Big Data Provenance - Challenges and Implications for Benchmarking

Boris Glavic

IIT DBGroup

IIT College of Science and Letters
ILLINOIS INSTITUTE OF TECHNOLOGY

December 17, 2012

Outline

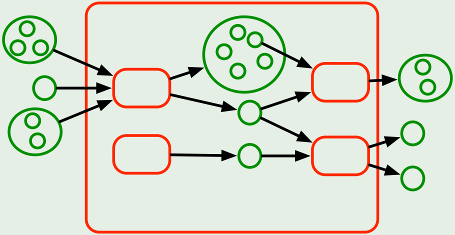
Disclaimer

- My previous benchmark experience
  - Limited to being a user
- My main background is in provenance
  - ... mostly for RDBMS
- The main point of this talk (visions)
  - Provenance as a benchmark use-case
  - Provenance as a supporting technology for benchmarking
- I will make some outrageous claims to get my point across

What is Provenance?

- Information about the creation process and origin of data
- **Data items** and collections
  - Data item = atomic unit of data
- **Transformation**
  - Abstraction for processing
  - Inputs and outputs are data items (collections)

Example



Types of Provenance

- Given a data item *d*

Data Provenance

- Which data items where used in the generation of *d*
  - Representation: e.g., a set of data items

Transformation Provenance

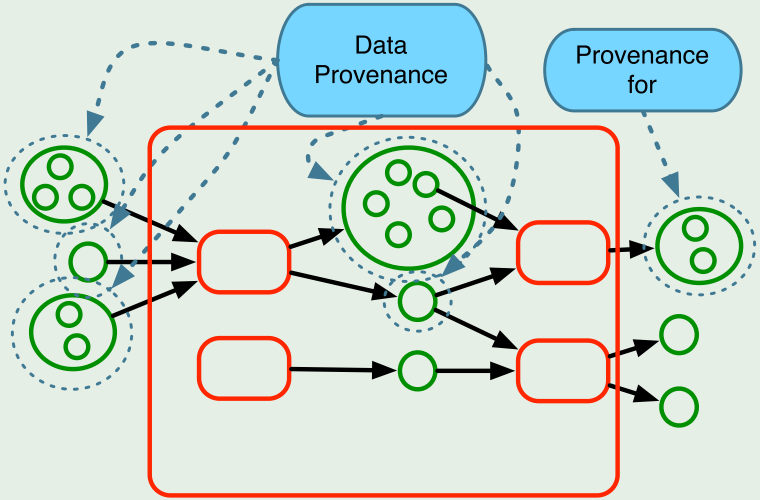
- Which transformations generated *d*
  - transitively

Additional Information

- Execution Environment
- User
- Time
- ...

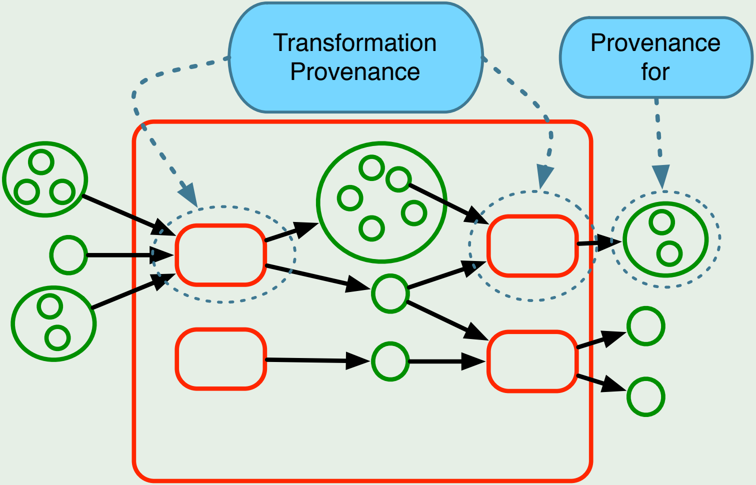
Data and Transformation Provenance

Example (Data Provenance)



Data and Transformation Provenance

Example (Transformation Provenance)



Coarse-grained vs. Fine-grained Data Provenance

Coarse-grained Provenance

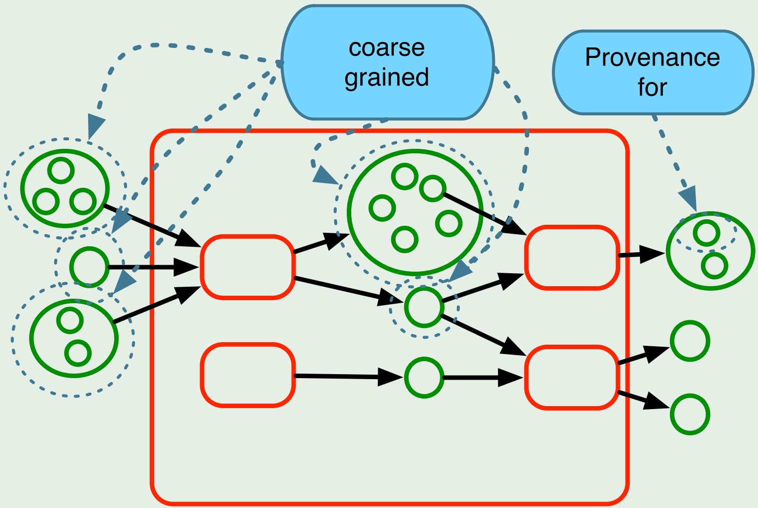
- Transformations are handled as black-boxes
- $\Rightarrow$Each output of transformation
  - ... depends on all inputs

Fine-grained Provenance

- Consider processing logic of transformation to determine data-flow
- $\Rightarrow$All data items that contributed to a result
  - Sufficiency: sufficient to derive d through transformation
  - Necessity: necessary in deriving d through transformation

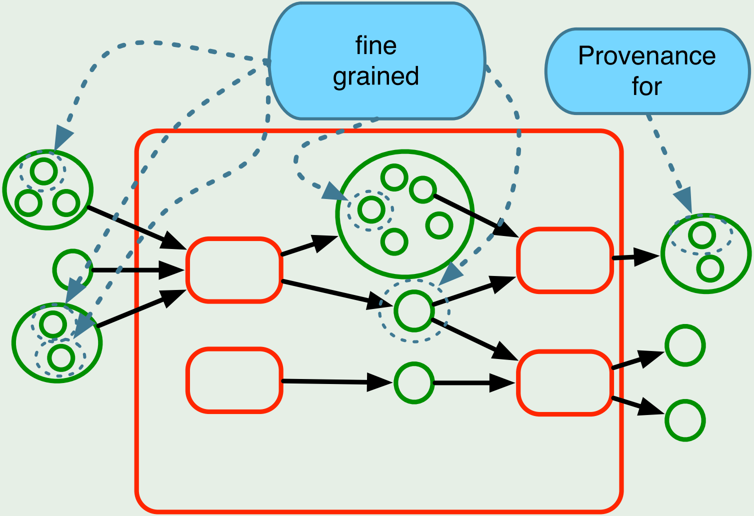
'

Example (Coarse-grained)



Granularity

Example (Fine-grained)



Use-cases

- Data debugging
  - E.g., tracing an erroneous data item back to erroneous inputs
- Trust and Quality
  - E.g., computing trust based on trust in data items in provenance
- Probabilistic data
  - Computing probability of result based on probabilities in provenance
- Security
  - Enforce access-control on query results based on provenance
- Understanding misbehaviour of systems
  - E.g., detecting security breaches

Annotations

Provenance as Annotations

- Model provenance as annotations on the data
- $\Rightarrow$Provenance management $=$
  - Efficient storage of annotations
  - Propagating annotations through transformations

Example (Systems applying this approach)

- Perm
- Orchestra
- DBNotes
- Trio
- ...and many more

Annotations

References

Deepavali Bhagwat, Laura Chiticariu, Wang-Chiew Tan, and Gaurav Vijayvargiya.
An Annotation Management System for Relational Databases.
VLDB Journal, 14(4):373–396, 2005.

Jennifer Widom.
Trio: A System for Managing Data, Uncertainty, and Lineage.
Managing and Mining Uncertain Data, 2008.

Boris Glavic and Gustavo Alonso.
Perm: Processing Provenance and Data on the same Data Model through Query Rewriting.
ICDE, 174-185, 2009.

T.J. Green, G. Karvounarakis, and Z.G.I.V. Tannen.
Provenance in ORCHESTRA.
IEEE Data Eng. Bull., 33(3):9-16, 2010.

Size Considerations

- Provenance models dependencies between inputs and outputs of a transformation
  - $\Rightarrow$A subset of Inputs $\times$ Outputs
  - $\Rightarrow$Can be quadratic in size
- DAG of transformations
  - multiply by maximal path length
- Provenance of two data items often overlaps
  - Reuse of intermediate results
  - Coarse-grained provenance

'

Outline

Big Provenance Challenges

- Big data analytics is all about agility
  - $\Rightarrow$un/semi-structured data and no extensive meta-data available
  - $\Rightarrow$hard to define provenance model
- Transparency of distribution
  - Provenance use-case may require location information
- Analytics that cross system boundaries
  - Inter-operability of storage formats and systems

Example - Provenance for Map-Reduce Workflows

- Fine-grained provenance for workflows that are DAGs of map and reduce functions
- Add provenance to values: $(key, value) \rightarrow (key, (value, p))$
- Provenance determined by map/reduce function semantics
- Wrap map and reduce functions to handle provenance
  - Strip of and cache provenance
  - Call original function
  - Re-attach provenance according to function semantics

References

R. Ikeda, H. Park, and J. Widom.
Provenance for generalized map and reduce workflows.
CIDR, 273-283, 2011.

Example - OS Provenance for the Cloud

PASS with Cloud Storage Backend

- File and process level provenance
- Each node runs a modified linux kernel (PASS)
- Intercept system-calls for file and process creation
- Store provenance in Amazon S3, SimpleDB, and/or SQS

Instrumenting the Xen Hypervisor

- File and process level provenance for virtual machines
- Intercept “hyper-calls” for file and process creation
- Store provenance in DB

Example - OS Provenance for the Cloud

References

M.I. Seltzer, P. Macko, and M.A. Chiarini.
Collecting provenance via the xen hypervisor.
TaPP, 2011.

K.K. Muniswamy-Reddy, P. Macko, and M. Seltzer.
Provenance for the cloud.
FAST, 15–14, 2010.

Example - Sketching Distributed Provenance

- File + process level provenance modelled as DAG
  - intra- and inter-host dependencies
  - inter = socket communication
  - Intercept system calls
- Distributed provenance graph storage
  - Each node stores the part of the provenance graph corresponding to its local processing
  - Links to other host for inter-host dependencies
- Query provenance
  - Nodes exchange summaries of provenance graphs using bloom filters

References

T. Malik, L. Nistor, and A. Gehani.
Tracking and sketching distributed data provenance.
eScience, 190–197, 2010.

Take-away Message

- Challenging problems
- Approaches that address distribution directly
- Approaches that map relational techniques to Big Data
- $\Rightarrow$more work needed!

Outline

Why Benchmark?

Impress the Customer - Competition

- Competition should be fair
  - Stable benchmark
  - Similar to real world workloads
  - Precise definitions
  - $\Rightarrow$probably several iterations of design + test

Understand (and Improve) System Performance

- Stable benchmark
- Similar to real world workloads
- Profiling and monitoring

Benchmarking 101

A Comprehensive Benchmark . . .

- Define Parameters
- Data-set specifications
  - Structure and interrelationships
  - Value ranges and distributions
  - Data type definitions
  - Provide data generator and/or validator
- Workload specification
  - Specify jobs
  - Restrictions for running the jobs
  - Provide workload generator and/or validator
- Benchmark metrics specification
  - What to measure
  - How to measure

Benchmarking 101

A Comprehensive Benchmark ... TPC-H

- Define Parameters e.g., SF
- Data-set specifications Standard Specification
  - Structure and interrelationships Schema provided
  - Value ranges and distributions e.g., `L_DISCOUNT` column values between 0.00 and 1.00
  - Data type definitions e.g., date is YYYY-MM-DD at least 14 years range
  - Provide data generator and/or validator dbgen
- Workload specification Standard Specification
  - Specify jobs Query and update workloads
  - Restrictions for running the jobs
  - Provide workload generator and/or validator qgen
- Benchmark metrics specification Standard Specification
  - What to measure e.g., Composite Query-per-Hour Metric
  - How to measure e.g., first query char to last output char

Big Data Benchmarking

- Data generation is a necessity
  - Shipping pre-generated datasets of Big Data dimensions is unfeasible
- Complex and mixed workloads better match real-world use of Big Data analytics
  - Hard to understand why a system performs bad/well on a complex workload
- Robustness of performance and scalability important
  $\Rightarrow$Measure it!
  - What is a good metric for robustness and how to compute it?

Generating Large Datasets and Workloads

- Provenance data is large
- Compute provenance to generate large datasets
  - Run provenance-aware system to collect provenance for a small workload
  - The resulting data-set can be used in the benchmark
- $\Rightarrow$Generate large data-sets from simple “generators”

Example

- Simple task: sharpen an image
- Apply approach that instruments the program to detect data-dependencies as provenance
- Huge and very fine-granular provenance

Stress-Testing Exploitation of Data Commonalities

- Provenance data large, but large overlap
- ⇒Use provenance to test how well a system is able to exploit data commonalities

Example

- Streaming data with multi-step windowed aggregation
- ⇒Can predict overlap in provenance
- Generate provenance
- Queries over provenance as workload

Data-centric Performance Information as Provenance

- Assume the hypothetical existence of **god**
  - Big provenance system
  - Efficiently computes all types of provenance
  - For any Big Data system
  - Also evaluates any types of queries

Data-centric performance information as provenance

- For each data item use **god** to record
  - execution times of each transformation in provenance
  - history of data movements for each data item
  - ...

Example (Measure Robustness)

- Benchmark that requires several runs of mixed job workloads
- Workloads between runs overlap
- Performance metric: Resource consumption variation per job

Using Provenance in Profiling

Profiling and Monitoring

- Identify causes for (benchmark) results
- Even more important for Big Data
  - Diverse workloads
  - Distribution
  - ...

Example (How can Provenance help?)

- How do I profile a program on one node
  - e.g., instrument and collect statistics about execution
- Provenance provides data-centric view
  - Identify repeated computations
    - Overlaps in provenance (same data, different nodes/transformations)
  - Identify unnecessary computations
    - Computations that are not in the fine-grained provenance of anything

Outline

Conclusions

- Big Provenance: Many interesting research problems
- Provenance as benchmark use-case
- Provenance for benchmark data generation
- Provenance-based performance metrics
- Supporting profiling and monitoring

Questions?

Info

- **Boris Glavic**: `http://www.cs.iit.edu/~glavic/`
- **IIT DBGroup**: `http://www.cs.iit.edu/~dbgroup/`
- **Perm**: `http://permdbms.sourceforge.net/`

IIT College of Science and Letters
ILLINOIS INSTITUTE OF TECHNOLOGY

COMPUTER SCIENCE