CIS 602: Provenance & Scientific Data Management

Provenance

Dr. David Koop
Reading Quiz
Reading Quiz

1. Which of the following best describes provenance?
   (a) The process of verifying that code performs as it is supposed to.
   (b) The process description (or sequence of steps) that, together with input data and parameters, led a data product’s creation.
   (c) A well-defined language for specifying complex tasks from simpler ones.
   (d) A mechanism that captures operating-system traces via a filesystem interface or a system call tracer.
1. Which of the following best describes provenance?

(a) The process of verifying that code performs as it is supposed to.

(b) The process description (or sequence of steps) that, together with input data and parameters, led a data product’s creation.

(c) A well-defined language for specifying complex tasks from simpler ones.

(d) A mechanism that captures operating-system traces via a filesystem interface or a system call tracer.
2. Retrospective provenance captures:
   (a) A computational task’s specification (e.g. the workflow or script).
   (b) The semantics of a set of tasks (e.g. in a biological language).
   (c) The steps executed as well as information about the environment used to derive a specific data product.
   (d) The user views of provenance, enabled via abstractions.
2. Retrospective provenance captures:

(a) A computational task’s specification (e.g. the workflow or script).
(b) The semantics of a set of tasks (e.g. in a biological language).
(c) The steps executed as well as information about the environment used to derive a specific data product.
(d) The user views of provenance, enabled via abstractions.
Reading Quiz

3. Three key components of a provenance management solution discussed in the paper are:

(a) Capture mechanisms, provenance models, and methods for storing, accessing, and querying provenance.

(b) Workflow systems, workflow evolution, and capture mechanisms.

(c) Workflow systems, OS-based systems, and Process-based systems.

(d) Provenance models, layered provenance, and normalized representation.
3. Three key components of a provenance management solution discussed in the paper are:

(a) Capture mechanisms, provenance models, and methods for storing, accessing, and querying provenance.

(b) Workflow systems, workflow evolution, and capture mechanisms.

(c) Workflow systems, OS-based systems, and Process-based systems.

(d) Provenance models, layered provenance, and normalized representation.
Reading Quiz

4. Provenance can be stored in:
   (a) Relational databases
   (b) Text files
   (c) XML databases
   (d) All of the above
   (e) (b) and (c) only
Reading Quiz

4. Provenance can be stored in:

(a) Relational databases
(b) Text files
(c) XML databases
(d) All of the above
(e) (b) and (c) only
Reading Response Example

• http://www.cis.umassd.edu/~dkoop/cis602/response.html
Reading Presentation
Provenance for Computational Tasks: A Survey

J. Freire, D. Koop, E. Santos, C. Silva

Presented by: David Koop
What is Provenance?

• Dictionary: “the source or origin of an object; its history and pedigree; a record of the ultimate derivation and passage of an item through its various owners.”

• Focus on **causality**—the sequence of steps that detail how a result was generated

• Provenance itself is **data**, this list of steps along with metadata for each step: when it occurred, who initiated it, notes about it

• Can be used to preserve information about an experiment and to answer many questions
Provenance in Science

• Provenance is as (or more) important as the result!

• Old solution:
  - Lab notebooks where detailed notes are kept

• New problems:
  - Large volumes of data
  - Complex analyses
  - Writing notes doesn’t scale

• Let computers handle this automatically!

[DNA Recombination, Lederberg]
Provenance in Science

- **Provenance is as (or more) important as the result!**
- **Old solution:**
  - Lab notebooks where detailed notes are kept
- **New problems:**
  - Large volumes of data
  - Complex analyses
  - Writing notes doesn’t scale
- **Let computers handle this automatically!**
Questions about provenance

• How does one capture provenance?
• How does one manage provenance for later use?
  - Answer questions like:
    • Who create this data product?
    • Who modified this data file?
    • What process was used to generate this data file?
    • Was the same data used to generate more than one result?
• What are the approaches and their tradeoffs?
Related Work

• Bose and Frew: “Lineage Retrieval for Scientific Data Processing: A Survey”
• Simmhan, Plale, and Gannon: “A Survey of Data Provenance in E-Science”
• Tan, “Provenance in Databases: Past, Current, and Future”
Provenance Management

- Provenance can be generated from tasks/programs/scripts/etc.
- Properties of provenance is related to the computational model
  - a specific application with a graphical interface
  - a script that automates the use of several command-line tools
  - a scientific workflow that combines several tools
Abstraction

The amount of abstraction (script, workflow, abstract workflow) impacts provenance

```python
code here
```
Prospective and Retrospective Provenance

• **Example:** Baking a Cake

• Prospective Provenance (Recipe):
  1. Gather ingredients (3/4 cup butter, 3/4 cocoa, 3/4 cup flour, ...)
  2. Preheat oven to 350 degrees
  3. Grease cake pan
  4. Mix wet ingredients in large bowl
  5. Mix dry ingredients in a separate bowl
  6. Add dry mixture to wet mixture
  7. Pour batter into cake pan
  8. Put pan in the oven and bake for 30 minutes
  9. Take cake out of oven and let it cool
Prospective and Retrospective Provenance

• Retrospective Provenance (What actually happened)

1. Went to store to buy butter
2. Gathered ingredients (3/4 cup butter, 3/4 cocoa, 1 cup flour, ...)
3. Greased cake pan
4. Preheated oven to 350 degrees
5. Mixed wet ingredients in large bowl
6. Mixed dry ingredients in a separate bowl
7. Added wet mixture to dry mixture
8. Poured batter into cake pan
9. Put pan in the oven and baked for 35 minutes
10. Took cake out of oven and let it cool for 10 minutes
Prospective and Retrospective Provenance

- Prospective provenance is what was specified/intended
  - a workflow, script, list of steps
- Retrospective provenance is what actually happened
  - actual data, actual parameters, errors that occurred, timestamps

Do not need prospective provenance to have retrospective provenance!

- Retrospective provenance the same type of information as prospective plus more

- Could have multiple retrospective provenance traces for one prospective provenance listing
Provenance & Causality

• Knowing what data/steps influenced other data/steps is important!
• Data dependencies: this output file depended on this input file
• Data-process dependencies: this output figure depended on these processes
• Causality can often be represented as a **graph** where connections represent dependencies
User-defined provenance

- Goal: capture lots of provenance automatically based on what steps are executed
- Problem: not everything can be captured automatically
- Annotations offer ability to keep notes about processes
- Users might also specify known causal links that cannot be automatically determined (e.g. a step depends on three system files that were not specified as inputs in the workflow)
Provenance Management Solution

• What is needed to capture, store, and use provenance?
  1. Capture mechanism
  2. Model for representing provenance
  3. Tools to store, query, and analyze provenance
Provenance Capture Mechanisms

- **Workflow-based**
  - Since workflow execution is controlled, keep track of all the workflow modules, parameters, etc. as they are executed

- **Process-based**
  - Each process is required to write out its own provenance information (not centralized like workflow-based)

- **OS-based**
  - The OS or filesystem is modified so that any activity it does it monitored and the provenance subsystem organizes it

- **Tradeoffs:**
  - Workflow- and process-based have better abstraction, OS-based requires minimal user effort once installed and can capture “hidden dependencies”
Provenance Models

- How provenance is represented (more abstract than the details of how it is actually stored)
- PROV (W3C Standard) has different storage backends for provenance but all of it conforms to the same model
- Model the objects involved and their relationships (e.g. activities, dependencies)
- Interoperability is a concern
  - Why? May use multiple tools/techniques to achieve a result, want to analyze the entire provenance chain
  - 2nd Provenance Challenge noted some efforts to integrate provenance from different sources
Layered Provenance

- As with relational databases, want to normalize provenance to **minimize redundant information**
- Example: Don’t store workflow specification each time that workflow is executed–store it once and reference it
- Also allow different layers for different aspects of provenance
Storing Provenance

- Files, relational databases, XML databases, RDF (linked data)
- Log files are good for preserving data but can be bad to query or analyze
- Relational databases are great for column-specific queries but can be bad for dependency queries
- XML databases are more portable than relational databases but are usually less efficient for queries
- RDF triples are better for dependencies and integrating domain-specific knowledge but can be slower
Querying Provenance

- Query methods are often tied to storage
- SQL, XQuery, Prolog, SPARQL, ...

**REDUX**

```
SELECT Execution.ExecutableWorkflowId, Execution.ExecutionId, Event.EventId, ExecutableActivity.ExecutableActivityId
from Execution, Execution_Event, Event, ExecutableWorkflow_ExecutableActivity, ExecutableActivity,
    ExecutableActivity_Property_Value, Value, Event Type as ET
where Execution.ExecutionId=Execution_Event.ExecutionId
and Execution_Event.EventId=Event.EventId
and ExecutableActivity.ExecutableActivityId=ExecutableActivity_Property_Value.ExecutableActivityId
and ExecutableActivity_Property_Value.ValueId=Value.ValueId and Value.Value=Cast('-m 12' as binary)
and ((CONVERT(DECIMAL, Event.Timestamp)+0)%7)=0 and Execution_Event.ExecutableWorkflow_ExecutableActivityId=
    ExecutableWorkflow_ExecutableActivity.ExecutableWorkflow_ExecutableActivityId
and ExecutableWorkflow_ExecutableActivity.ExecutableWorkflowId=Execution.ExecutableWorkflowId
and ExecutableWorkflow_ExecutableActivity.ExecutableActivityId=ExecutableActivity.ExecutableActivityId
and Event.EventType=ET.EventType and ET.EventType='Activity Start';
```

**VisTrails**

```
wf[*]: x where x.module='AlignWarp' and x.parameter('model')='12'
    and (log(x): y where y.dayOfWeek='Monday')
```

**MyGrid**

```
SELECT ?p
where (?p <http://www.mygrid.org.uk/provenance#startTime> ?time) and (?time > date)
using ns for <http://www.mygrid.org.uk/provenance#> xsd for <http://www.w3.org/2001/XMLSchema#>

SELECT ?p
where <urn:sid:www.mygrid.org.uk:experimentinstance:HXQOVQA2ZI0>
?inputParameter <ont:model> <ontology:twelfthOrder>.
using ns for <http://www.mygrid.org.uk/provenance#> ont for <http://www.mygrid.org.uk/ontology#>
```
Querying Provenance

- Graph-type queries can be problematic
  - Could use graph database languages or provenance-specific languages
  - Could use visual query-by-example style methods

---

Visual query canvas

Visual query

Results

---

History view

Isosurface

Isosurface Script

Volume Rendering SW

Clipping Plane SW

Combined Rendering SW

Image Slices SW

Histogram

HistogramFile

vtkStructuredPointsReader

vtkCountourFilter

vtkVolumeRayCastCompositeFunction

vtkVolumeRayCastMapper

vtkCamera

vtkVolumeProperty

vtkImplicitPlaneWidget

vtkRenderer

vtkActor

vtkPiecewiseFunction

vtkColorTransferFunction

vtkPiecewiseFunction

vtkVolumeRayCastMapper

vtkVolumeRayCastCompositeFunction

vtkVolumeProperty

vtkImplicitPlaneWidget

vtkRenderer

vtkActor

vtkPiecewiseFunction

vtkColorTransferFunction
Provenance Overload

• Can be difficult to look at large volumes of provenance
• Try to abstract the provenance as with workflows (user views)

```python
import vtk

data = vtk.vtkStructuredPointsReader()
data.setFileName("../../../examples/data/head.120.vtk")
contour = vtk.vtkContourFilter()
contour.SetInput(0, data.GetOutput())
contour.SetValue(0, 67)
mapper = vtk.vtkPolyDataMapper()
mapper.SetInput(contour.GetOutput())
mapper.ScalarVisibilityOff()
actor = vtk.vtkActor()
actor.SetMapper(mapper)
cam = vtk.vtkCamera()
cam.SetViewUp(0,0,-1)
cam.SetPosition(745,-453,369)
cam.SetFocalPoint(135,135,150)
cam.ComputeViewPlaneNormal()
ren = vtk.vtkRenderer()
ren.AddActor(actor)
ren.SetActiveCamera(cam)
ren.ResetCamera()
renwin = vtk.vtkRenderWindow()
renwin.AddRenderer(ren)
style = vtk.vtkInteractorStyleTrackballCamera()
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renwin)
iren.SetInteractorStyle(style)
iren.Initialize()
iren.Start()
```

(a)
Provenance-Enabled Systems

Table 1. Provenance-enabled systems.

<table>
<thead>
<tr>
<th>System</th>
<th>Capture mechanism</th>
<th>Prospective provenance</th>
<th>Retrospective provenance</th>
<th>Workflow evolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redux</td>
<td>Workflow-based</td>
<td>Relational</td>
<td>Relational</td>
<td>No</td>
</tr>
<tr>
<td>Swift</td>
<td>Workflow-based</td>
<td>SwiftScript</td>
<td>Relational</td>
<td>No</td>
</tr>
<tr>
<td>VisTrails</td>
<td>Workflow-based</td>
<td>XML and relational</td>
<td>Relational</td>
<td>Yes</td>
</tr>
<tr>
<td>Karma</td>
<td>Workflow- and process-based</td>
<td>Business Process Execution Language</td>
<td>XML</td>
<td>No</td>
</tr>
<tr>
<td>Kepler</td>
<td>Workflow-based</td>
<td>MoML</td>
<td>MoML variation</td>
<td>Under development</td>
</tr>
<tr>
<td>Taverna</td>
<td>Workflow-based</td>
<td>Scufl</td>
<td>RDF</td>
<td>Under development</td>
</tr>
<tr>
<td>Pegasus</td>
<td>Workflow-based</td>
<td>OWL</td>
<td>Relational</td>
<td>No</td>
</tr>
<tr>
<td>PASS</td>
<td>OS-based</td>
<td>N/A</td>
<td>Relational</td>
<td>No</td>
</tr>
<tr>
<td>ES3</td>
<td>OS-based</td>
<td>N/A</td>
<td>XML</td>
<td>No</td>
</tr>
<tr>
<td>PASOA/PreServ</td>
<td>Process-based</td>
<td>N/A</td>
<td>XML</td>
<td>No</td>
</tr>
</tbody>
</table>
Provenance-Enabled Systems

Table 1. Provenance

<table>
<thead>
<tr>
<th>System</th>
<th>Storage</th>
<th>Query support</th>
<th>Available as open source?</th>
</tr>
</thead>
<tbody>
<tr>
<td>REDUX</td>
<td>Relational database management system (RDBMS)</td>
<td>SQL</td>
<td>No</td>
</tr>
<tr>
<td>Swift</td>
<td>RDBMS</td>
<td>SQL</td>
<td>Yes</td>
</tr>
<tr>
<td>VisTrails</td>
<td>RDBMS and files</td>
<td>Visual query by example, specialized language</td>
<td>Yes</td>
</tr>
<tr>
<td>Karma</td>
<td>RDBMS</td>
<td>Proprietary API</td>
<td>Yes</td>
</tr>
<tr>
<td>Kepler</td>
<td>Files; RDBMS planned</td>
<td>Under development</td>
<td>Yes</td>
</tr>
<tr>
<td>Taverna</td>
<td>RDBMS</td>
<td>SPARQL</td>
<td>Yes</td>
</tr>
<tr>
<td>Pegasus</td>
<td>RDBMS</td>
<td>SPARQL for metadata and workflow; SQL for execution log</td>
<td>Yes</td>
</tr>
<tr>
<td>PASS</td>
<td>Berkeley DB</td>
<td>nq (proprietary query tool)</td>
<td>No</td>
</tr>
<tr>
<td>ES3</td>
<td>XML database</td>
<td>XQuery</td>
<td>No</td>
</tr>
<tr>
<td>PASOA/PreServ</td>
<td>Filesystem, Berkeley DB</td>
<td>XQuery, Java query API</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Conclusions

• Provenance solutions have many benefits to scientists and engineers
• Capture mechanisms seem well-developed
• Storage, interoperability, and querying methods are still being improved
• Different types of systems have different benefits and challenges
• Sharing provenance can help other people better understand exactly how an experiment was done or how an application works
Discussion

• How might workflow-based and OS-based provenance systems be integrated or used concurrently?
• Are current data storage/query tools good enough for provenance or do provenance-specific solutions need to be implemented?
• What other uses of provenance (beyond causality, checking steps) are there?

• Other questions?
Reading Presentation

- Should be 15-25 minutes
- Should cover the whole paper, explaining the material presented
- **Do not simply copy/quote the paper!**
- **Look up references** if something is unclear and needs to be explained
- Detail **points of discussion**, potential issues with the approach
- Insert your own **viewpoints** into the presentation
- If figures or images from the paper are useful, please include them in the presentation
  - May also create your own diagrams if they help explain something
Course Project

• Many questions about examples of projects, language or system requirements

• Goal is to create something new relating to provenance, scientific workflows, or scientific data management

• No language requirement, but I need to be able to run the code on a standard computer unless we agree upon something else

• I will allow groups of 2, but I will expect more projects that are twice as involved as individual projects. I will also expect equal participation from both group members in presentations as well as reports.

• Will require a proposal, progress report, and final report
  - I will give feedback about my impression of your project so you know what is expected for the final project and report
Course Project Ideas

- Add meaningful provenance capture to an existing tool that currently lacks it. Such a project could use a custom solution or capture provenance in an existing format (like PROV or using the VisTrails SDK). For example, you might use the VisTrails SDK to store and replay provenance from an application like a web browser. ([http://www.vistrails.com/sdk.html](http://www.vistrails.com/sdk.html))
Course Project Ideas

VisTrails Plugin for ParaView
Course Project Ideas

• Wrap an existing library for use in a scientific workflow system (e.g. Kepler, VisTrails, Taverna) and create new useful workflows. You should then be able to create provenance by running those workflows.
Course Project Ideas

• Create a new technique for analyzing, visualizing, or querying provenance using existing provenance (e.g. the ProvBench datasets: https://github.com/provbench). For example, you might design a new visualization technique for organizing similar provenance traces in an efficient manner.
Course Project Ideas

• Use a scientific or graph database to store and query information that is used in a tool. For example, you might use neo4j to support a recommendation system for movies based on a social network.
Course Project Ideas

- Some other project of your choice that relates to provenance, workflows, scientific data management.
Next Class (9/11/2014)

• New paper: Scientific Workflow Management and the Kepler System (download from course web page)
• Send Reading Response to dkoop@umassd.edu by 12pm
  - Text or PDF (no Word documents, please)
• Keep thinking about course project ideas
  - Please stop by during office hours if you want to discuss a project idea
• I will be asking for preferences for reading presentations soon. I will email the class on the procedure.