

A panoramic view of the Chicago skyline at dusk. The sky is a deep blue, and the city lights are beginning to glow. Several prominent skyscrapers are visible, including the Willis Tower and the Trump Tower. The text "SIGMOD 2017 Teaser Talks" is overlaid in white, and "Thursday - May 18th 2017" is overlaid in white below it.

SIGMOD 2017 Teaser Talks

Thursday - May 18th 2017



Fast Searchable Encryption with Tunable Locality

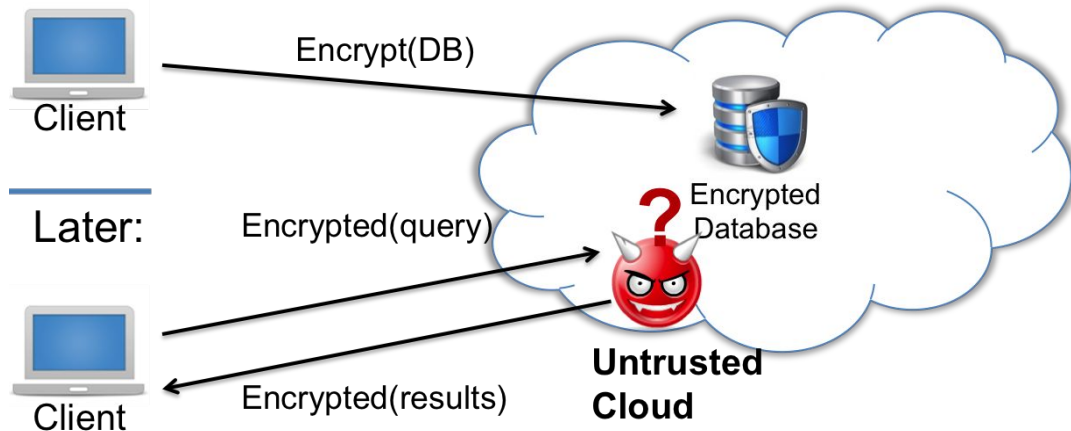
Encryption

Ioannis Demertzis, Charalampos Papamanthou (University of Maryland)

Problem: Privacy Preserving Querying via Searchable Encryption.

Our Searchable Encryption scheme has:

1. Formal proofs based on CRYPTO security definitions
2. Improved Efficiency
 - a. Up to **580x** for external disk
 - b. Up to **12x** in-memory
3. Different trade-offs tuning
 - a. Space
 - b. False Positives
 - c. Locality
 - d. Parallelism
 - e. Communication overhead



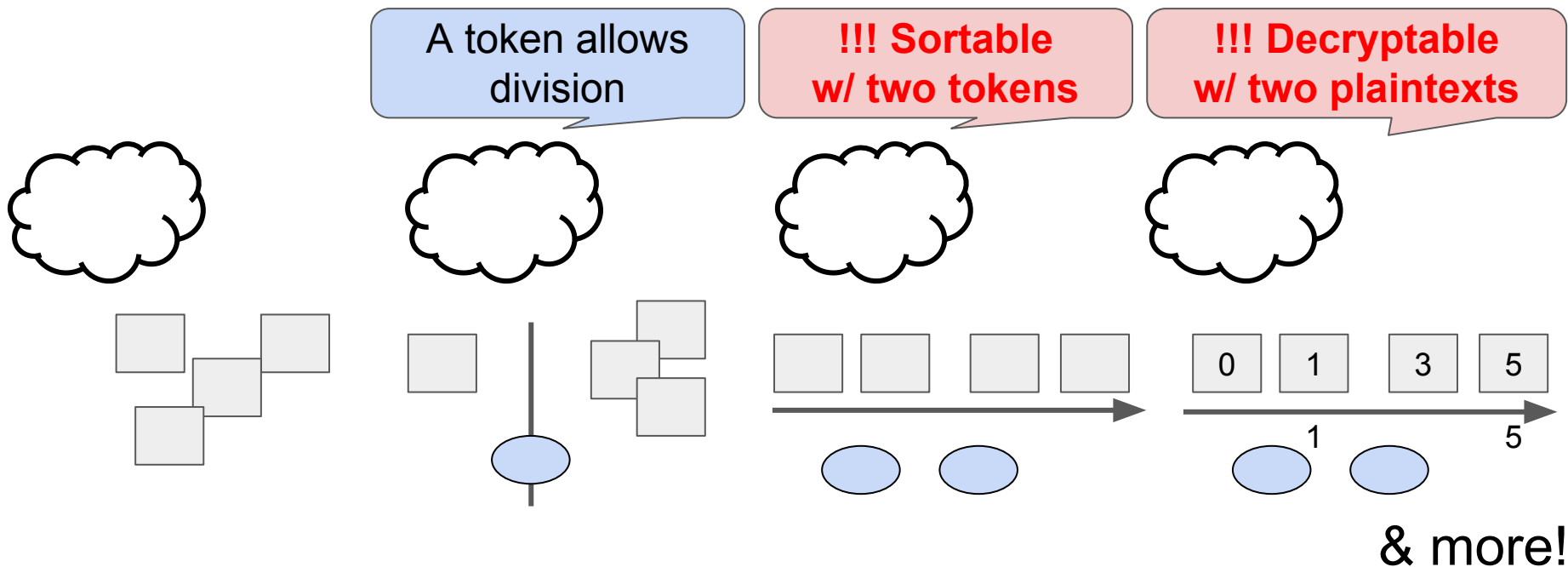


Cryptanalysis of Comparable Encryption in SIGMOD'16

Encryption

Caleb Horst (UW Tacoma) & Ryo Kikuchi, Keita Xagawa (NTT Secure Platform Laboratories)

Problem: Can a cloud break comparable encryption in [Karras et al. SIGMOD'16]?



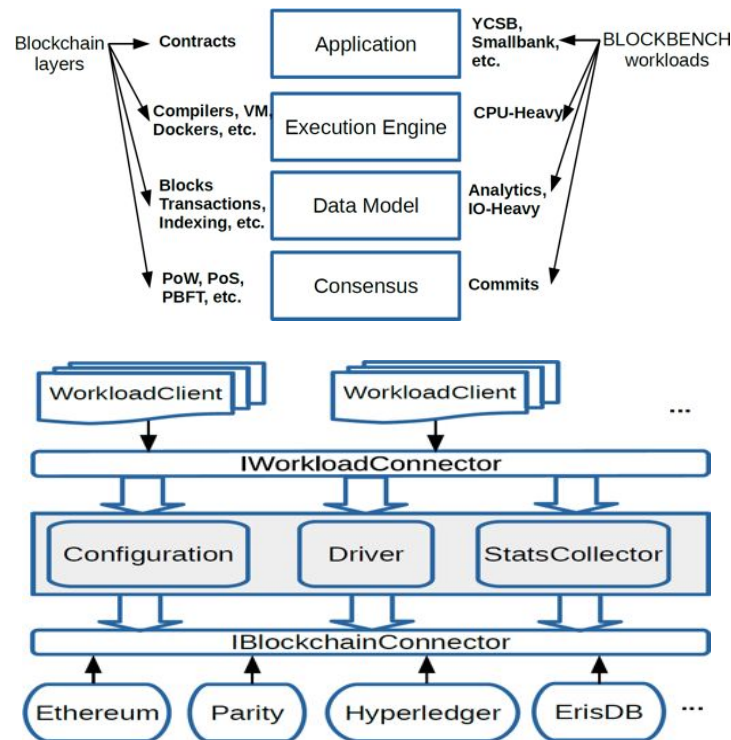


BLOCKBENCH: A Framework for Analyzing Private Blockchains

Encryption

Tien Tuan Anh Dinh, Ji Wang (NUS) ; Gang Chen (Zhejiang U.); Rui Liu, Beng Chin Ooi, Kian-Lee Tan (NUS)

- **Problem:**
 - Understanding and comparing existing blockchain systems, for data processing workloads
- **Challenges:**
 - Vast design space, many platforms, lack of data processing workloads
- **BLOCKBENCH:**
 - 4 layers of abstraction, extensible framework, with macro- and micro-benchmark workloads
 - Used to analyze Ethereum, Hyperledger Fabric and Parity





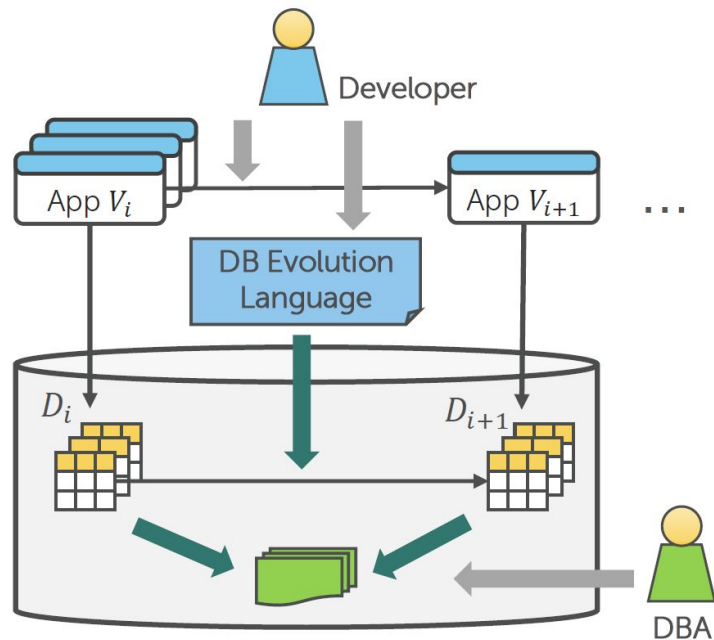
Co-Existing Schema Versions with a Bidirectional Database Evolution Language

Kai Herrmann, Hannes Voigt, Andreas Behrend, Jonas Rausch, Wolfgang Lehner (TU Dresden)

**Co-Existing
Schema
Versions**

**Independent
Physical
Migration**

Formal Evaluation of Correctness





Synthesizing Mapping Relationship Using Table Corpus

Cleaning

Yue Wang (U. Massachusetts Amherst); Yeye He (Microsoft Research)

Input: 100M+ web tables



Output: Synthesized Mappings

Company	Ticker
Microsoft	MSFT
Microsoft Corp.	MSFT
Microsoft Inc.	MSFT
Intel	INTC
General Electric	GE
...	...

Country	ISO
United States	USA
United States of America	USA
Korea (Republic)	KOR
Korea (South)	KOR
Republic of Korea	KOR
...	...

Why Synthesis?

- Better coverage, e.g., synonyms.
- Easy to curate

Application 1: Auto-Join

Company	# Employer	Ticker	Market Cap
Microsoft Corp.	...	MSFT	...
Walmart	...	WMT	...
Oracle	...	GE	...
General Electric	...	ORCL	...
AT&T Inc.	...	UPS	...
...

Application 2: Auto-Correction

ID	Employee	Company
...	...	MSFT
...	...	INTC
...	...	INTC
...	...	Microsoft → MSFT
...	...	Intel → INTC
...



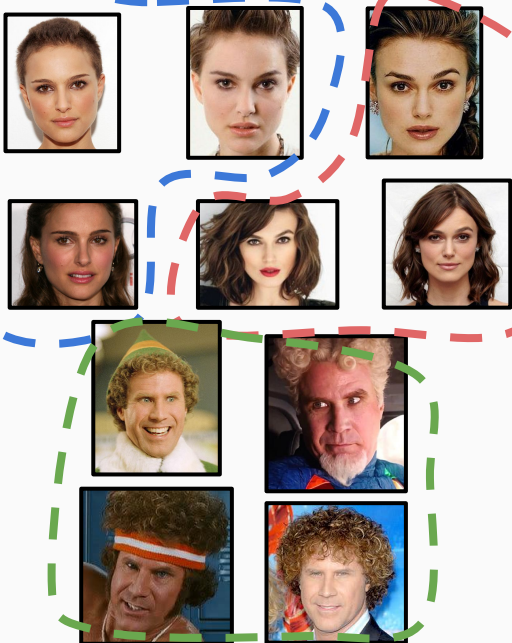
Waldo: An Adaptive Human Interface for Crowd Entity Resolution

Cleaning

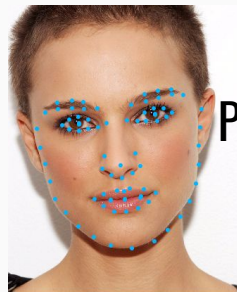
Vasilis Verroios, Hector Garcia-Molina (Stanford)

& Yannis Papakonstantinou (UC San Diego)

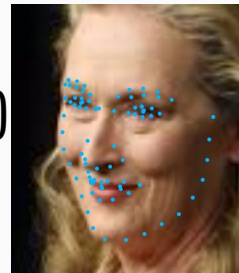
Entity Resolution



Computer Algorithms



$\text{Pr}(\text{Same Entity})$
20%



Human Tasks

Pairwise

Same Entity?

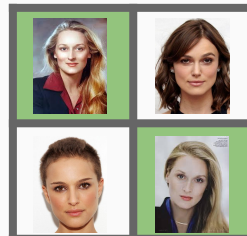


☐ YES

☐ NO



Multi-Item





ZipG: A Memory-efficient Graph Store for Interactive Queries

Tree & Graph

Anurag Khandelwal*, Zongheng Yang*, Evan Ye*, Rachit Agarwal†, Ion Stoica* (*UC Berkeley, †Cornell University)

Interactive graph serving

- Social networks
 - FB, Twitter, LinkedIn
- Graphs are **huge**
 - E.g., FB: ~billion nodes, ~trillion edges, rich attributes → 1.5 PB of data
- Graph queries: **complex**
 - Exhibit **little or no locality**
 - E.g., “Friends of my friends in Chicago”
- **Interactivity** requirements
 - Low latency, high throughput

ZipG, a **memory-efficient** graph store

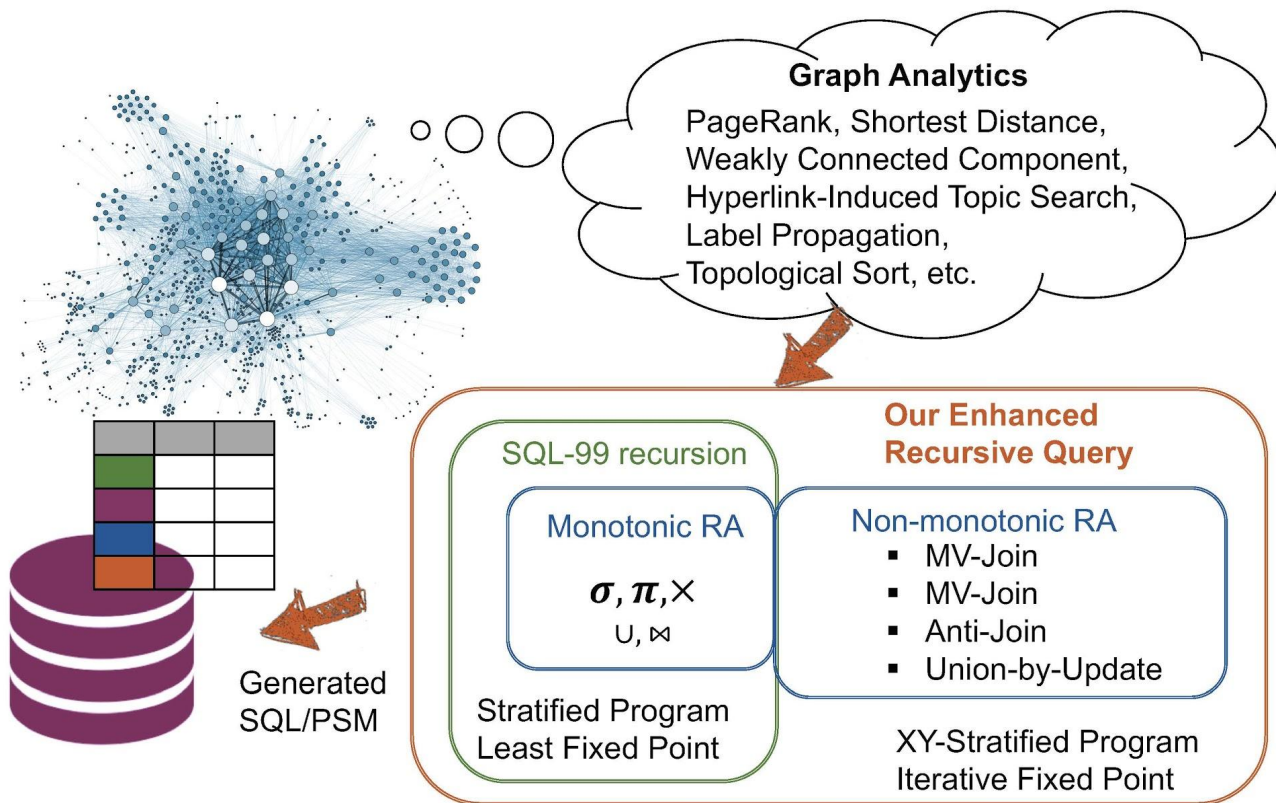
- Executes queries **directly on compressed graph** representation
 - No decompressions or scans
- Rich **functionality**
 - Queries from several industrial workloads; Regular path queries & graph traversals
- New **log-structured** graph storage
 - Efficiency for both read & write queries



All-in-One: Graph Processing in RDBMSs Revisited

Tree & Graph

Kangfei Zhao & Jeffrey Xu Yu (CUHK)





Computing A Near-Maximum Independent Set in Linear Time by Reducing-Peeling

Lijun Chang (UNSW Sydney), Wei Li, Wenjie Zhang

- Objective: compute **large** independent set for large graphs in a **time-efficient** (Subquadratic or more desirable linear to m) and **space-effective** ($2m + O(n)$ space) manner
 - m is the number of undirected edges

Algorithm	Time Complexity	Space Complexity	Exact Reduction Rules Used
BDOne	$O(m)$	$2m + O(n)$	Degree-one reduction [21]
BDTwo	$O(n \times m)$	$6m + O(n)$	Degree-one reduction [21] & Degree-two vertex reductions [21]
LinearTime	$O(m)$	$2m + O(n)$	Degree-one reduction [21] & Degree-two path reduction (this paper)
NearLinear	$O(m \times \Delta)$	$4m + O(n)$	Dominance reduction [21] & Degree-two path reduction (this paper)

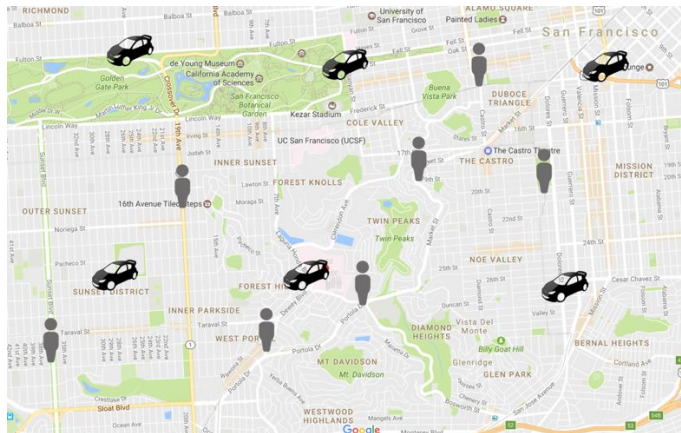
Table 1: Overview of our approaches (n : number of vertices, m : number of edges, Δ : maximum vertex degree)

Graphs	Independence Number	Gap to the Independence Number						NearLinear	Accuracy of NearLinear	Kernel Graph Size by NearLinear
		Greedy	DU	SemiE	BDOne	BDTwo	LinearTime			
GrQc	2,459	5	1	1	0	0	0	0*	100%	0
CondMat	9,612	17	5	1	4	2	1	0*	100%	0
AstroPh	6,760	24	10	1	2	0	1	0*	100%	0
Email	246,898	76	0	1	0	0*	0	0*	100%	0
Epinions	53,599	170	3	14	0	0	0	0	100%	6
dblp	434,289	484	63	53	45	5	4	0*	100%	0
wiki-Talk	2,338,222	536	0	14	0	0	0	0*	100%	0
BerkStan	408,482	11,092	3,000	4,458	1,088	385	766	428	99.895%	55,990
as-Skitter	1,170,580	34,591	2,336	5,886	319	55	170	39	99.997%	9,733
in-2004	896,724	14,832	3,553	5,918	656	351	412	57	99.993%	19,575
LiveJ	2,631,903	32,997	6,138	7,364	1,494	343	378	33	99.998%	10,173
hollywood	327,949	98	45	8	16	4	4	0*	100%	0

Table 3: The gap of the reported independent set size to the independence number computed by VCSolver [1] (* denotes that the independent set is reported as a maximum independent set by our algorithms)



Peng Cheng, Hao Xin, Lei Chen (HKUST)



Design the schedules for the vehicles to maximize the overall **satisfaction** of riders under the constraints:

- the deadlines of the riders
- the capacity of the vehicles



Riders' Satisfaction:

- Vehicle (Driver)-Related Utility
- Rider-Related Utility
- Trajectory-Related Utility



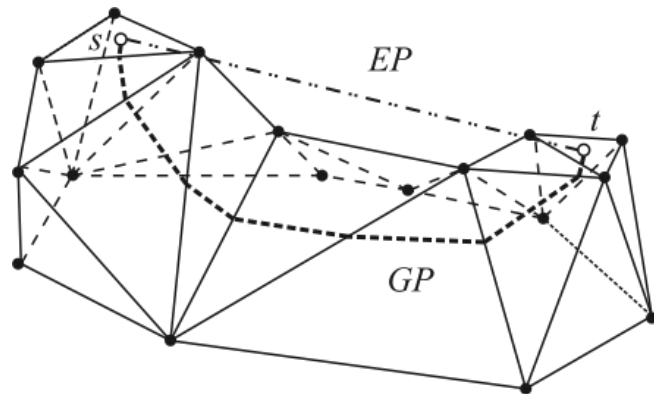
Victor Junqiu Wei (HKUST), Raymond Chi-Wing Wong (HKUST),
Cheng Long (Queen's University Belfast), David M. Mount (U of Maryland)

Problem

Given two POIs s and t on the terrain surface, estimate the geodesic distance between s and t .

Existing Method

- Computing Geodesic Distance On-The-Fly
 - Very Large Query Time
- Distance Oracle
 - ϵ -approximate (ϵ is a user-specified parameter)
 - Introduces a large amount of Steiner points/edges
 - Large Space and Building Time



Contributions

- We proposed a Distance Oracle, SE.
- Accuracy Guarantee: ϵ -approximate (ϵ is a user-specified parameter)
- Significantly outperforms State-of-the-Art

Building Time: 1-2 orders of magnitude smaller
Oracle Size: 1-3 orders of magnitude smaller
Query Time: 2-3 orders of magnitude smaller
With the same error guarantee ϵ

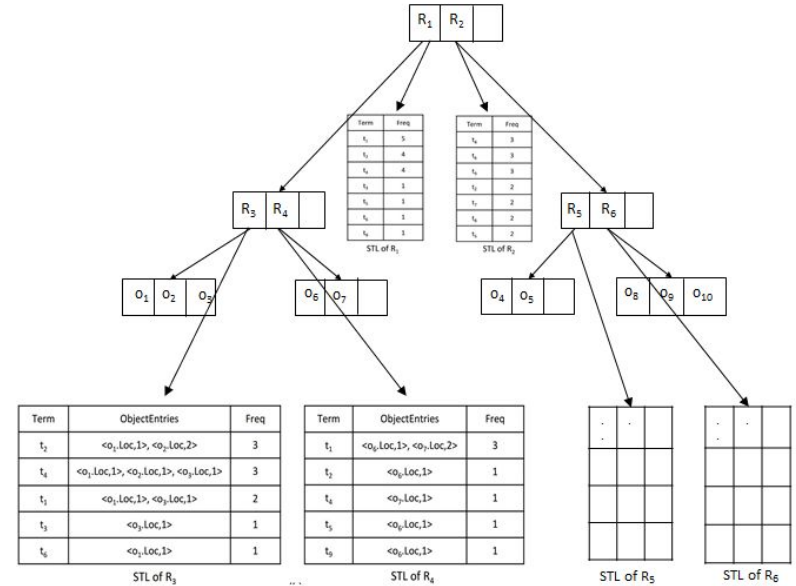


Efficient Computation of Top-k Frequent Terms over Spatio-Temporal Ranges

Space & Multidim

Pritom Ahmed, Mahbub Hasan, Abhijith Kashyap, Vagelis Hristidis and Vassilis J Tsotras (UC Riverside)

- **kFST Problem:** Given a spatio-temporal region R_Q find the most frequent terms among the social posts in R_Q .
- **Setting:** No predefined region borders, large disk resident data, exact answers
- **Obvious solution:** Use R-tree
- **Our solution:**
 - STL-enhanced indexing and top-k algorithms
 - Theoretical model to optimize STL space requirements
 - Space versus query trade-offs
 - various indexing options from no STLs to full and/or partial STLs





Optimizing Iceberg Queries with Complex Joins

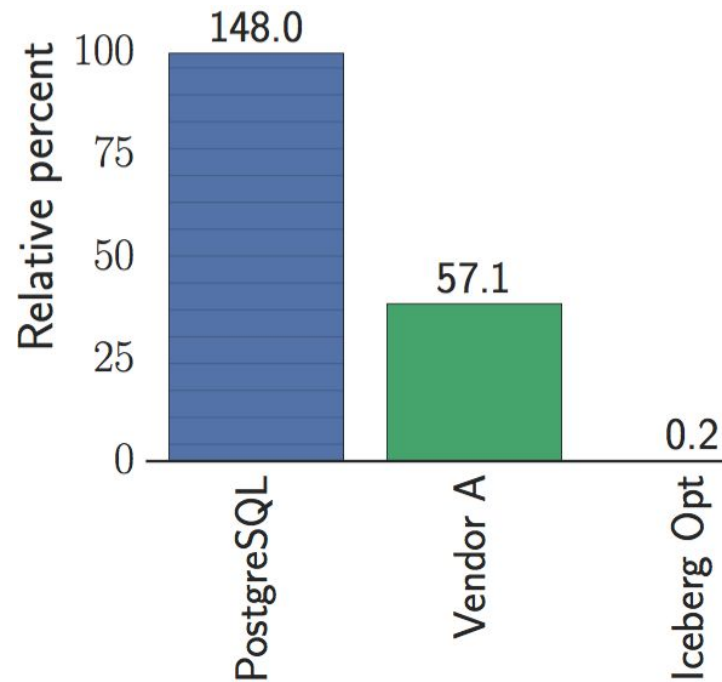
Opt & MainMem

Brett Walenz, Sudeepa Roy, Jun Yang (Duke U.)

iceberg query, *noun, SQL*.

aggregate query with arbitrary, complex joins and having clause

1. Formulate existing problems as iceberg queries
 - Market basket analysis
 - Skyline
2. Completely new framework for combining complementary techniques from existing problems
3. Formal conditions for applicability of techniques
4. Implementation in Postgres



The Dynamic Yannakakis Algorithm: Compact and Efficient Query Processing Under Updates

Muhammad Idris, Stijn Vansummeren and Martín Ugarte

Opt & MainMem

1

Dynamic Query Evaluation

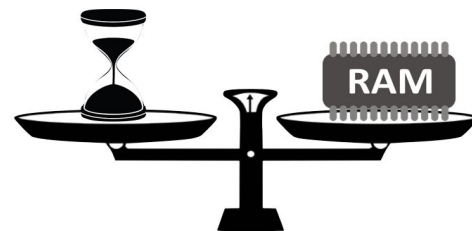


How to quickly
react under database
updates

2

Incremental View Maintenance

Keep (sub) results materialized
Only change what is *necessary*



3

Can we avoid the tradeoff?

Desiderata:

- In-memory data structure
- Constant-delay enumeration of results
- Space linear in the size of the database
- Efficiently adapt under updates

4

Dynamic Yannakakis

A practical algorithm

Desiderata



+

Match two theoretical lower bounds

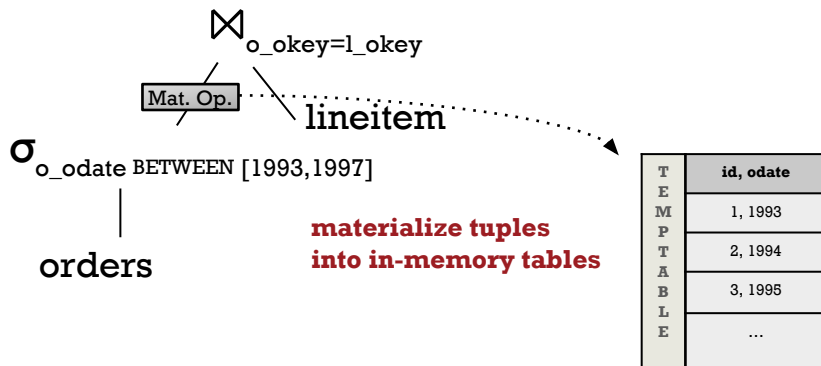


Revisiting Reuse in Main Memory Database Systems

Opt & MainMem

Kayhan Dursun, Carsten Binnig, Ugur Cetintemel, Tim Kraska (Brown)

HOW REUSE IS DONE TODAY ?



EXPENSIVE MATERIALIZATION COSTS

THESE MAY NOT PAY OFF IN THE FUTURE

IF YOU WOULD LIKE TO SEE HOW WE GET REUSE FOR FREE, PLEASE COME AND SEE MY TALK 😊

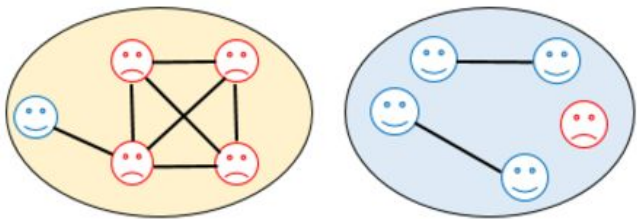
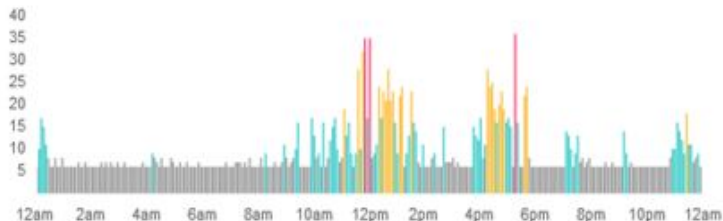
Teaser Talks (Second Part)



Pufferfish Privacy Mechanisms for Correlated Data

Shuang Song, Yizhen Wang, Kamalika Chaudhuri (UCSD)

Sensitive Data with Correlation



Challenge:

DP does not hide sensitive information about individual records in the presence of correlations.

Our Contribution

- a general privacy-preserving mechanism for any Pufferfish privacy framework – *the Wasserstein Mechanism*
- a mechanism when the correlation is described by a Bayesian network – *the Markov Quilt Mechanism*
- an efficient implementation when the correlation is described by a Markov chain
- experiments on real data-sets



Differentially Private Stochastic Gradient Descent for in-RDBMS Analytics

Privacy

Xi Wu and others (U. Wisconsin-Madison)

- **Better** differentially private Stochastic Gradient Descent (SGD).
 - SGD is a **popular optimization algorithm** for machine learning.
 - Differential privacy is the **de facto standard** for formalizing privacy.
- **Improve** private SGD on the following aspects simultaneously:
 - Easier to **implement**: “Bolt on” with an existing implementation.
 - Run **faster**,
 - Better **convergence/accuracy** and
 - Support a stronger **privacy model**.
- **Essence behind the “all-win” improvements**: A novel analysis of the L_2 -sensitivity of SGD.



Pythia: Data Dependent Differentially Private Algorithm Selection

Privacy

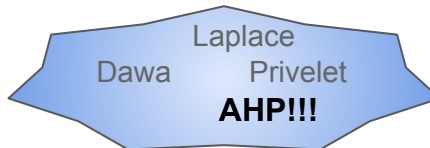
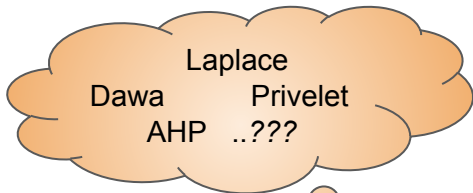
Ios Kotsogiannis, Ashwin Machanavajjhala, Gerome Miklau, Michael Hay

Algorithm Selection...

- Private evaluation of task T
- Algorithms A_T suitable for T
- Choose $A^* \in A_T$ to answer T

...Without Data Access

- No clear winner in A_T for all instances of T
- Running all algorithms **violates privacy**



Pythia

- End-to-end privacy
- Chooses the right algorithm



Utility Cost of Formal Privacy for Releasing National Employer-Employee Statistics

S Haney, A Machanavajjhala, J Abowd, M Graham, M Kutzbach, L Vilhuber



US Law:

Title 13
Section 9

\approx



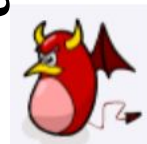
Pufferfish
Privacy
Requirements



??



DP-like
Privacy
Definition



Noisy
Employer
Statistics



Comparable or lower error than current non-private methods



Online Deduplication for Databases

Lianghong Xu (CMU); Andy Pavlo (CMU);

Sudipta Sengupta (Microsoft Research); Gregory Ganger (CMU)



mongoDB

WIREDTIGER



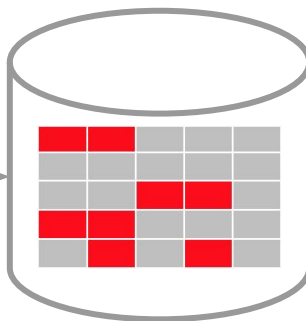
QFix: Diagnosing errors through query histories

Xiaolan Wang, Alexandra Meliou (U. Massachusetts Amherst) &
Eugene Wu (Columbia U.)

QFix: Fixing bad queries for dynamic DBMS

Find & fix errors in query histories.

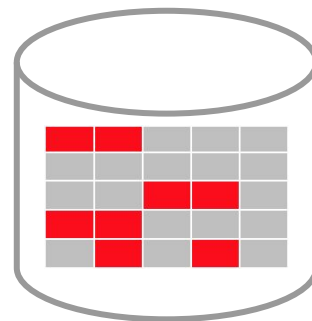
INSERT
DELETE
UPDATE



Queries Change Database

Traditional Data Cleaning

Find & fix errors
directly on current db



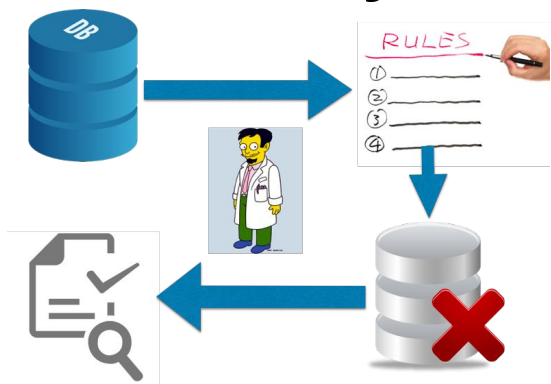
Static Database



UGuide – User-Guided Discovery of FD-Detectable Errors

S. Thirumuruganathan, L. Berti-Equille, M. Ouzzani, J. Quiane-Ruiz, N. Tang (HBKU)

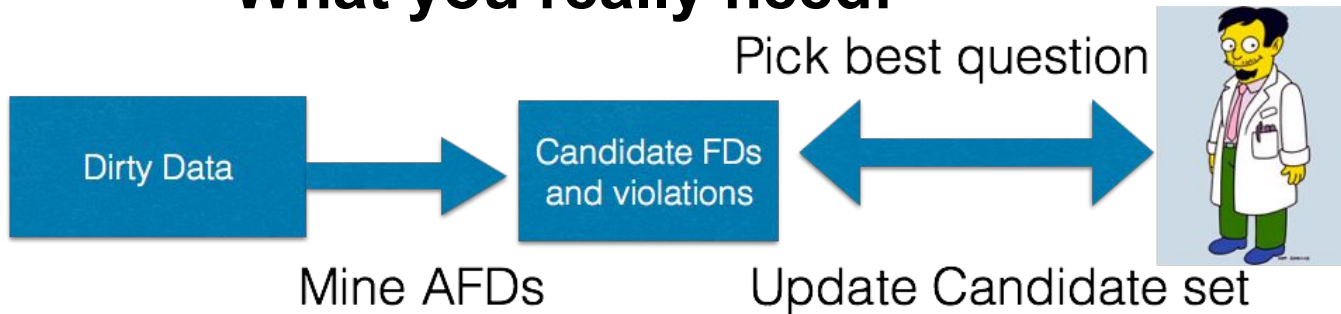
Ideally!



Reality!



What you really need!





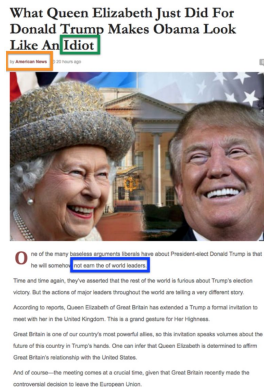
SLiMFast: Guaranteed Results for Data Fusion and Source Reliability

Cleaning

Theo Rekatsinas; Manas Joglekar; Hector Garcia-Molina;
Aditya Parameswaran; Christopher Ré

Problem: Clean inaccurate, conflicting data and find hoax sources!

SLiMFast: New ML data fusion framework; subsumes and generalizes most existing models; theoretical guarantees on the quality of its output.



Features of the Article

Author: American News

Sentiment:

- polarity = 0.374
- subjectivity = 0.640

Text Quality:

- # of misspelled words
- # of grammatical errors

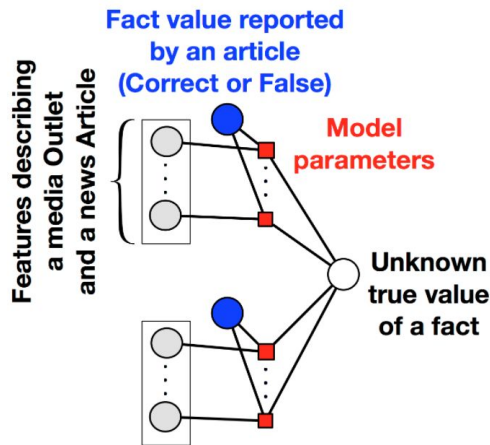
Web Traffic Statistics:

80.40% **22,352** **3,271**

1.34 **2:50**

Use features to describe sources and fix inaccurate data twice more accurately!

In most cases, Logistic Regression is enough to solve data fusion!





Crowdsourced Top-k Queries

Crowdsourcing

by Confidence-Aware Pairwise Judgments

Ngai Meng KOU¹, Yan LI¹, Hao WANG², Leong Hou U¹, Zhiguo GONG¹

¹University of Macau, ²Nanjing University

Problem: find the top-k items from a set of computationally challenging items.

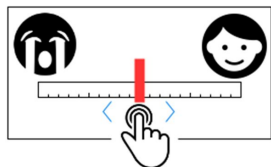
UI of microtask: pairwise comparison.

What's new?

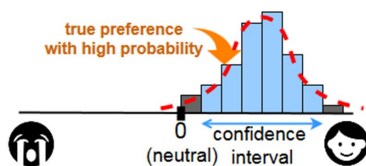
Previous work: the budget for every pair is constant and the query processing is not confidence-aware.

Ours: the **budget** for a pair is dynamically **decided** by the **hardness** with **confidence**.

Pairwise Preference Judgments



Preference Distribution



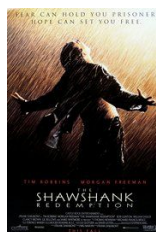
IMDb 14th

Forrest Gump



IMDb 1st

The Shawshank Redemption



2016 Biggest Disappointment Golden Schmoes

Batman v Superman: Dawn of Justice



IMDb 1st

The Shawshank Redemption



HARD

Needs more budget

EASY

Needs less budget

Then, how to design a method that optimizes cost and latency with quality guarantee?



Falcon: Scaling Up Hands-Off Crowdsourced Entity Matching to Build Cloud Services

Crowdsourcing

Sanjib Das^{*}, Paul Suganthan G. C.^{*}, AnHai Doan^{*}, Jeff Naughton^{*}, Ganesh Krishnan⁺,
Esteban Arcaute⁺, Rohit Deep⁺, Vijay Raghavendra⁺, Youngchoon Park⁺⁺

^{*}University of Wisconsin-Madison, ⁺WalmartLabs, ⁺⁺Johnson Controls

Table A

Name	City	State
Dave Smith	Madison	WI
Joe Wilson	San Jose	CA
Dan Smith	Middleton	WI

Table B

Name	City	State
David D. Smith	Madison	WI
Daniel W. Smith	Middleton	WI



Domain scientists
@ UW-Madison



Crowd workers



Challenge: Scale up EM workflow

- DAG involving rules, ML, crowdsourcing
- Use crowd time to mask machine time

Results

Matches tables of **1M - 2.5M tuples, \$54-66, 2-14 hours**

Deployed as a cloud service at CloudMatcher.io

Used extensively at several organizations

- e.g., UW Depts., Johnson Controls, WalmartLabs, etc.

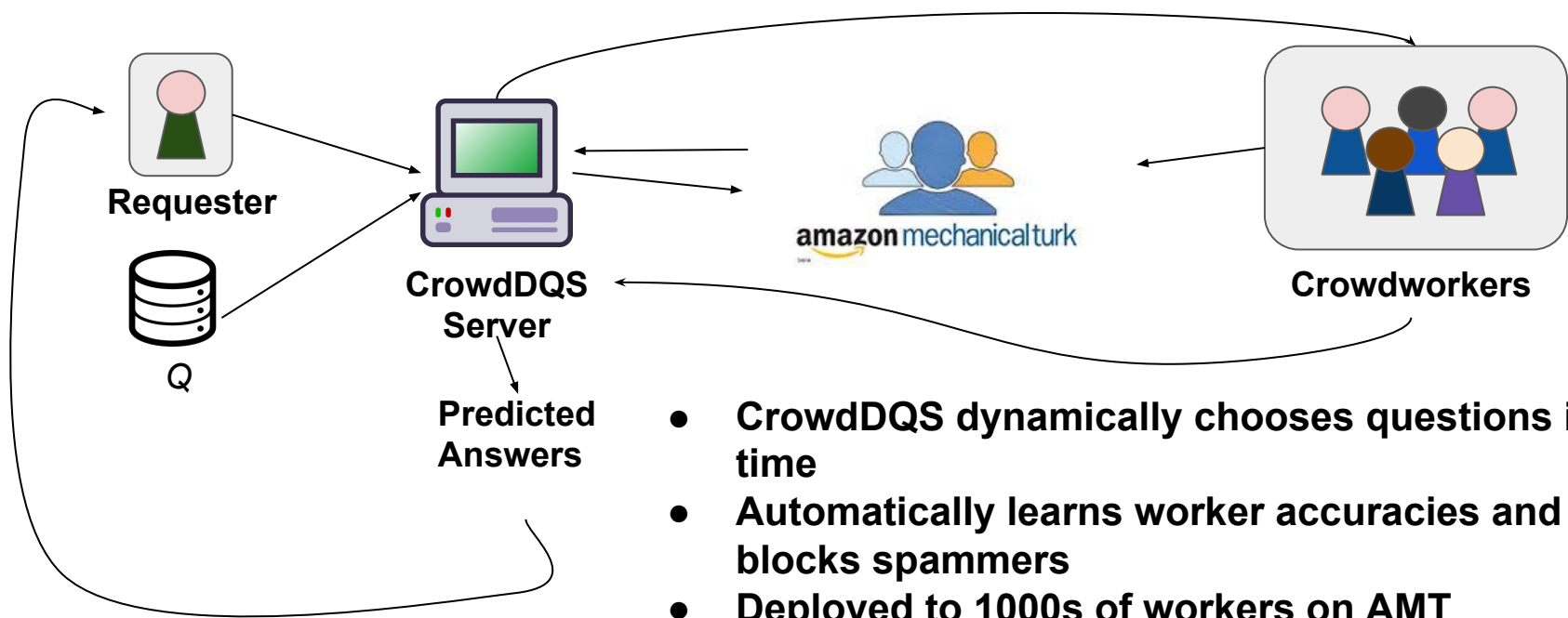
Talk@Session 28, Buckingham (Thur 14:00-15:40)



CrowdDQS: Dynamic Question Selection in Crowdsourcing Systems

Crowdsourcing

Asif R. Khan & Hector Garcia-Molina (Stanford)



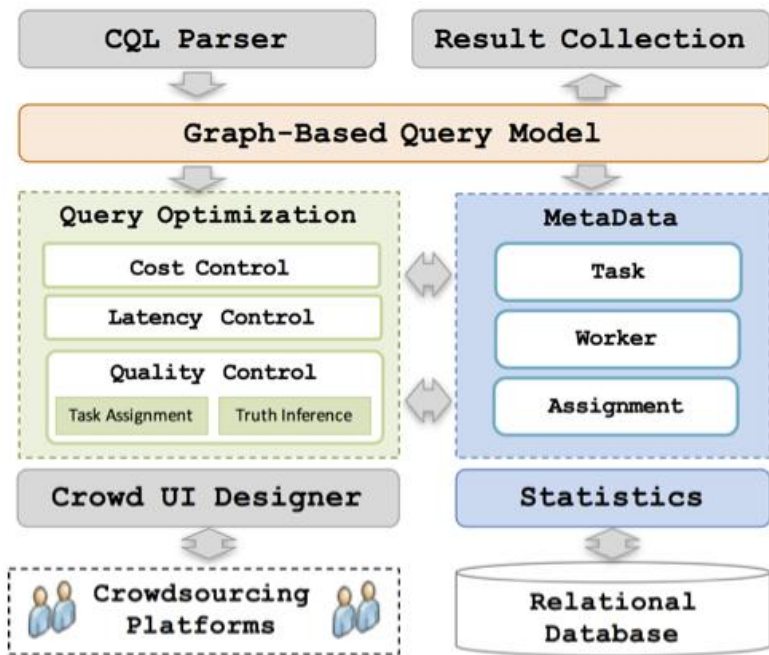
- CrowdDQS dynamically chooses questions in real time
- Automatically learns worker accuracies and blocks spammers
- Deployed to 1000s of workers on AMT
- Can reduce costs up to 6x



CDB: A Crowd-Powered Database System

Crowdsourcing

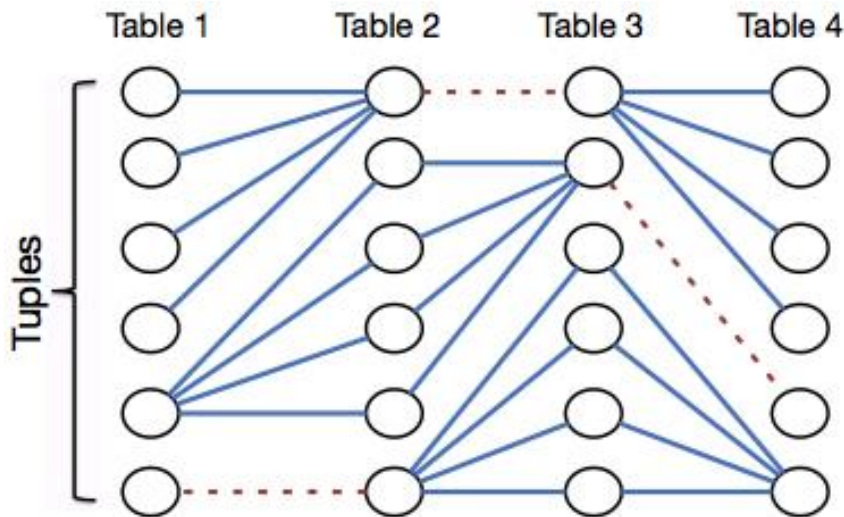
Guoliang Li and others (Tsinghua U.)



Graph-based Tuple-level Optimization Model

- Tree Model: 15 questions
- Graph Model: 3 questions

Multi-Goal Optimization(Cost, Quality, Latency)





Scaling Locally Linear Embedding

Space & Multidim

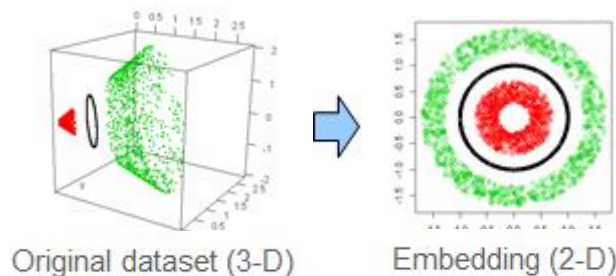
Yasuhiro Fujiwara and others (NTT Communication Science Laboratories)

LLE reduces the dimensionality of dataset

Step1 k-NN graph

Step2 Edge weight by regression

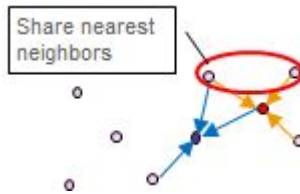
Step3 Eigen decomposition of $(I-W)^T(I-W)$



We reduce the computation cost as follows:

1. used common nearest neighbors

- Efficiently find k-NN
- Incrementally compute edge weight



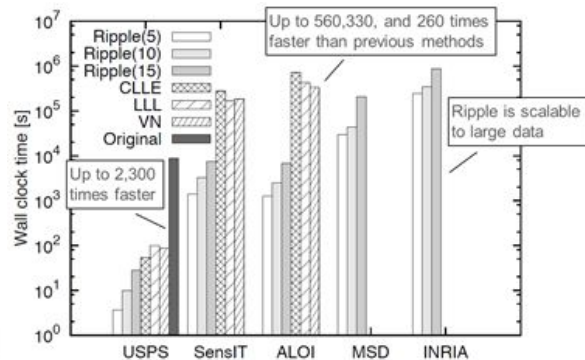
2. LU decomposition

- Efficiently compute Eigen decomposition
- Low memory consumption

$$(I - W)^T(I - W) = U^T L^T L U$$

$$\lambda_N = \{(a_\tau)^T a_\tau\} / \{(a_\tau)^T a_{\tau-1}\}$$

$$\text{where } a_{\tau-1} = U^T b, b = U^T b', b' = L b'', b'' = U a_\tau$$



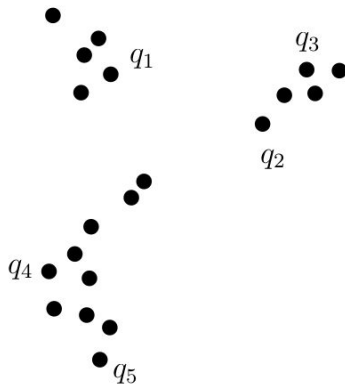


Dynamic Density Based Clustering

Space & Multidim

Junhao Gan and Yufei Tao (U of Queensland)

New Query: A **cluster-group-by query** is given a set Q of data points, and groups the points of Q by the clusters they belong to.



Contributions:

Data structures with fast update and query time.

+

Lower bounds when such structures do not exist.

For $Q = \{q_1, q_4, q_5\}$, answer: $\{q_1, q_4, q_5\}$.

For $Q = \{q_1, q_2, q_4\}$, answer: $\{q_1, q_4\}, \{q_2\}$.



Extracting Top-K Insights from Multi-Dimensional Data

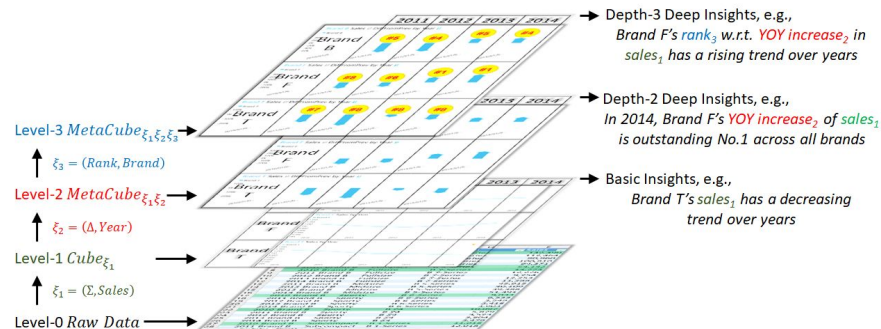
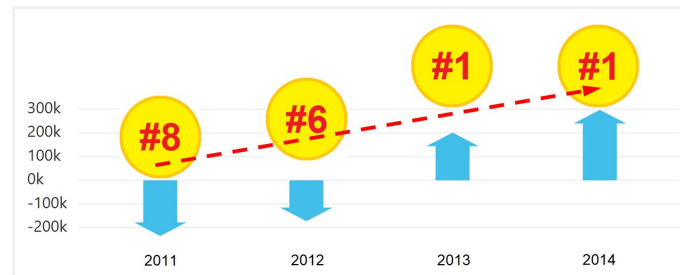
Bo Tang (PolyU); Shi Han (MSR); Man Lung Yiu (PolyU); Rui Ding (MSR); Dongmei Zhang (MSR)

Deep Insights has been a sub-branch project of the **Auto Insights** research framework at Microsoft Research

Auto Insights has been continuously shipping new techniques (e.g., Quick Insights, Scoped Insights, etc.) to **Microsoft Power BI** since Dec 2015, as enabling techniques for **leading** the BI & Analytics market

Mining Deep Insights against hierarchical meta cube

E.g., Brand F's rank_3 (across all brands) w.r.t. YOY increase_2 of sales_1 has a rising trend





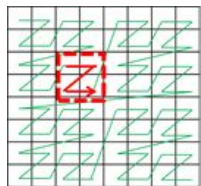
Framework Based on Query-Aware and Skew-Tolerant Space-Filling Curves

Shoji Nishimura (NEC) & Haruo Yokota (Tokyo Institute of Technology)

Problem:

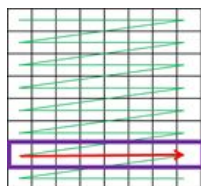
The optimal curve for the target query pattern

Square Query

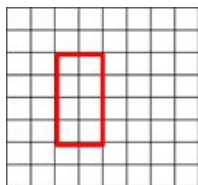


Z-Curve

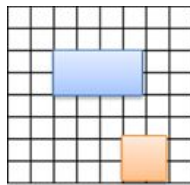
Elongate Rect. Query



C-Curve (Composite Index)



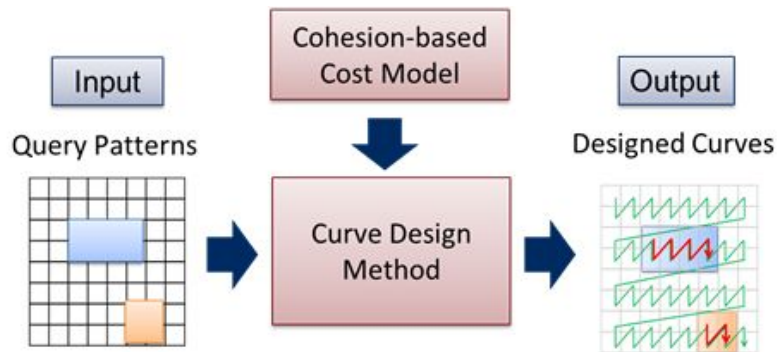
Intermediate Query?



Multiple Queries?

Contributions:

- **Cohesion-based Cost Model**
 - Measure **curve property** for **query pattern** and **data distribution**
- **Curve Design Method**
 - Heuristics to design **effective curves** in terms of **the cost model**





Leveraging Re-costing for Online Optimization of Parameterized Queries with Guarantees

Opt & MainMem

Anshuman Dutt, Vivek Narasayya and Surajit Chaudhuri (Microsoft Research)

Parameterized query

Select attributes
From relations
Where join predicates and
other predicates and
 **$i_current_price < @Param1$ and
 $cs_sales_price < @Param2$**

Query instance (q_i)

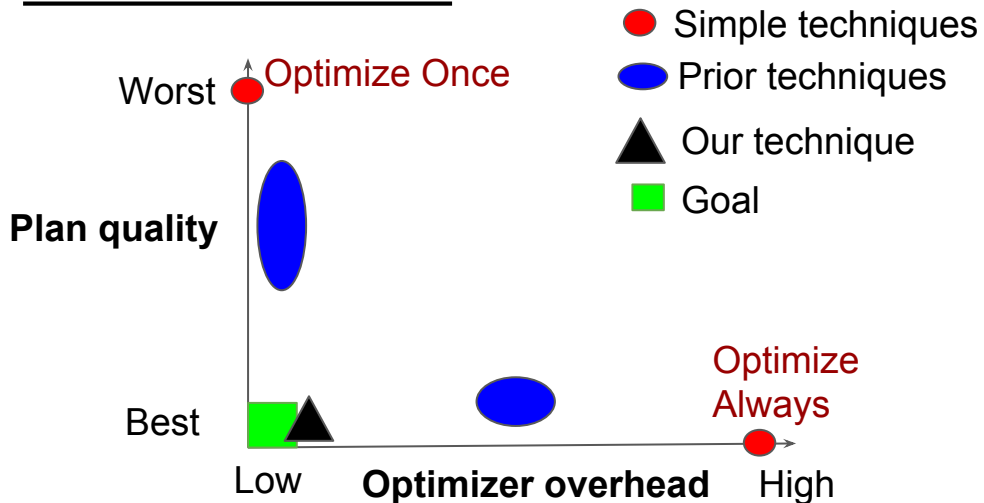
$[@Param1 = 10, @Param2 = 15]$

Many different query instances may lead
to same optimal execution plan

Opportunity: to avoid optimizer overhead

Problem: online version of parametric query optimization (PQO)

Performance Trade-off





Handling Environments in a Nested Relational Algebra

Opt & MainMem

with Combinators and an Implementation in a Verified Query Compiler

Joshua Auerbach and others (IBM Research)

Handling Environments:

- Keep Variables: simple plans, complex rewrites
- Remove Variables: simple rewrites, complex plans

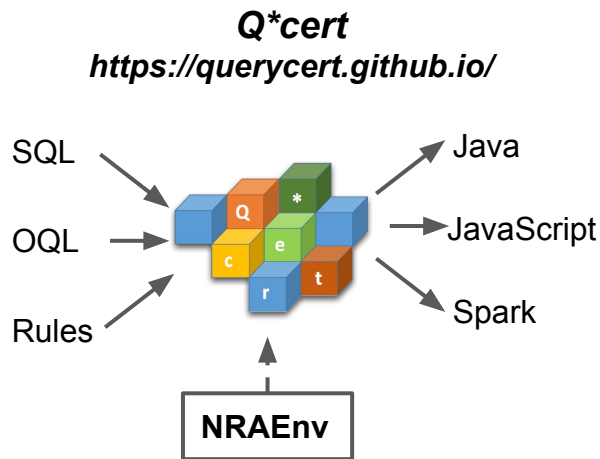
Nested Relational Algebra with Combinators:

- **NRAEnv** = NRA Combinators + Environment
- Definition, Expressivity, Rewrites, Applications

Implementation:

- Written with Coq Proof Assistant
- Algebraic Optimizer Verified Correct
- Q*cert demo at SIGMOD 2017

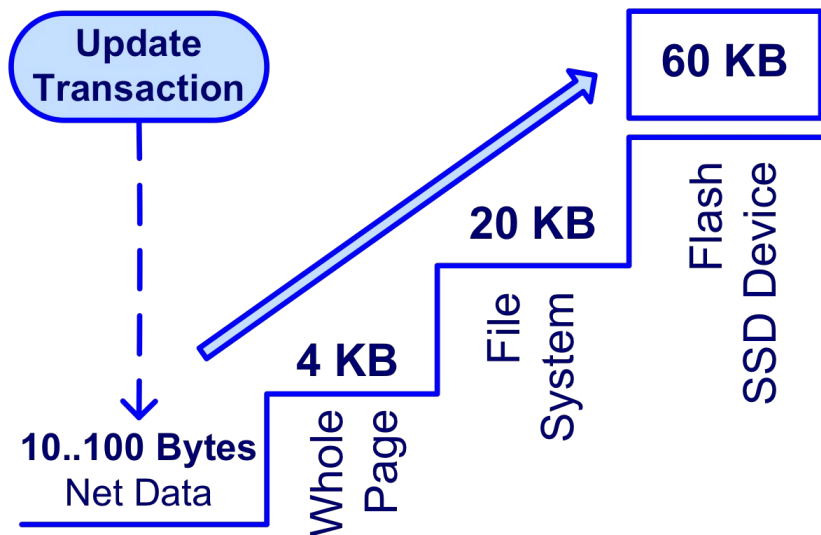
Verified Query Compiler:



From In-Place Updates to In-Place Appends: Revisiting Out-of-Place Updates on Flash

S. Hardock^{*}, I. Petrov⁺, R. Gottstein^{*} and A. Buchmann^{*}

(*TU Darmstadt, +Reutlingen University)



Problem:

- Small updates → write-amplification **600x**

Approach:

- Small updates → physical in-place appends

IPA: Flash updates without a prior erase