Provenance-Aware Storage Systems

The First Workshop on Provenance Aware Storage Systems
October 20, 2005
Margo Seltzer
Logistics

- Morning and keynote in this room.
- Afternoon sessions in 123 (up one floor).
- We’ll ask for volunteers to scribe for the three sessions.
- Need anything: ask Tristen Hubbard
- Please stay for HIP tomorrow if your schedule permits.
Provenance

- From the French word for “source” or “origin”
- The complete history or lineage of an object
- In computer terms:
  - On what is this object based?
  - How was this object created?
  - How can it be re-created?

Examples
- Source code control
- make
Applications of Provenance

- Homeland security
- Archival
- Science
- Business compliance
- Software development
- Publishing
The State of Provenance Today

- Most provenance is entered manually.
- Provenance is stored in a database, separately from the data it describes.
- Many provenance systems are domain-specific.
- In many fields, provenance support is simply lacking.

There must be a better way!
Outline

• What is Provenance?
• What is a Provenance-Aware Storage System?
• PASS Architecture
• PASS Prototype
• Research Challenges
• Conclusions
Observations

• Provenance is meta-data
• Storage systems manage meta-data
  – Low-level layout information
  – Access control lists
  – Modify/access times
  – Generation numbers
• Some of that meta-data is provenance

Why not make the storage system responsible for provenance?
Advantages of Provenance in the Storage System

- Generate provenance automatically.
- Requires effort to subvert the provenance system.
- Data and its provenance can be tightly bound.
- System-level provenance facilitates reproducibility.
- Provenance can subsume or augment document management systems.
What is a PASS?

• Storage systems (e.g., file systems) in which provenance is a first class entity.
• Provenance:
  – is generated and maintained as transparently as possible.
  – can be indexed and queried.
  – will be created from objects imported from non-PASS sources.
  – is maintained in the presence of deletes, copies, renames, etc.
Limitations of Automatic Provenance

• (For now) Assume that all input comes from objects that reside in a PASS.

• With no human intervention, it is possible to automatically construct the provenance for any objects produced/created.
  – Removes any source of inconsistency
  – Guarantees that all objects are “provenanced”

• Some data does not come from objects in PASS.
  – What is provenance of a file into which you type?
  – What is the provenance of file generated by a measurement device (e.g., telescope, medical sensor, router)?
Index and Query

• Users will want to query provenance
  – Show me everything derived from my file
  – Show me everything upon which I depend
• Provenance schema is not fixed
  – My experiment will have different parameters from yours; parameters are part of the provenance of the result.
  – Applications and users may care about provenance that is not visible to the system.

The challenges of manually created data, index, query, and application-specific provenance require an extensible data management approach.
Degree of Automation

- Data that come from PASS files is easy.
- What about data from user input? Or devices?
- Proposed Solution: three types of provenance
  - Fully automatic: the easy stuff
  - Semi-automatic: provenance that can be deterministically extracted from a data source.
  - Manual: grab what you can; allow users to provide annotations as well.
Extensibility via annotations

• How do you capture provenance for GUI input?
  – Capture mouse clicks?
  – Do nothing?
  – Allow provenance-aware applications?

• Similar challenges arise in “environments”
  – R
  – Matlab

• Proposed Solution: permit additions to provenance from applications and users. Call these annotations.
Outline

• What is Provenance?
• What is a Provenance-Aware Storage System?
• PASS Architecture
• PASS Prototype
• Research Challenges
• Conclusions
Our Definition of Provenance

• Attribute-value pairs.
• Some attributes are standard (OS, CPU).
• Some attributes are application-specific (e.g., a particular experiment has a value for nParticles).
• Some attributes may be domain-specific (e.g., all LHC experiments have a particle type).

Attribute set must be extensible
PASS Components

From User Process

Collector

From Network

Control

Data

Query Tool

Provenance Records

Schema Access Control

Storage System
The Collector

- Monitor process event stream and translate it into provenance event stream.
- Transmit provenance events to the storage system for safe-keeping.
- What process events are relevant:
  - Process management: fork, exec
  - File I/O: close, read, write
  - Other information flow: shmap, mmap
- Handle trickiness
  - Cycles detection and elimination or cycle avoidance
  - When provenance gets written
The Schema Layer

- Implements provenance access control.
- Provides a schema from which to query.
- Provides a general-purpose schema on top of which domain-specific systems can be constructed.
The Storage System

- A file system: store data as would a non-PASS.
- Store provenance and its metadata:
  - schema
  - indexes
  - access control information
  - the actual provenance
- Guarantee integrity and coordination of data and provenance.
Miscellania

• Early experience with our prototype suggests:
  – The collector and schema should coordinate so that users can create scripts that transform some input into another input.
  – The query tool and schema should coordinate to limit the depth of ancestry/descendancy returned.
  – Since provenance is inherently forest-structured, visual display is as important (or more so) than textual display.
  – Performing in-kernel cycle detection and eradication is complex and ugly; avoidance would be a nice approach.
Outline

• What is Provenance?
• What is a Provenance-Aware Storage System?
• PASS Architecture
• PASS Prototype
• Research Challenges
• Conclusions
Implementation Platform

- Linux 2.4
- In-kernel transactional data store
  - Port of Berkeley DB into the kernel.
  - Provided by SUNY Stonybrook.
- Provenance And STorAge layer: PASTA
  - Stacked file system
  - Constructed using FiST
Architecture to Implementation

- From User Process
- From Network
- Query Tool

Intercepts At Syscall Layer

Data
Control

Provenance Records

KBDB

PASTA
Terminology

- **Pnode**: provenance node
  - An in-memory structure that represents all provenance on an object.
- **Pnode number**
  - A unique identifier for a pnode.
  - All on-disk provenance is tagged by pnode number.
Record Types

- NAME: full path name
- INPUT_FILE: pnode number of a file that was read
- PREV_VERSION: if it exists, the pnode number of a previous version of an object.
- ARGUMENTS: reference to a command line
- ENVIRONMENT: reference to an environment description
- SYSTEM: reference to a system description
- PROC_NAME: name of a process that wrote to an object.
- PID: pid of a process writing to an object.
Schema

- Five BDB Databases
  - Provenance: keyed by a pnode number and record type; data is value for that record type (duplicates OK).
  - Map: keyed by inode number; data is pnode number.
  - Argdata: collection of all command lines (keyed by unique ID)
  - Cmdindex: inverse mapping of argdata (key is command line; data is unique ID)
  - Argindex: index on uses of a provenanced object (keyed by command line options; data is pnode number)

  \[ \text{lpr -d missmuffet file} \]
  Assume file's pnode number is 7: \{lpr,7\},\{-d,7\}, \{missmuffet,7\},\{file 7\}
Query Tool

- C-program
- Directly reads KBDB tables
- Supports
  - Ancestry queries
  - Trivial script generation
Notable Missing Features

- Access control model for provenance
- Proper tagging of files obtained from a network connection
- Full and complete pipe handling
- System provenance
- General purpose query tool
- Management and configuration tools
- Annotation support
Outline

- What is Provenance?
- What is a Provenance-Aware Storage System
- PASS Architecture
- PASS Prototype
- Research Challenges
- Conclusions
Research Challenges (1)

• Security: what is the right security model of provenance (and how do you expose that model to the user)?
• Search: can we do better than general-purpose search? How?
• APIs: how do we export provenance to support applications?
• Schema: how do we construct a substrate on to of which one can easily construct domain-specific systems?
• The Wire: how do we implement distributed PASS?
• Pruning: when can we delete provenance?
Research Challenges (2)

- Evaluation: how do we evaluate PASS? To what do we compare? What are the relevant metrics?
- Distributed Search: If you want to perform global search across a massive collection of PASS’s, what do you do? Ship meta-data? Ship queries?
- Versioning: Does it make sense to build PASS on a non-versioning file system?
- Your Good Areas go here!
Outline

• What is Provenance?
• What is a Provenance-Aware Storage System
• PASS Architecture
• PASS Prototype
• Research Challenges
• Conclusions
Conclusions

- The “Provenance Problem” arises in many domains.
- There is sufficient commonality among the domains that we should be building a common infrastructure on which to develop domain-specific approaches.
- Addressing the research challenges will be stimulating.
- Building this will be fun!
Workshop Goals

• Build a community.
• Discuss in depth some of the important challenges.
• Identify other fertile research areas.
• Decide when to meet again.
  – Pick dates
  – Suggest names for other attendees
  – Talk up the event