CIS 602: Provenance & Scientific Data Management

Querying Provenance and Review

Dr. David Koop
Querying Provenance

• Capturing provenance is good, capturing it efficiently is better, but really we want to do something with it
• Potential provenance queries:
  - How was a particular dataset was used?
  - How was a given dataset produced?
  - What inputs were used in creating a particular result?
  - What processes were used to transform input A to output B?
  - What time was this data processed
  - Who ran a particular step of the process?
• How do you specify these queries?
• What should the results look like (e.g. granularity)?
Reading Quiz

1. How is a user view created?
   (a) A user selects a set of relevant modules, and then an algorithm is run.
   (b) Given provenance, an algorithm automatically creates the view.
   (c) A user creates a provenance query, and the results are stored as a user view.
   (d) A user selects a number of groups of modules, and each group becomes a new module.
Reading Quiz

2. How do user views aid in querying provenance?
   
   (a) Users receive more compact results that mainly show the relevant modules they have selected.
   
   (b) User views enable the use of synthetic provenance.
   
   (c) Queries over user views require no computation because they are static.
   
   (d) User views link workflow provenance with database provenance.
Reading Quiz

3. What is a composite module?
   (a) A module that is itself a smaller workflow
   (b) A module that has multiple inputs
   (c) A module that has both multiple inputs and multiple outputs
   (d) A module that has user-defined annotations
1. Which of the following is **false**?

(a) Bioinformatics workflows often have a number of formatting tasks that are **not** central to scientific goals.

(b) User views preserve **both** dataflow and completeness with respect to dataflow.

(c) The authors performed experiments with **both** real and synthetic workflows.

(d) The `RelevUserViewBuilder` algorithm presented in the paper **always** produces the **optimum** solution.
Querying and Managing Provenance through User Views in Scientific Workflows

O. Biton, S. Cohen-Boulakia, S. B. Davision, C. S. Hara

Presented by: Avin Myneni
User Views & Querying Provenance

• How to select relevant modules?
• Efficiency of the algorithm?
  - Could a polynomial-time algorithm exist?
  - What would an algorithm that finds the optimum solution be?
• Interesting comment in the evaluation of querying:
  “The best results were obtained by the following strategy: first compute UAdmin and then remove information hidden within composite steps of the given user view.”
• Types of queries addressed (data provenance)
• How are prospective and retrospective provenance used in this technique?
Choosing a Data Model and Query Language for Provenance

Provenance Queries

• Attribute queries are certainly useful but not very different from normal data queries
• Lineage queries which deal with paths are more interesting and require some thought
ProQL

• “Querying Data Provenance” by G. Karvounarakis, Z. G. Ives, V. Tannen
• Focus on database provenance
• Use Cases for Provenance Graph Queries [from paper]
  - Q1. The ways a tuple was derived.
  - Q2. Relationships between tuples.
  - Q3. Results derivable from a given mapping or view.
  - Q4. Identifying tuples with common/overlapping provenance
• Also Use Cases for Tuple Annotation Computation
• Language that looks somewhat similar to SQL but with new clauses
Querying Provenance by Example

- Provenance is represented as graphs: hard to specify queries using text!
- Querying workflows by example [Scheidegger et al., TVCG 2007; Beeri et al., VLDB 2006; Beeri et al. VLDB 2007]
  - WYSIWYQ -- What You See Is What You Query
  - Interface to create workflow is same as to query

[Scheidegger et al.]