuous Provenance Polynomial

1. It has to be computed efficiently in a continuous fashion along with the execution of the query.

2. It has to take into account the missing and recovered pieces of the data.

3. It has to show how the query execution process evolves over time.
Provenance for Performance

➢ Heavy analytics

➢ Hypothetical queries

➢ Reasoning
Take Home Message

Provenance can be traced in an efficient way and can be leveraged to improve performance of query execution.