Scaling SPADE to "Big Provenance"

Ashish Gehani Hasanat Kazmi Hassaan Irshad

SRI



SPADEv1 (2008-2009)

- Certification / verification of file lineage
  - Metadata replication (for availability)
  - Verification reordering (for performance)
  - Causality witnesses (to avoid clocks)
- Scalability issues
  - Collection, storage tightly coupled
  - Static architecture
  - Fine-grained cryptographic protection
  - Provenance propagated with files
    - Latency, storage overhead
- Motivated rewrite in 2010

What is SPADEv2?

- Open source middleware
- *Reporters*:
  - OS (Linux, OS X, Windows, Android)
  - Compiler (LLVM)
  - Imports (in DSL, JSON, Graphviz)
  - Bitcoin
- *Storage*:
  - Graph (Neo4j, Graphviz)
  - PROV-O, PROV-N
  - Kafka, SQL, Datalog (In progress)

Case Study: Bitcoin

- Market crossed $1B in 2013
- Accepted by 75,000 companies in 2014
- *Blockchain* is public append-only ledger
- *Blocks* of *mined* (verified) *transactions*
- Transaction of incoming, outgoing *payments*
- Provenance queries:
  - All payers in ancestors
  - All payees in descendants
  - Financial flows via all paths
- 99% blocks $\rightarrow$ 522M vertices, 1042M edges

Case Study: Forensics

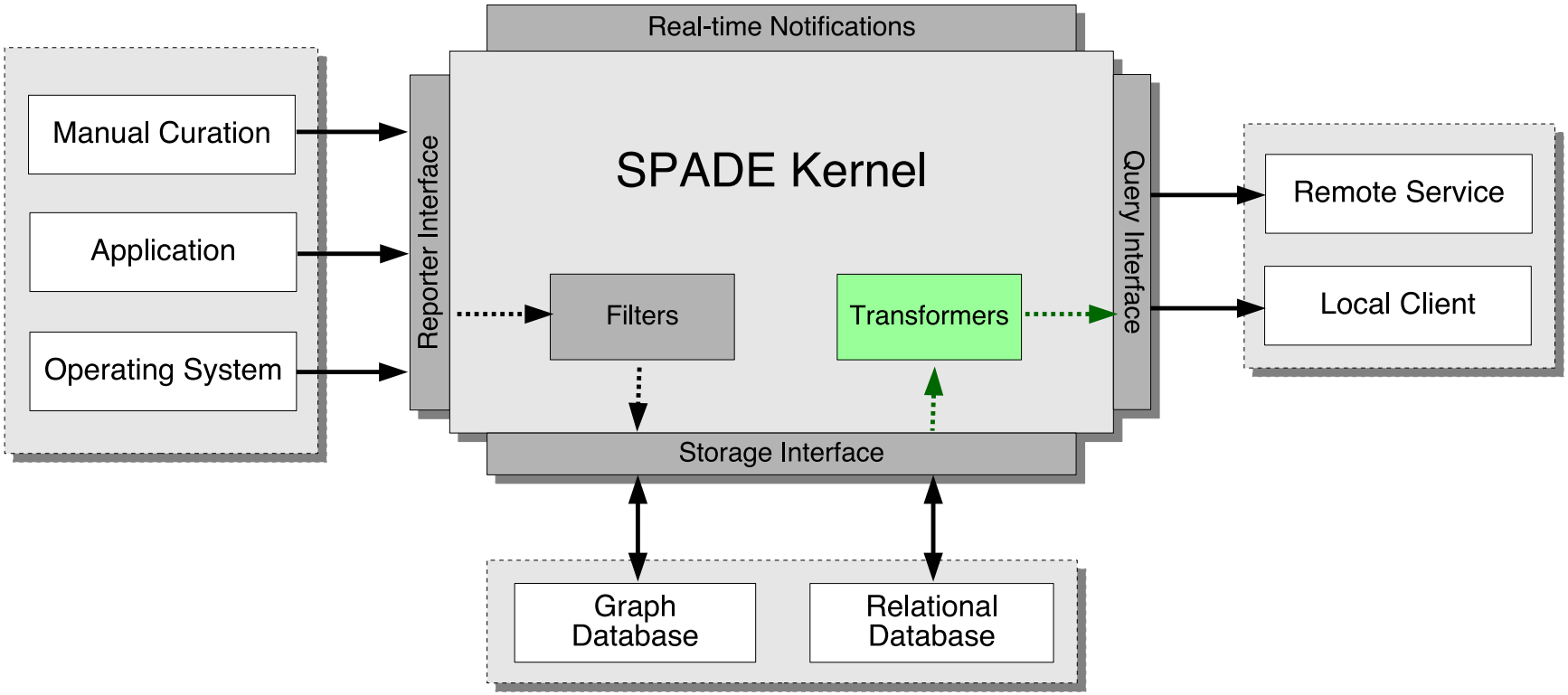
- "Common Criteria" unifies security standards
- Linux Audit designed to satisfy requirements
- In Oracle, RedHat, SUSE, ... guidelines
- Forensic analysts use logs after attack
- Event reconstruction often manual
- Provenance queries:
  - History of suspicious activity
  - Impact of malicious act
  - Check for sensitive flows
- 1 day of logs $\rightarrow$ 210M vertices, 833M edges

Challenge: Collection Volume

- Previously used as *data microscope*
  - *Focus* on specific attributes, timeframes
    - Files for software release
    - I/O hotspot identification
    - Sensitive Android flows
- Expensive to re-collect “big provenance”
- Suggests fine-grained instrumentation
- *Providing too much detail overloads users*

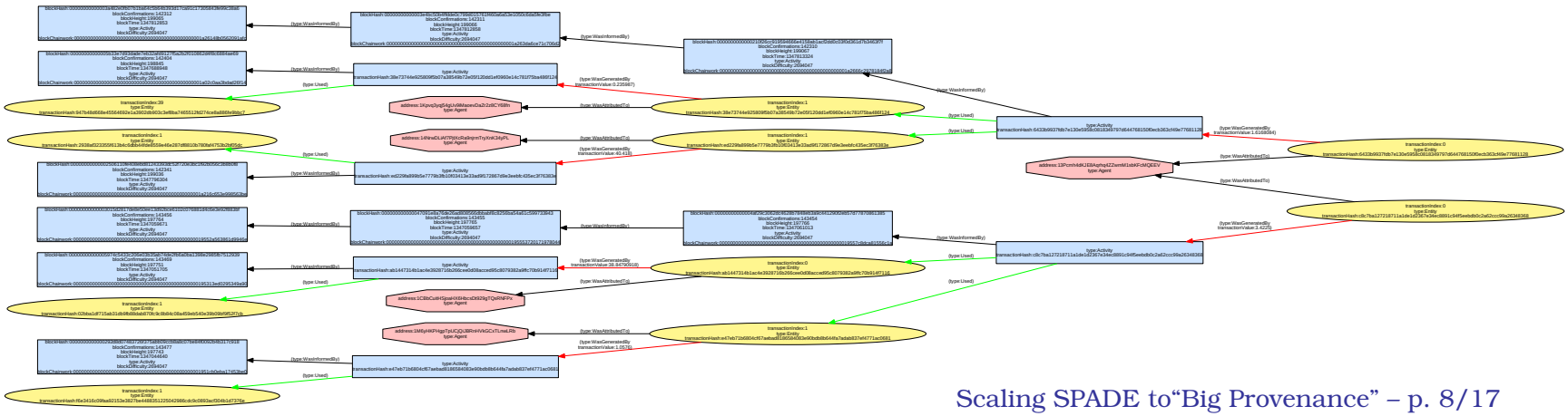
Approach: Transformers

- Support query *response rewriting*
- Can be composed



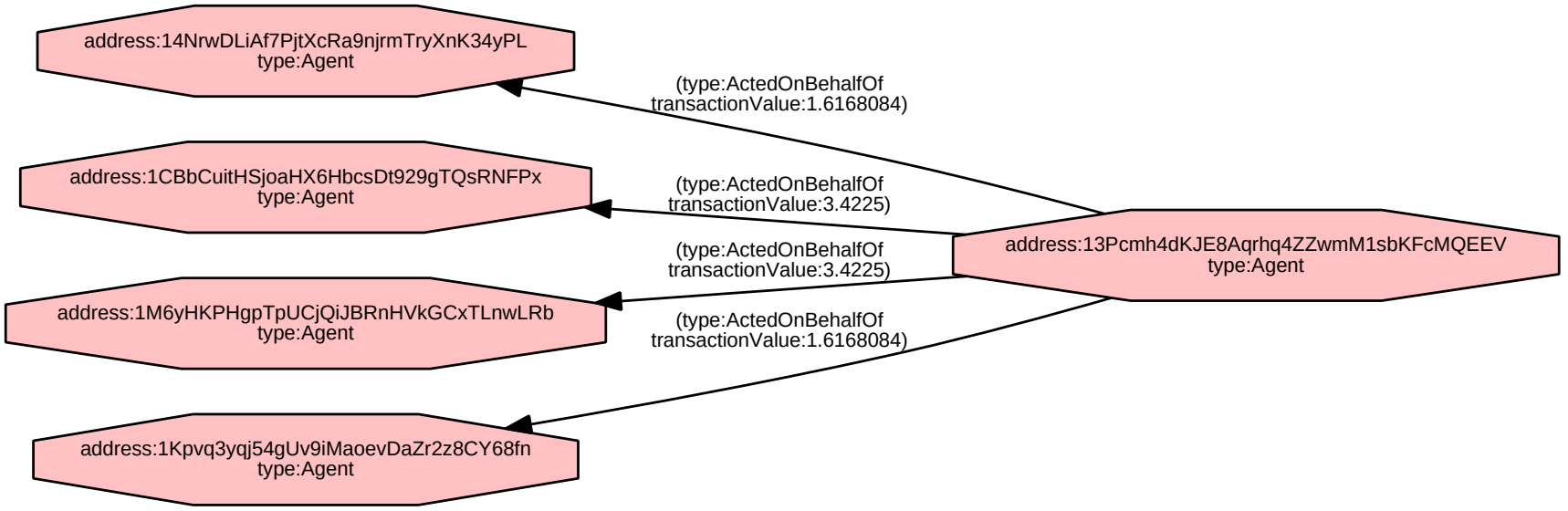
Untransformed Provenance

- Agents (in red): Bitcoin addresses
- Entities (in yellow): Payments
- Activities (in blue):
  - Transactions of incoming, outgoing payments
  - Blocks of transactions *mined* (verified) together
- Provenance of “bad” Bitcoin address



Transformed Response

- Transformer operates on original response
- Leverages provenance semantics
- Outputs Agent “network”



Agent Abstraction

- Results are (more) comprehensible
- Operates on (typically small) responses

Lineage levels	Original vertices	Original edges	Abstract vertices	Abstract edges
2	11	10	1	0
4	31	30	5	4
8	110	109	16	14
16	626	691	73	79

Composing Transformers

- Provenance of file read by web server
- Focus on aspects of interest
- System administrator can adjust results

Transformer	Vertices	Edges
None	1969	2831
+ Temporal traversal	1061	1114
+ No versions	9	59
+ Merge I/O	9	8

Challenge: Ingestion Rate

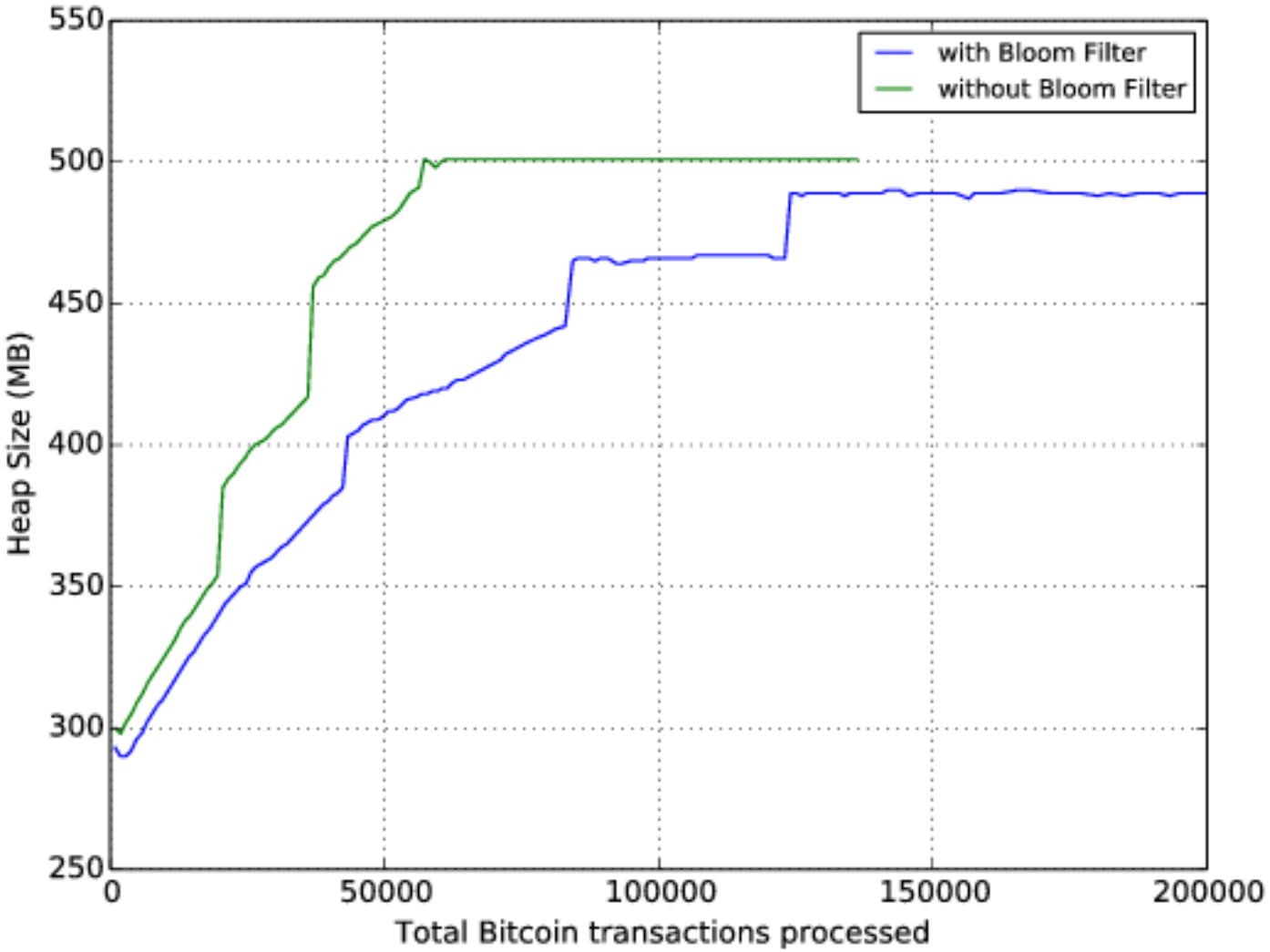
- Reporters send vertices, edges
- Edges can repeat endpoint vertices
- Put operations are idempotent
  - Minimizes state at source
- System must reconcile duplicates
- Baseline approach queries storage
  - Degrades ingestion performance
- Optimization used memory-bound cache
  - Memory pressure stops ingestion



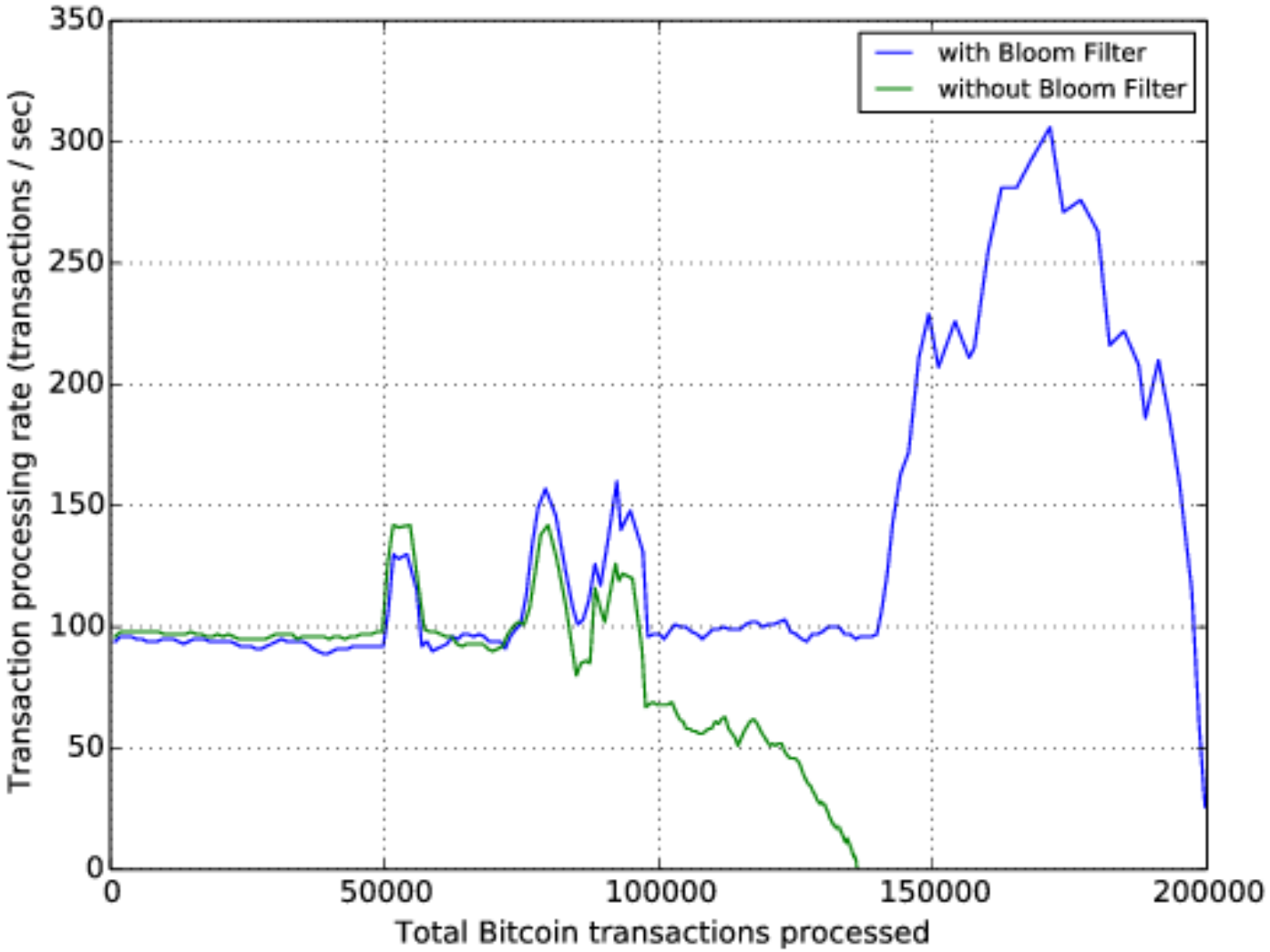
Approach: Hybrid Screening

- Aim to minimize queries to storage
- Bloom filter as primary screen
- Fixed size cache as secondary screen
- When vertex arrives, check in Bloom filter:
  - If *yes*, check if in cache:
    - If *yes*, stop ← Big effect
    - If *no*, check in storage:
      - If *yes*, stop ← Big effect
      - If *no*, put in storage
  - If *no*, put in storage
- For correctness argument, see paper

Memory Pressure



Ingestion Speed



Challenge: Integration

- SPADE integration *filters* used for
  - Aggregation in time
  - Fusion of related events
  - Connecting abstraction levels
- Operate on vertex, edge streams
- Larger integration window $\rightarrow$ More memory
- Smaller window $\rightarrow$ Lost integration chances
- Approach
  - *Content-based integration*
  - See paper



Conclusion

- Scaling is a work in progress
- Acknowledgement
  - TaPP '16 organizers, reviewers
  - NSF Grants IIS-1116414, ACI-1547467
  - DHS Science & Technology Directorate
- URL: https://github.com/ashish-gehani/SPADE/
- Email: ashish.gehani@sri.com
- Questions?

