Data Sharing:
Querying and Linking Distributed and Autonomous Data
Yang Cao, Wenfei Fan, Peter Buneman
University of Edinburgh

Background

Distributed databases: healthcare, business intelligence, e-government, ...  
- Tasks: querying, linking and sharing
- Data: distributed, heterogeneous, large scale

Challenges:
- Privacy & security: data owners (“private”) vs. users (“open”)
- Heterogeneity: relations, key-value, graphs
- Scalability: limited resources vs. big data analytics
- Accountability: ownership and accountability of shared data collection

Querying Shared Data with Security Heterogeneity [4]

Challenges
- heterogeneous security requirements
- centralized evaluation not possible

Solution:
1. Modeling data sharing protocols
2. Query evaluation under protocols
   - heterogeneous distributed query plans
   - optimal security-efficiency trade-offs
   - data movement and security allocation
   - leverage security heterogeneity in plans

Data sharing protocols $\rho$ specifying:
- capsules: logic units for computations over shared data
- hosts: data owners that host capsules
- pair-wise privacy requirements:
  - permitted capsule specifications
  - secure communication overheads

Heterogeneous distributed query plan: A DAG of atomic operations: $\delta = (op, t_c, X_1, \ldots, X_n, j)$
- $op$: an RA operator
- $t_c$: type of the security facility for operation $\delta$
- $X_i$: data from site $i$ for $\delta$
- $j$: the host site of $\delta$

Cost model: $\text{toll}(\delta)$:
- upfront cost, secure communication cost, computation cost
- submodular set function: $X_i$ to capsule at $j$

Planning under protocols: find optimal (minimum toll) plan

Complexity:
- decidable in NEXPTIME;
- PSPACE-hard even when $\rho$ is linear;
- $\Sigma^P_2$-hard even when $Q$ is in $\mathcal{SPC}$ and $\rho$ is linear.

Moreover, 2 and 3 hold even when $\mathcal{D}$ has two sites only.

Intractable to make the best (optimal) use of heterogeneity in data sharing.

Algorithm: Nonetheless, a two phase approach with guarantees:
Step (1): generating toll-minimized distributed plan $\xi_Q$
- toll-minimized $\xi_{op}$ for each $op$ of $Q$
- an $O((\log n))$-approximation algorithm for $\Xi$

Step (2): optimizing $\xi_Q$ within the toll budget
- via an automic operator $\kappa$ for “rebalancing” $\xi_Q$
- a near-optimal design of $\kappa$ (2-approximation of the optimal for $\Xi$)

Effectiveness: it speeds up state-of-the-art secure database system (SMCQL) by 18+ times over 1GB of TPCH data.

Heterogeneous Entity Resolution [5]

Main task: link entities across a relational database $D$ and a graph $G$

Challenge: traditional ER methods work for relations only (homogeneous).

HER: decide whether tuple $t \in D$ matches vertex $v$ of $G$:
- convert relations $D$ into a graph $G_D$
- whether $t$ matches $v$ $\Rightarrow$ whether $v_D$ in $G_D$ matches $v$ in $G$
- parametric simulation between $G_D$ and $G$

- graph simulation extended with label matching functions
- remains in quadratic-time
- learned threshold for score functions

parallelizable to scale over large graphs and relations

Scaling via Consistent Data Caching [1]

Background:
- large communication overhead and limited resources per node
- transactional accesses (workloads) become prevalent

Main task: scale out via data cache with consistency guarantees

Results:
- prove traditional policies (e.g., LRU) are not competitive for transactions
- formulate consistent cache scheme; show it’s NP-complete even for uni-size accesses, in contrast to trivially PTIME for conventional caching
- develop a consistent cache policy that is theoretically competitive and optimal when transactions access data items of unit size
- transaction batching and reordering for caching
- implemented and tested with Memcached@HBase: 126% improvement on average for transaction throughput, while guaranteeing consistency

Accountability of Shared Curated Data [3, 2]

Background:
- datasets are often shared/copied $\rightarrow$ modified $\rightarrow$ published $\rightarrow$ ...
- a data collection contains contributions from multiple users
- contributions form a dependency hierarchy (copy-modify-contribute)
- however, entire dataset is often treated as one single “article”

Main task: how to account the ownership of pieces of data in a dataset

Results:
- a model of citation graph for databases
- method for generating data summaries of “optimal” granularity
- stress measures of data summaries
- compute accountability by measuring stress
- application to a collectively curated pharmacology database (GtoPdb)

Publications