SMOKE: A Fast Record-Level Provenance System for Interactive Applications

Fotis Psallidas  
fotis@cs.columbia.edu

Eugene Wu  
ewu@cs.columbia.edu

Introduction
Provenance is at the core of many applications: visualization, debugging, diagnosis, profiling, why-not, interpretability, …

Overhead of provenance capture can cripple query engine performance.

Smoke introduces principles to reduce overhead from >100x slowdown to ~1.3x slowdown.

What is provenance?
A graph connecting inputs with outputs across a workflow:

Provenance (or Lineage) Graph

Capture Overhead (GB)

Crossfilter Performance

Example: Group-By
Two Provenance Capture Paradigms

SMOKE Overview
4 Design Principles

P1. Tight Integration
Instrument operators to write lineage idxs

P2. Reuse work
Lineage indexes ~ Hash tables

P3. Apriori Knowledge
Don’t capture it if not used

P4. Lineage Consumption
Push computation into lineage capture

Example

Conclusion
Provenance capture can be
• Fast enough to be interactive
• Competitive with hand-written implementations
• Useful for data-intensive interactive apps

Interested in More?
[VLDB18] Smoke: Fine-Grained Lineage At Interactive Speed
[SIGMOD18] A Deep breath of Lineage Applications
[HILDA18] Provenance for Interactive Visualizations
[CIDR17] Combining Design and Performance in a DVMS