Influence-Based Provenance for Dataflow Applications with Taint Propagation

Jason Teoh¹, Muhammad Ali Gulzar², Miryung Kim¹
¹University of California, Los Angeles, ²Virginia Tech

Problem Statement
What inputs are responsible for producing suspicious output?

Key Insight
FlowDebug improves provenance precision by tracking input contribution within UDFs.

Novelty 1: UDF Tainting
UDF-Aware Tainting and Influence Functions can be used together to improve provenance trace precision.

Novelty 2. Influence Function
FlowDebug extends Spark’s combineByKey API with Influence Functions to define flexible, user-defined provenance.

Current State of the Art
• Data Provenance: Trace the movement of records through operators (e.g. Aggregate)
• Delta Debugging: Use an output test function to guide binary search reduction of input space.

Evaluation Results
Comparisons against Titian (Provenance), BigSift (Delta Debugging), and Spark (baseline)

[RQ1] Precision
• 15-100% precision improvement vs Titian
• 96.8-99.3% recall improvement vs BigSift

[RQ2] Instrumentation Overhead
• 5-8X faster with Influence Functions
• 50% overhead with UDF-Aware Tainting
• 0.4-6.1X overhead vs Spark

[RQ3] Tracing Time
• 12-73X, 374-1506X faster than Titian and BigSift
• Tracing at most 25% of total job