Human-in-the-loop Data Integration

Guoliang Li

Department of Computer Science, Tsinghua University, China

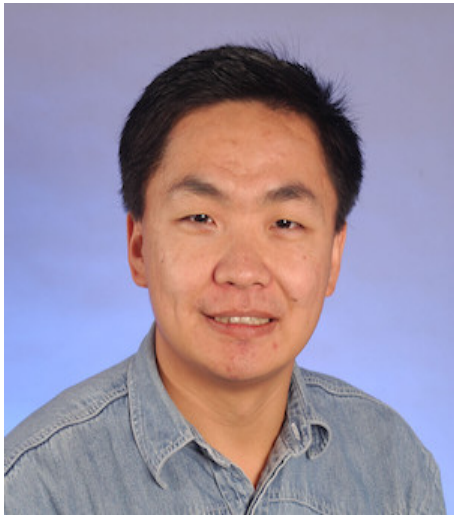
http://dbgroup.cs.tsinghua.edu.cn/ligl

Acknowledgement

Jianhua Feng
@Tsinghua

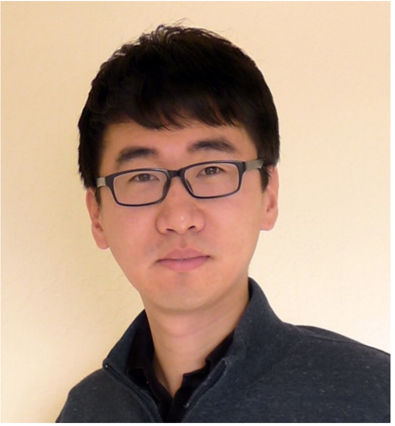
Lizhu Zhou
@Tsinghua

Beng Chin Ooi
@NUS



Chen Li
@UCI

Acknowledgement



Jiannan Wang
aP@SFU

Dong Deng
PostDoc@MIT
On Job Market!

Ju Fan
AP@RUC

Yudian Zheng
@HKU

Thank everyone who support me!

- ✓ Collaborators
- ✓ Students
- ✓ Friends
- ✓ ……

Data Integration (DI)

Combine data in different sources and provide users with a unified view

Brand	Product	Region	Price
Apple	iPhone6S	Beijing	4000
Apple	iPhone6SP	Beijing	5000
Samsung	Galaxy S7	Beijing	3500

Name	Loc	Sales
6S 4.7’	Bei Jing	40K
6S 5.5’	Bei Jing	30K
S7	Bei Jing	35K

Data Integration

Brand	Product	Loc	Price	Sales
Apple	iPhone6S	Beijing	4000	40K
Apple	iPhone6SP	Beijing	5000	30K
Samsung	Galaxy S7	Beijing	3500	35K

Data Analysis

Is there any correlation between location and revenue?

Data Science Pipeline: Data Integration → Data Analysis

Data Integration (DI)

- **Data Integration is important and challenging**
  - New York Times
    - 80% of a data science project is to clean and integrate the data, while 20% is actual data analysis
  - Mark Schreiber of Merck
    - data scientists spend 98% of on "grunt work" and
    - only one hour per week on "useful work"

- **In many communities**
  - **DB, AI, KDD, Web**

Entity Matching in DI

☐ Date Integration

– data acquisition, extraction, cleaning, schema matching, entity matching, etc.

☐ Entity Matching (EM): A core problem

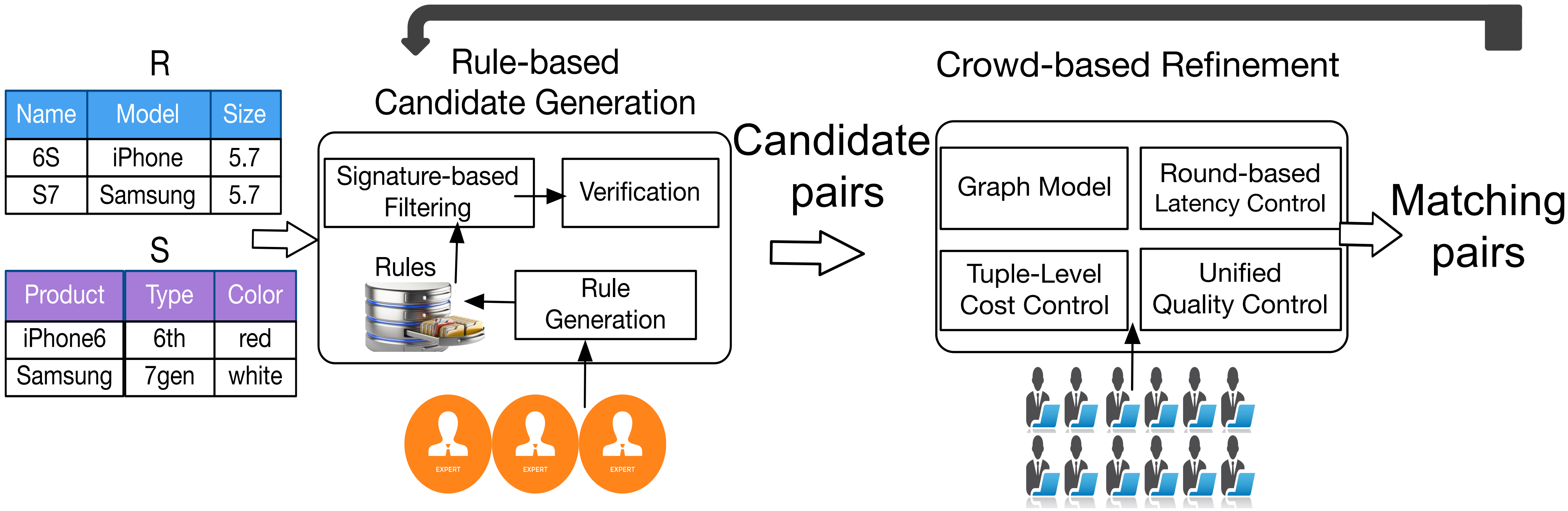
– Find pairs of records referring to the same entity

Brand	Product	Region	Price
Apple	iPhone6S	Beijing	4000
Apple	iPhone6SP	Beijing	5000
Samsung	Galaxy S7	Beijing	3500

Name	Loc	Sales
Apple 6S 4.7’	Bei Jing	40K
Apple 6S 5.5’	Bei Jing	30K
Samsung S7	Bei Jing	35K

Data are full of errors and inconsistencies.

Hybrid Human-Machine EM



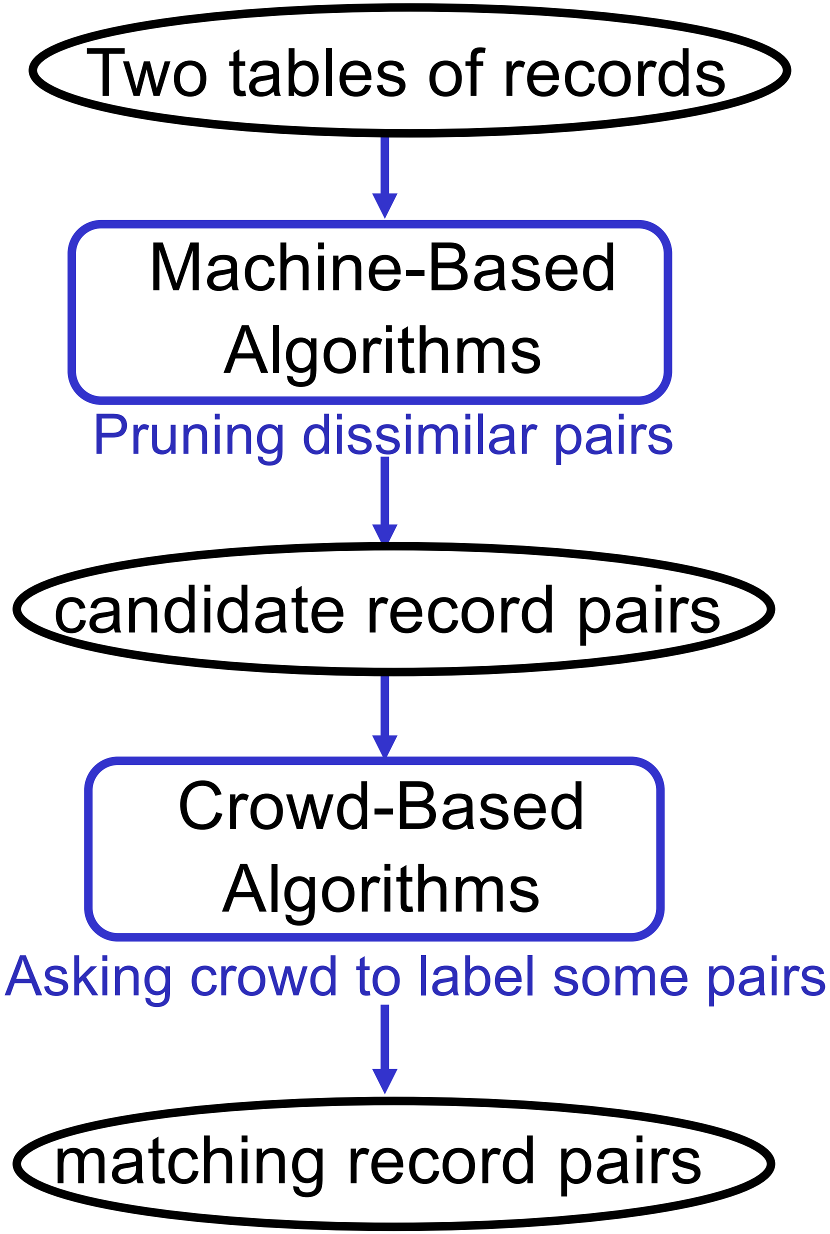
DIMA System

- Similarity-based processing system
- Entity Matching
- Ease to use

CDB System

- Crowd-powered SQL
- New optimization Model
- Entity Matching
- Cost, Latency, Quality

Hybrid Human-Machine EM



iPhone6S
iPhone6SP
Galaxy S7

iPhone 6S 4.7’
iPhone 6S 5.5’
Samsung S7

Pruning 4 dissimilar pairs

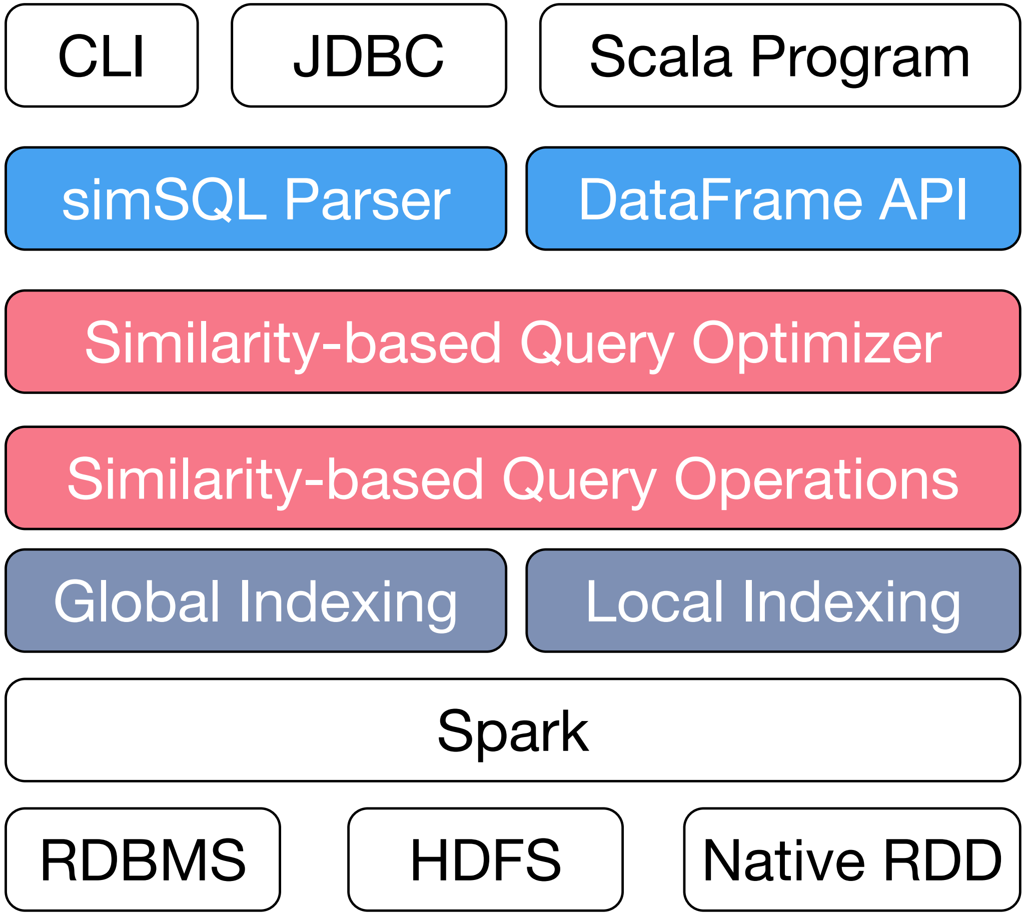
iPhone 6S	iPhone 6S 4.7’	0.75
iPhone 6S	iPhone 6S 5.5’	0.75
iPhone 6SP	iPhone 6S 4.7’	0.72
iPhone 6SP	iPhone 6S 5.5’	0.72
Galaxy S7	Samsung S7	0.5
iPhone6S	Samsung S7	0.1
iPhone6S	Samsung S7	0.1
Galaxy S7	iPhone 6S 4.7’	0.1
Galaxy S7	iPhone 6S 5.5’	0.1

Jaccard
Edit distance
Semantics

Removing 2 non-matched pairs

iPhone 6S	iPhone 6S 4.7’
iPhone 6SP	iPhone 6S 5.5’
Galaxy S7	Samsung S7

Dima: Distributed In-memory Similarity-based System



Query interface

- ✓ **Extended SQL**
- ✓ **Easy to use**

Distributed In-memory Processing Engine

- ✓ **Indexing**
- ✓ **Similarity Operations**
- ✓ **Optimizer**

Support similarity-based query processing

Ji Sun, Zeyuan Shang, **Guoliang Li**, Dong Deng, Zhifeng Bao. Dima: A Distributed In-Memory Similarity-Based Query Processing System. **VLDB** 2017

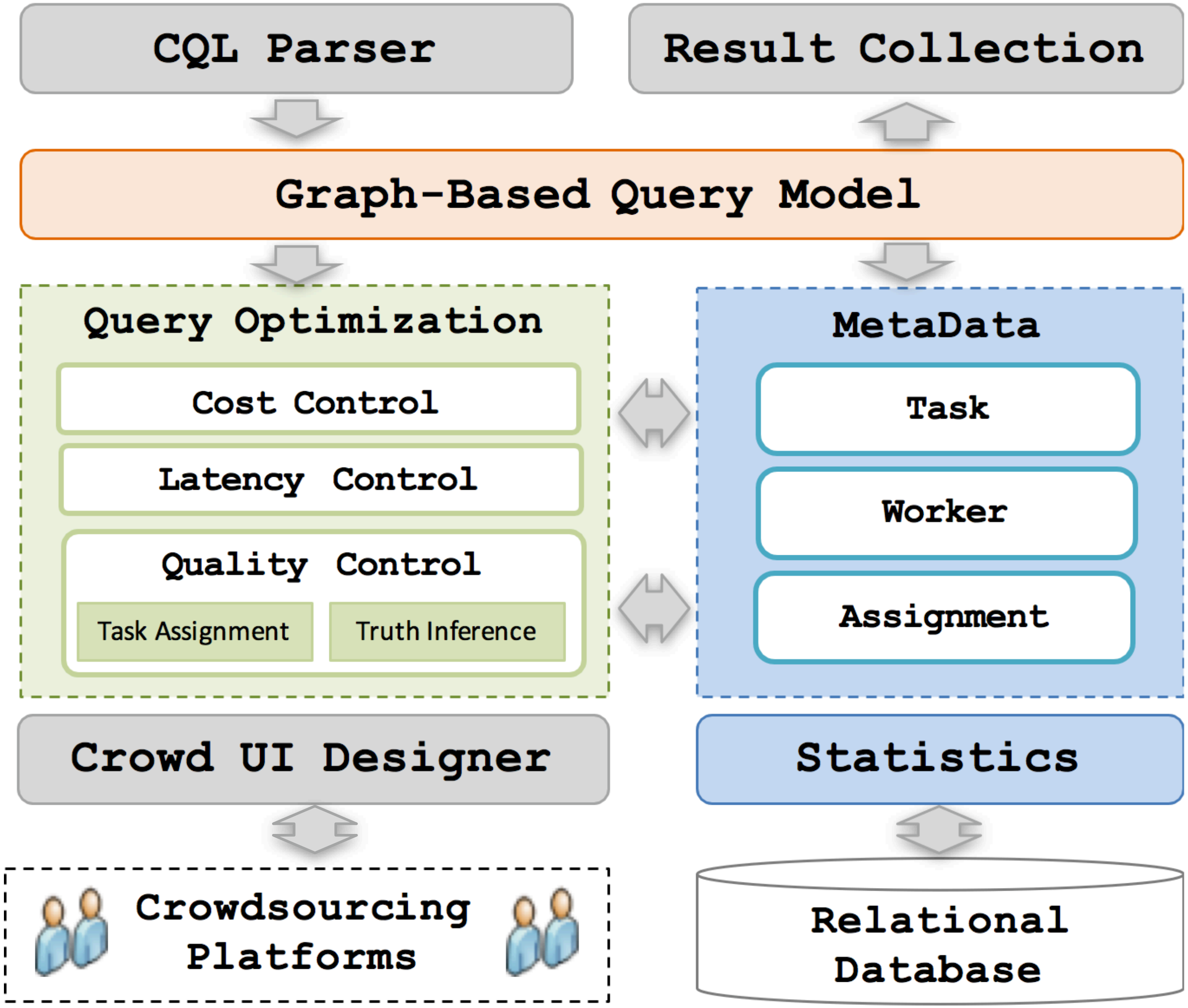
CDB: A Crowd-powered Database

Fine-grained Tuple-level Graph model

Multi-goal optimization

EaseCrowd

ChinaCrowd



Guoliang Li, Chengliang Chai, Ju Fan, Jian Li, Yudian Zheng. CDB: A Crowd-Powered Database. **SIGMOD** 2017.

DIMA: Rule-Based Matching

Name	Storage	Sales
Apple 6S 4.7’	64GB	40K
Apple 6S 5.5’	128GB	30K
Samsung S7	64GB	35K

Brand	Capacity	Price
Apple iPhone6S	64	4000
Apple iPhone6SP	128G	5000
Samsung Galaxy S7	64G	3500

Name is similar to Brand

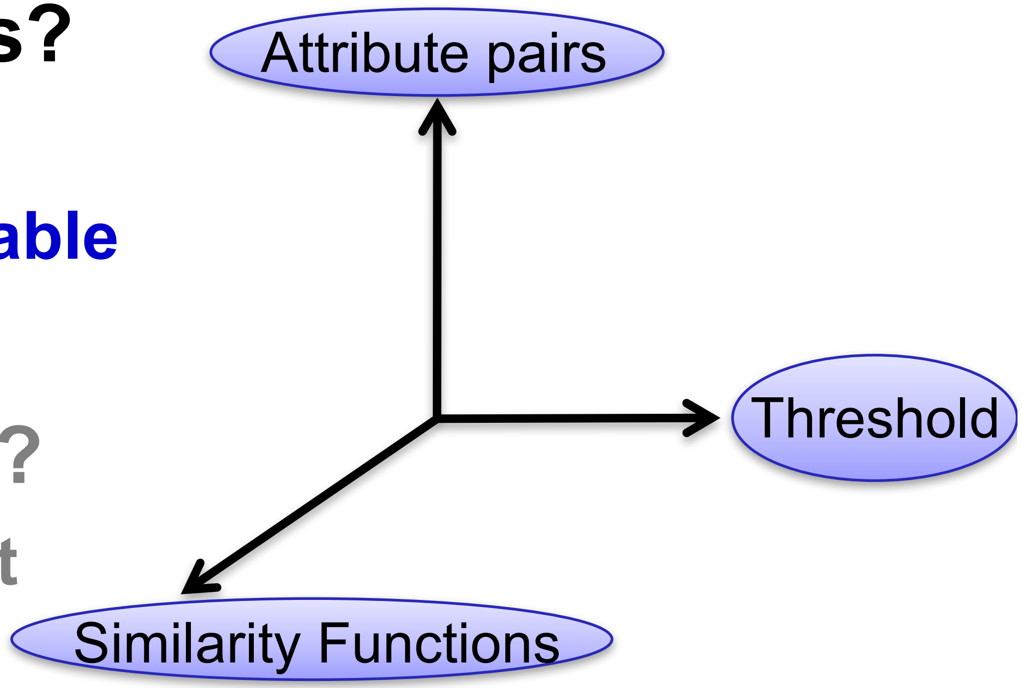
Storage is similar to Capacity

$$\text{Jaccard(Name, Brand)} \geq 0.8 \wedge \text{ED(Storage, Capacity)} < 2$$

Jiannan Wang, **Guoliang Li**, Jeffrey Xu Yu, Jianhua Feng: Entity Matching: How Similar Is Similar. **VLDB**, 2011:622-633.

DIMA: Rule-Based Matching

- **Challenges**
  - **How to obtain the rules?**
    - **High-quality rules**
    - **Explainable, Programmable**
  - **How to apply the rules?**
    - **Avoid Cartesian product**
    - **Fast and Scalable**



Quantifying Rules

M: positive examples   ***N:*** negative examples

$M\varphi$: record pairs that satisfy a rule φ

M_Ψ: record pairs that satisfy a rule set $\Psi=\{\varphi\}$

Objective function: $|M_\Psi \cap M|$- $|M_\Psi \cap N|$

Goal: Find a rule set Ψ to maximize $|M_\Psi \cap M|$-$|M_\Psi \cap N|$

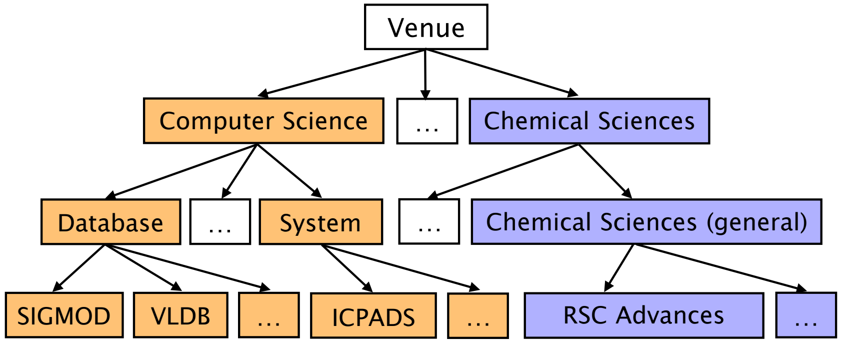
Rule Generation

a,b; Attributes

Attribute pairs

NP-hard

Effective Algorithm



Threshold

δ: [0,1]

Infinite → Finite

✓ F(a,b) on attribute pairs

Similarity Functions

F

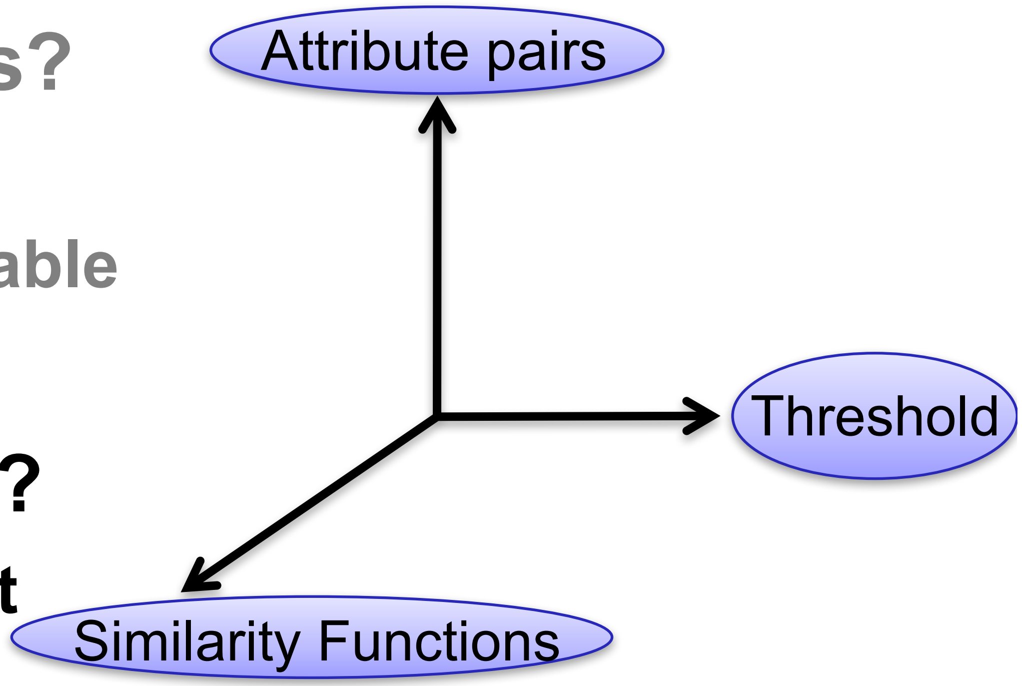
String: edit distance

Set: Jaccard

Knowledge

Rule-Based Matching

- **Challenges**
  - **How to obtain the rules?**
    - **High-quality rules**
    - **Explainable, Programmable**
  - **How to apply the rules?**
    - **Avoid Cartesian product**
    - **Fast and Scalable**



Applying Rules

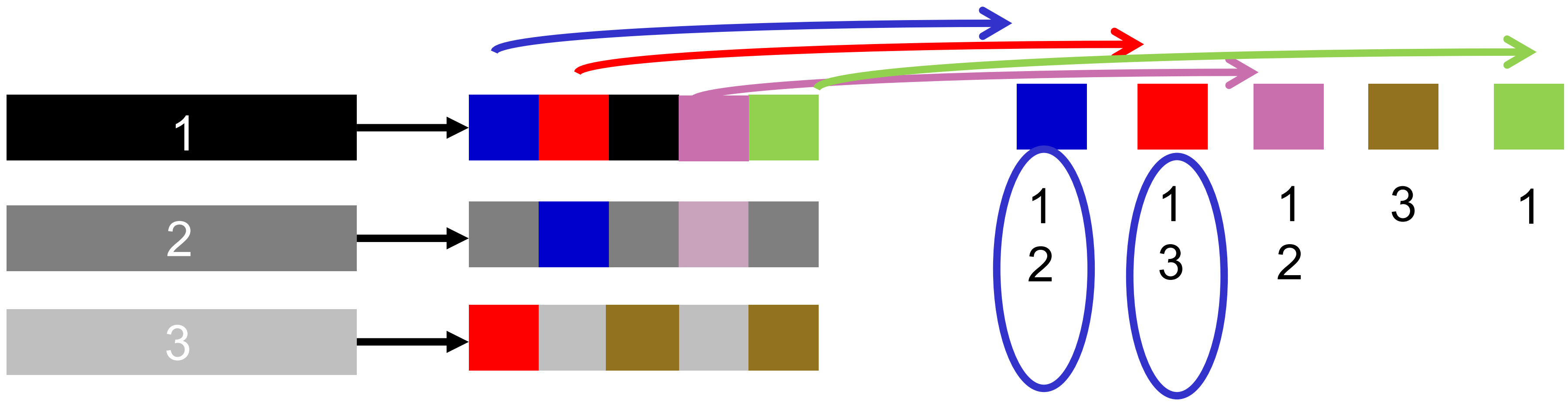
A Rule: Name is similar to Brand

It is expensive to enumerate every record pairs!

- ✓ **10million*10million**

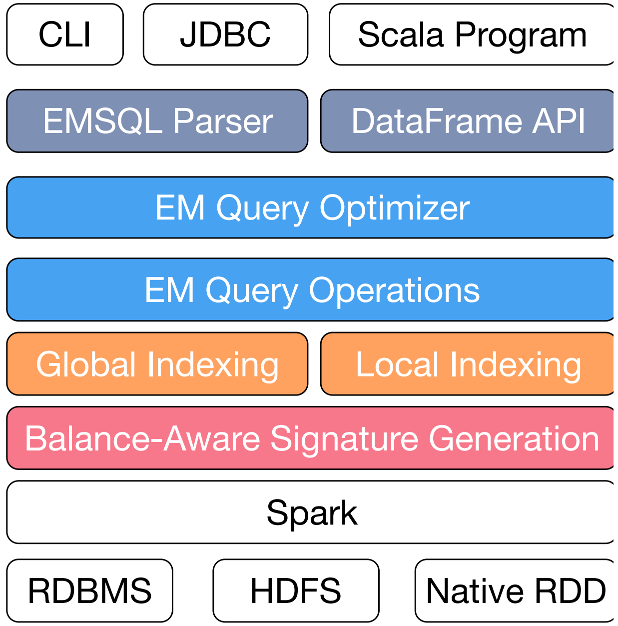
Signature-based Method

- ✓ **If two records do not share a common signature, they cannot be matching**



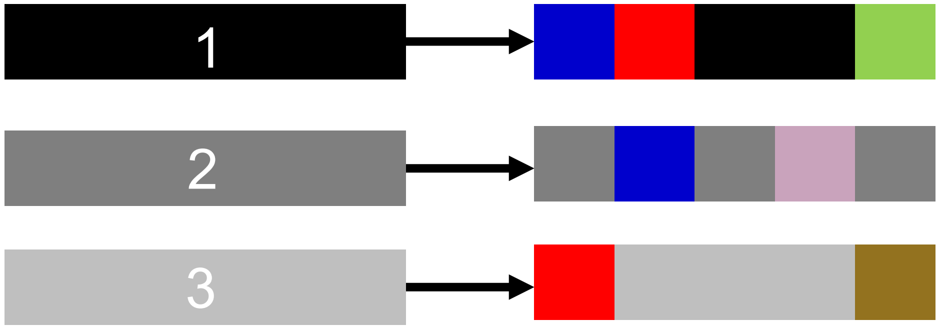
Guoliang Li, Dong Deng, Jiannan Wang, Jianhua Feng: Pass-Join: A Partition-based Method for Similarity Joins. **VLDB**, 2012:253-264.

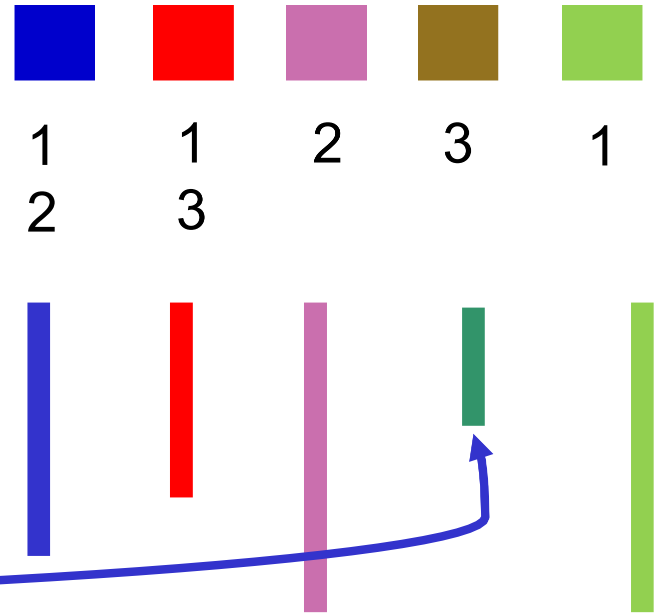
Dima - Signature



Signature-based Method

✓ If two records do not share a common **signature,** they cannot be matching

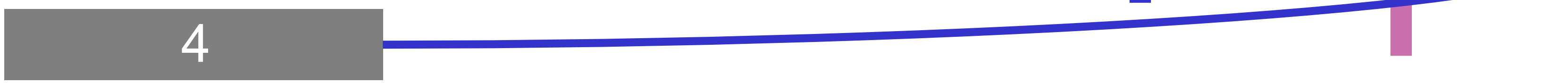




Balance-Aware Signature

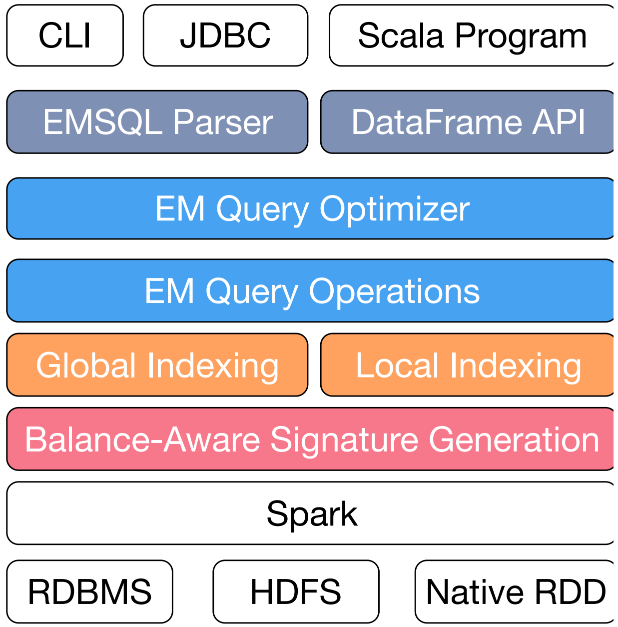
✓ The signatures are selectable

✓ Balance the workload



Dong Deng, **Guoliang Li**, He Wen, Jianhua Feng. An Efficient Partition Based Method for Exact Set Similarity Joins. **VLDB**, 2016

Dima: Load Balance



Challenges

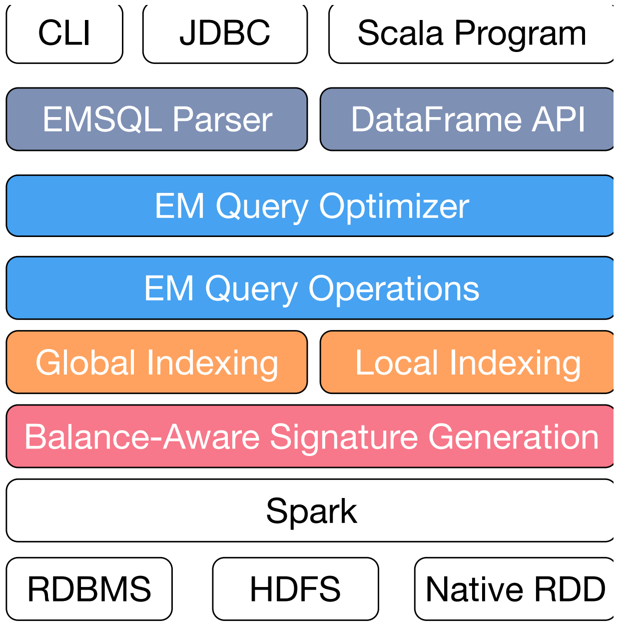
- □ How to **generate signatures**?
  - Partition-based
- □ How to **select the signatures**?
  - Dynamic programming
- □ How to **balance the workload**?
  - NP-hard
  - Greedy algorithms

$$\mathcal{W}_j = \sum_{i=1}^{\eta_l} \Big(b_i \sum_{g \in \mathtt{pSig}^+_{s,i,l} \& \mathcal{P}(g)=j} \mathcal{F}^-[g] + c_i \sum_{g \in \mathtt{pSig}^-_{s,i,l} \& \mathcal{P}(g)=j} (\mathcal{F}^+[g] + \sum_{g \in \mathtt{pSig}^+_{s,i,l} \& \mathcal{P}(g)=j} \mathcal{F}^-[g] + \mathcal{F}^+[g]) \Big)$$

$$b_i = \begin{cases} 1 & Z[i] = 1 \\ 0 & Z[i] \neq 1 \end{cases} \quad c_i = \begin{cases} 1 & Z[i] = 2 \\ 0 & Z[i] \neq 2 \end{cases}$$

$$s.t. \sum_{i=1}^{\eta_l} Z[i] \geq \theta_{|s|,l}.$$

Dima: Indexing



Signature-based partition

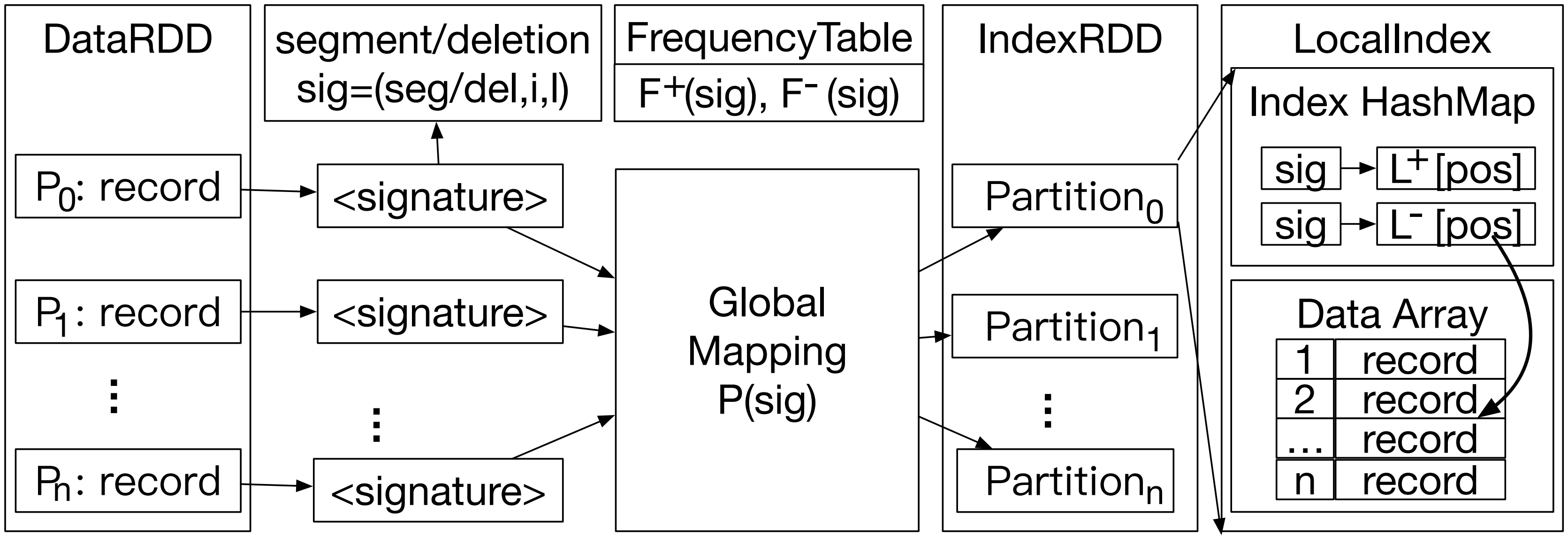
- ✓ Local join
- ✓ Avoid join on different nodes

Global Indexing

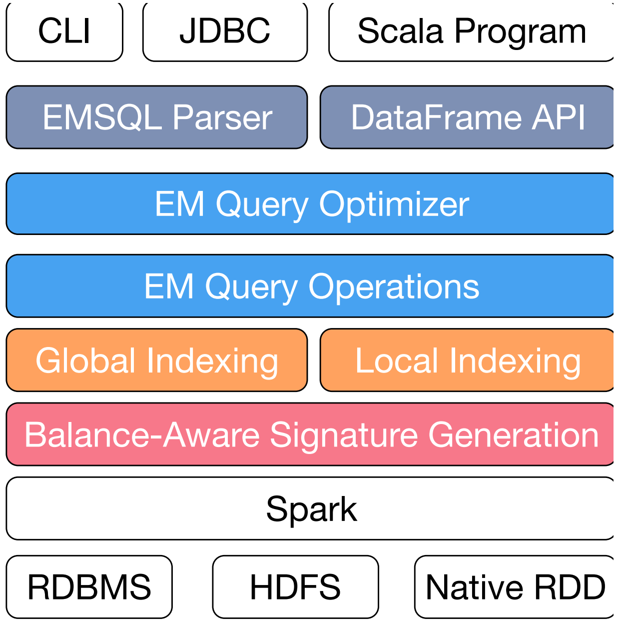
- ✓ Signature→ nodes

Local Indexing

- ✓ Signature → records



Dima: Query Processing

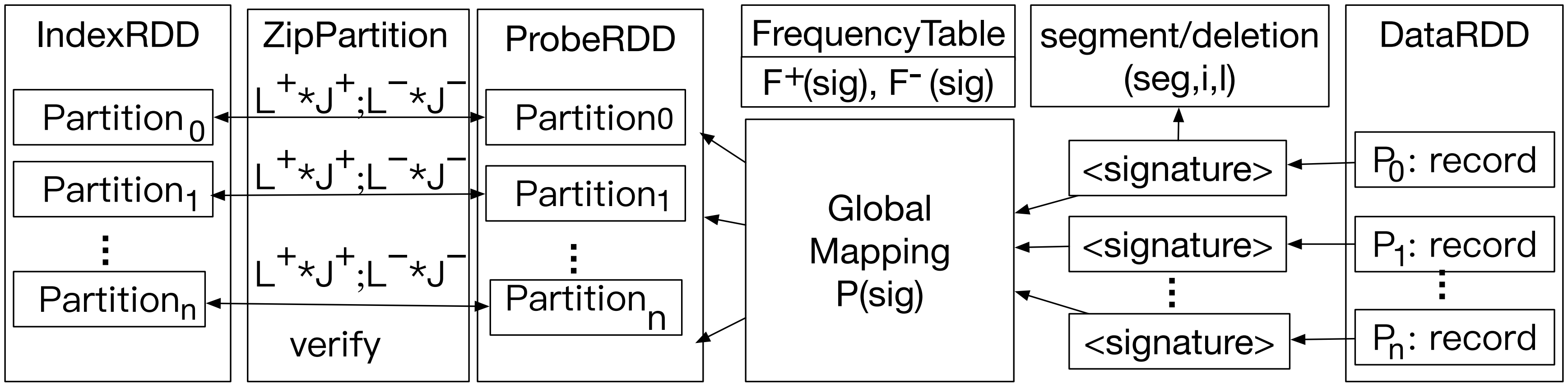


EM Operation

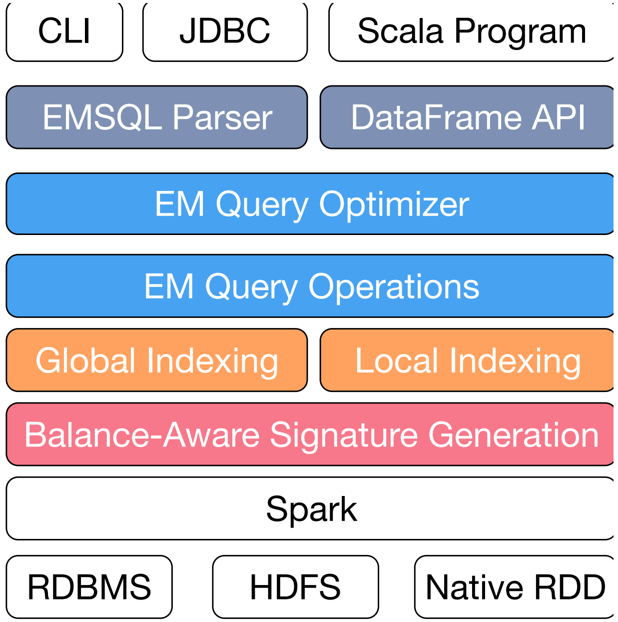
- ✓ Selection
- ✓ Join
- ✓ Topk

EM Query Processing

- ✓ Global
  - ✓ ZipPartition; Balance-aware
- ✓ Local
  - ✓ Avoid Duplicates



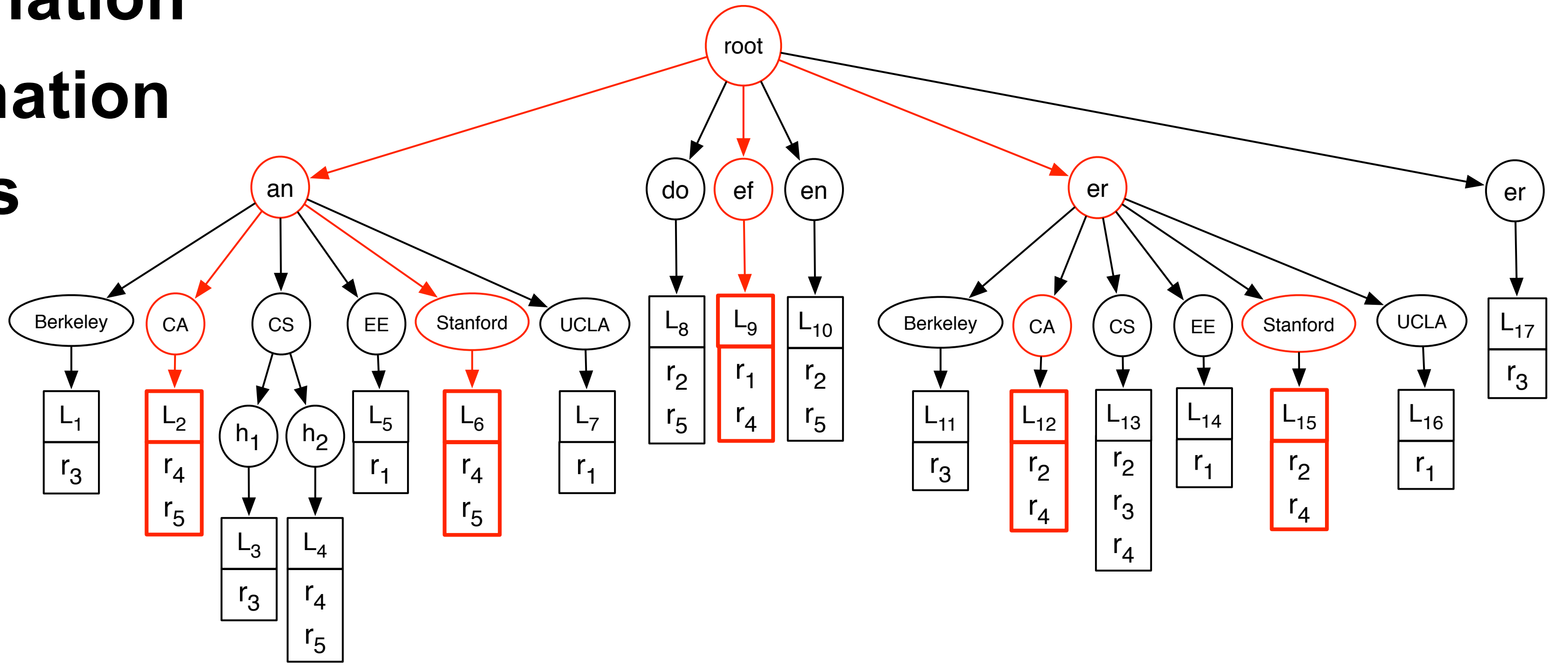
Dima: Query Optimization



$\varphi = \bigwedge \lambda$ where λ: $F(a,b) \geq \delta$

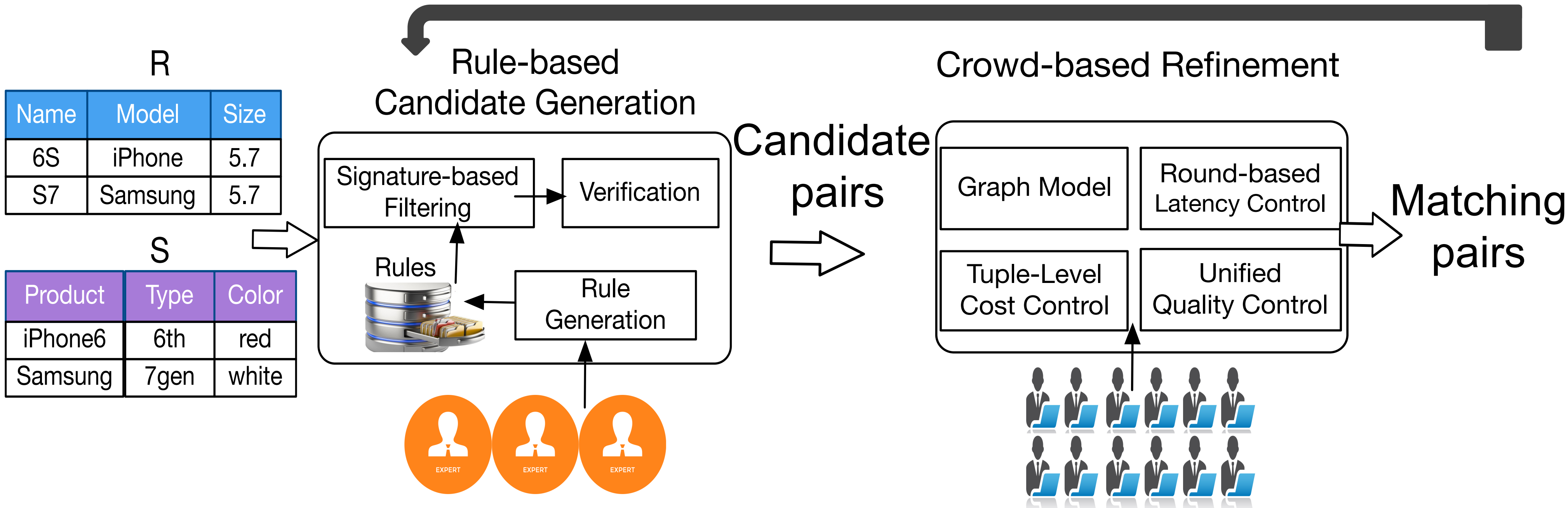
EM Optimizer for multiple attributes

- ✓ **Join order**
- ✓ **Selection Order**
- ✓ **Cost Estimation**
- ✓ **Size Estimation**
- ✓ **#Partitions**



Guoliang Li, Jian He, Deng Dong, Jian Li, Jianhua Feng. Efficient Similarity Search and Join on Multi-Attribute Data. **SIGMOD** 2015.

Hybrid Human-Machine EM



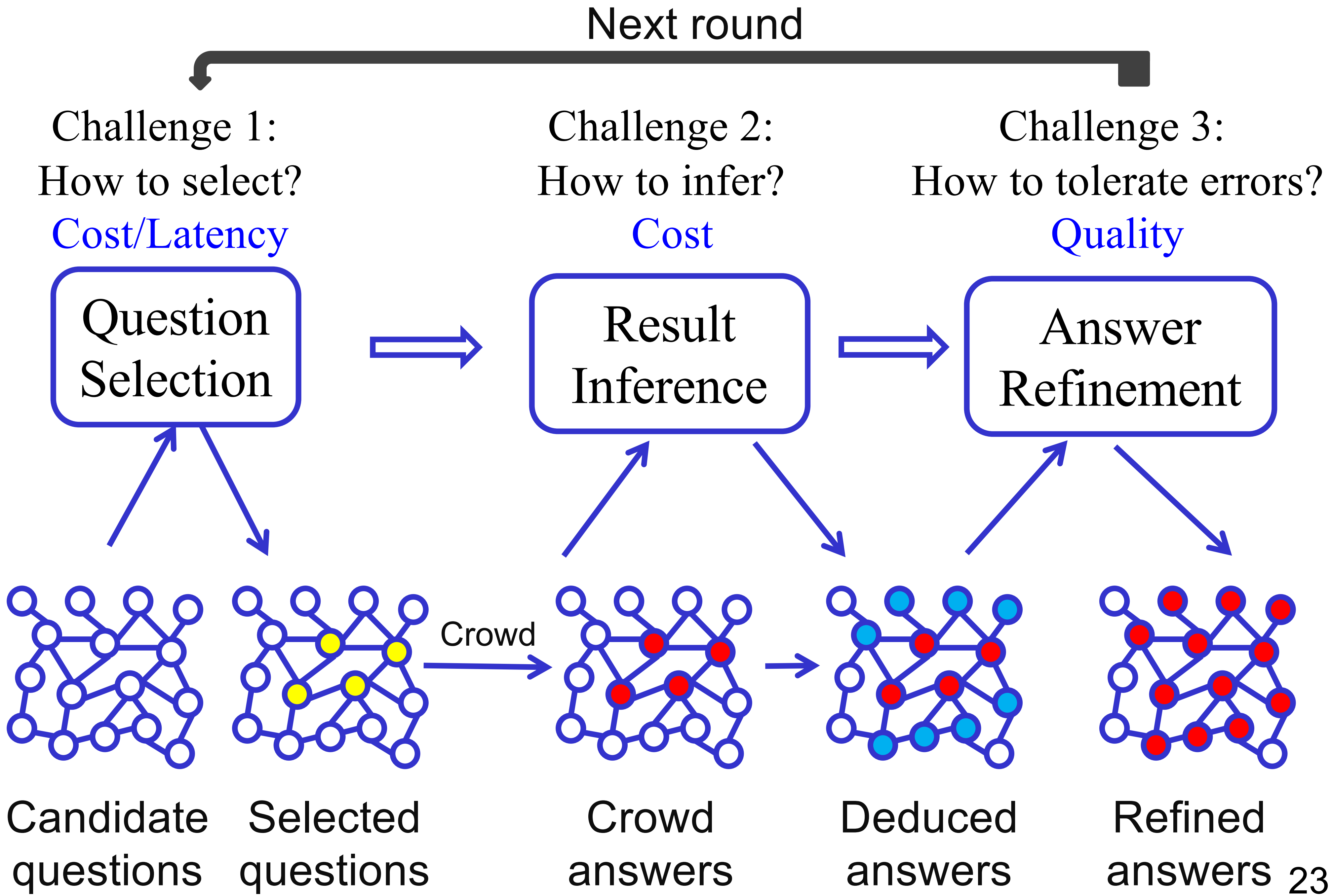
DIMA System

- Similarity-based processing system
- Entity Matching
- Ease to use

CDB System

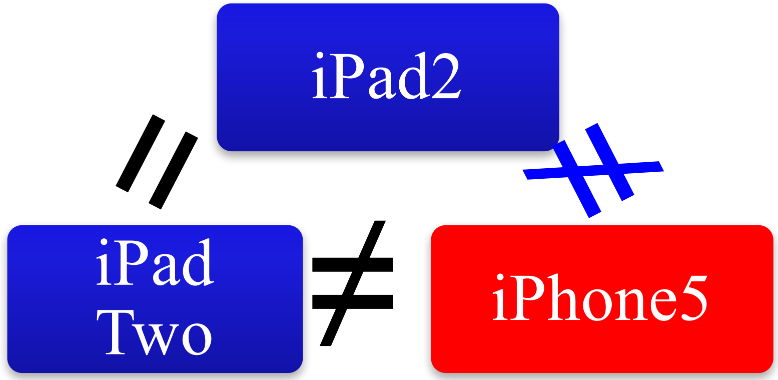
- Crowd-powered SQL
- New optimization Model
- Entity Matching
- Cost, Latency, Quality

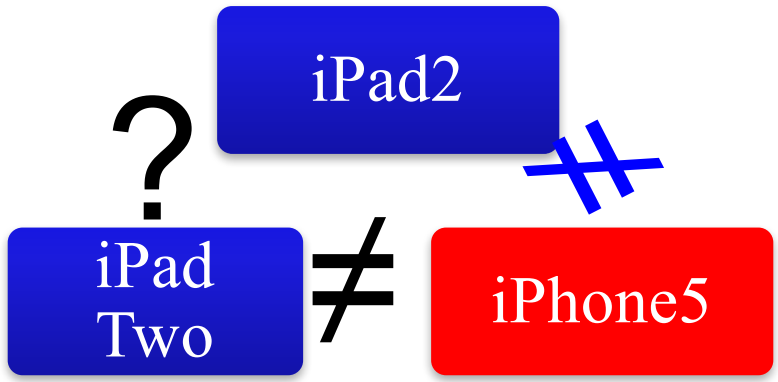
CDB: Selection-Inference-Refine



Inference - Transitivity

- **Challenges**
  - **Labeling order**
    - **A=B, B≠C → A≠C**
    - **B≠C, A≠C → A?B**
  - **Cost**
    - **Optimal Order**
  - **Latency**
    - **Parallel crowdsourcing**
  - **Quality**
    - **Transitivity Errors**
    - **If workers give B=C, then deduce A=C**

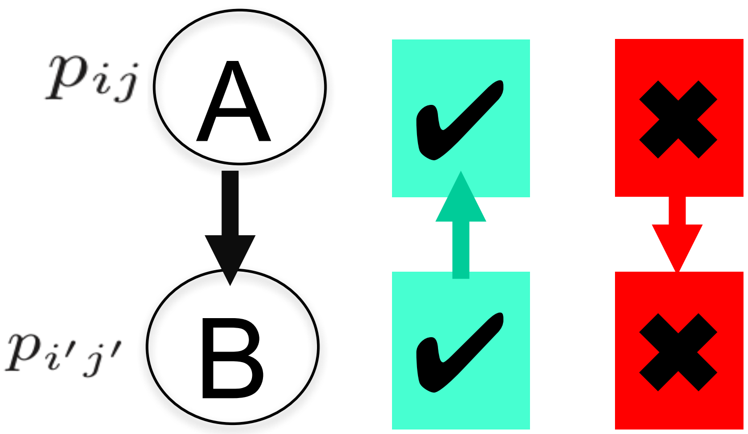


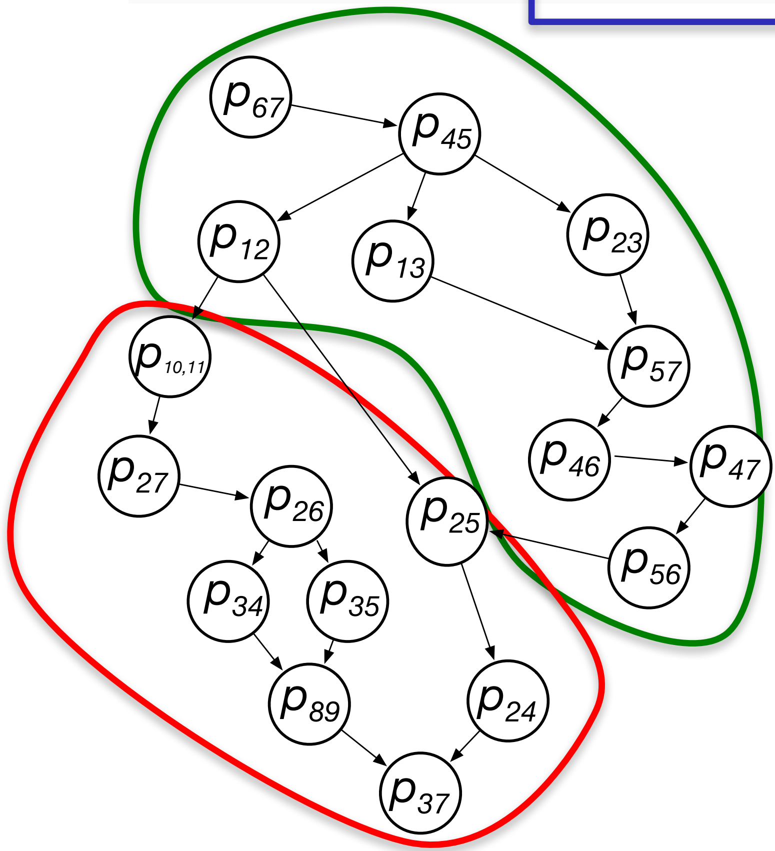


Inference - Partial Order

- Candidate pairs p_{ij}
- Partial order
  - $p_{ij} > p_{i'j'}$ if $s^k_{ij} >= s^k_{i'j'}$

p_{ij}	s^1_{ij}	s^2_{ij}	s^3_{ij}	s^4_{ij}	p_{ij}	s^1_{ij}	s^2_{ij}	s^3_{ij}	s^4_{ij}
p_{12}	0.72	0.4	1	0.88	p_{37}	0.28	0.2	0.33	0
p_{13}	0.75	0.75	0.33	0.8	p_{45}	0.92	1	1	1
p_{23}	0.77	0.5	0.33	0.69	p_{46}	0.69	0.5	0.33	0
p_{24}	0.51	0.2	0.33	0	p_{47}	0.65	0.5	0.33	0
p_{25}	0.53	0.2	0.33	0	p_{56}	0.63	0.5	0.33	0
p_{26}	0.42	0.2	1	0	p_{57}	0.71	0.5	0.33	0
p_{27}	0.45	0.2	1	0	p_{67}	0.94	1	1	1
p_{34}	0.39	0.2	1	0	p_{89}	0.33	0.2	1	0
p_{35}	0.39	0.2	1	0	$p_{10,11}$	0.5	0.25	1	0

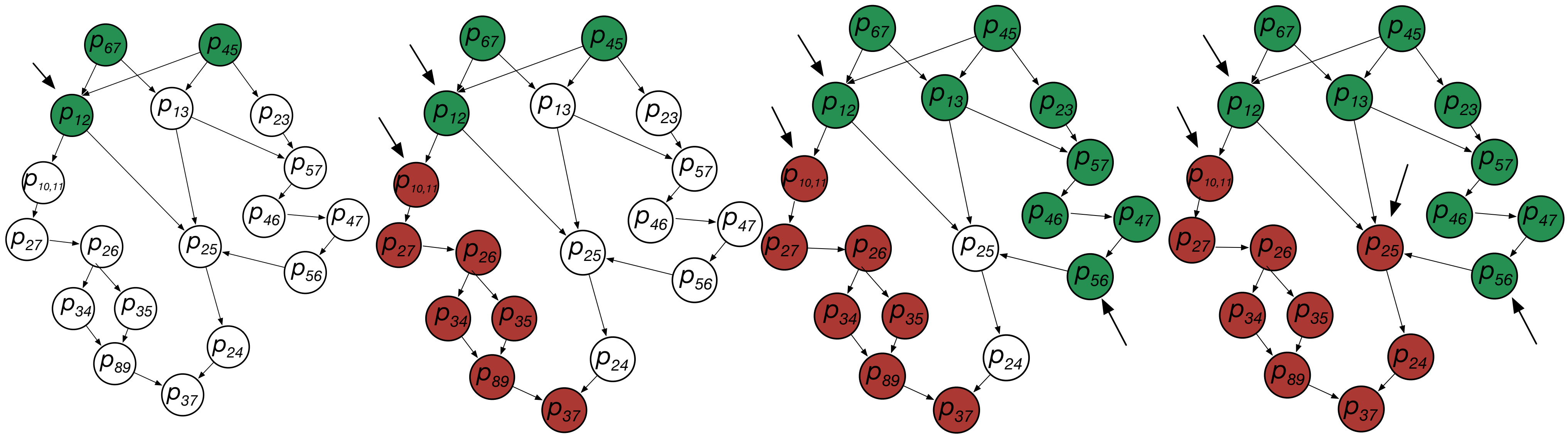




Chengliang Chai, **Guoliang Li**, Jian Li, Dong Deng. Cost-Effective Crowdsourced Entity Resolution: A Partial-Order Approach. **SIGMOD** 2016.

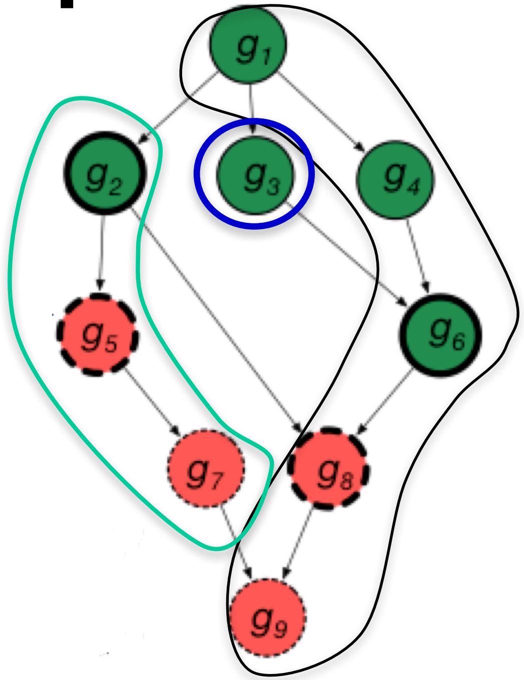
Selection-Inference-Refine Framework

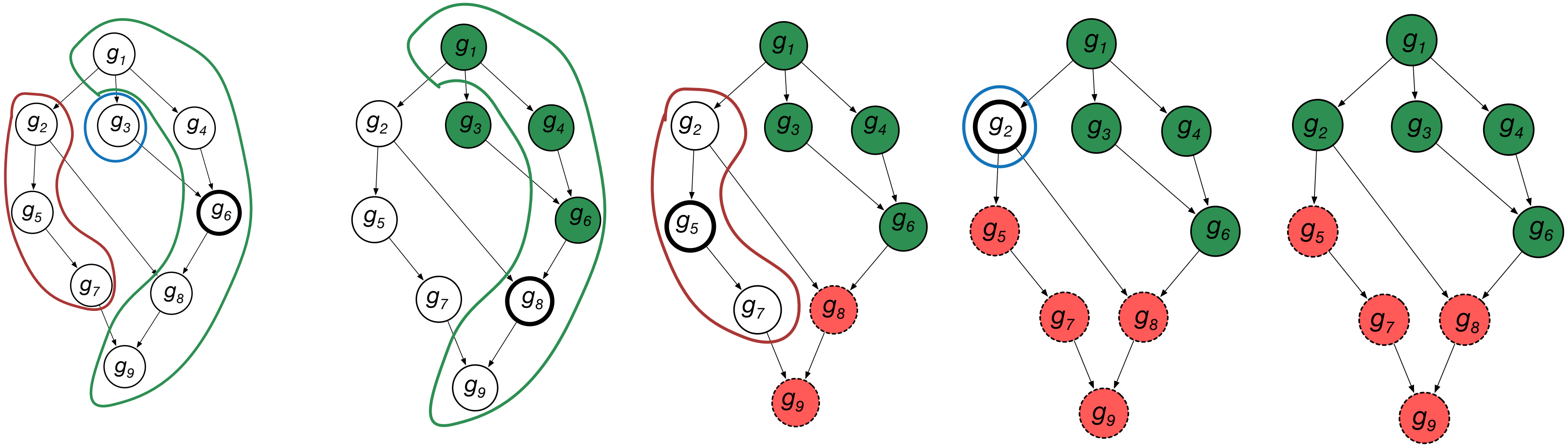
- Selection-Inference-Refine Framework
  - Selection: minimize the number of questions
  - Inference: infer the answers of no-asked questions
  - Refine: tolerate errors of partial order and crowd



Question Selection

- **Serial Algorithms: Ask one question in each iteration**
  - **Comparable Vertices**
    - **O(log|*P*|), |P| is length of path**
  - **Incomparable Vertices**
    - **O(*B*log|*V*|), B is path number**
    - **|V| is vertex number**



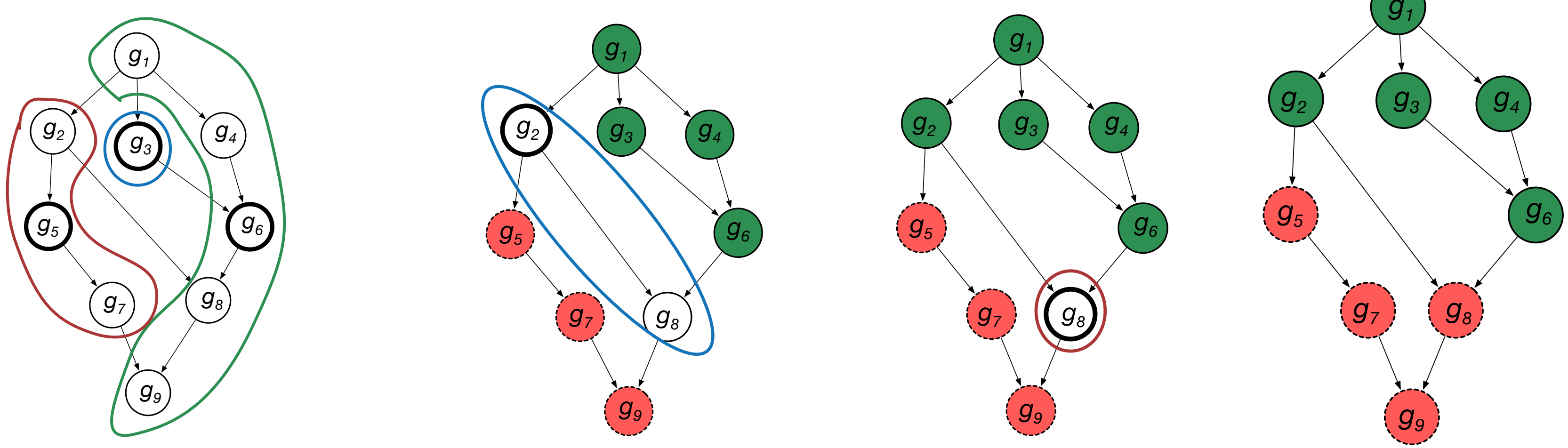


Question Selection

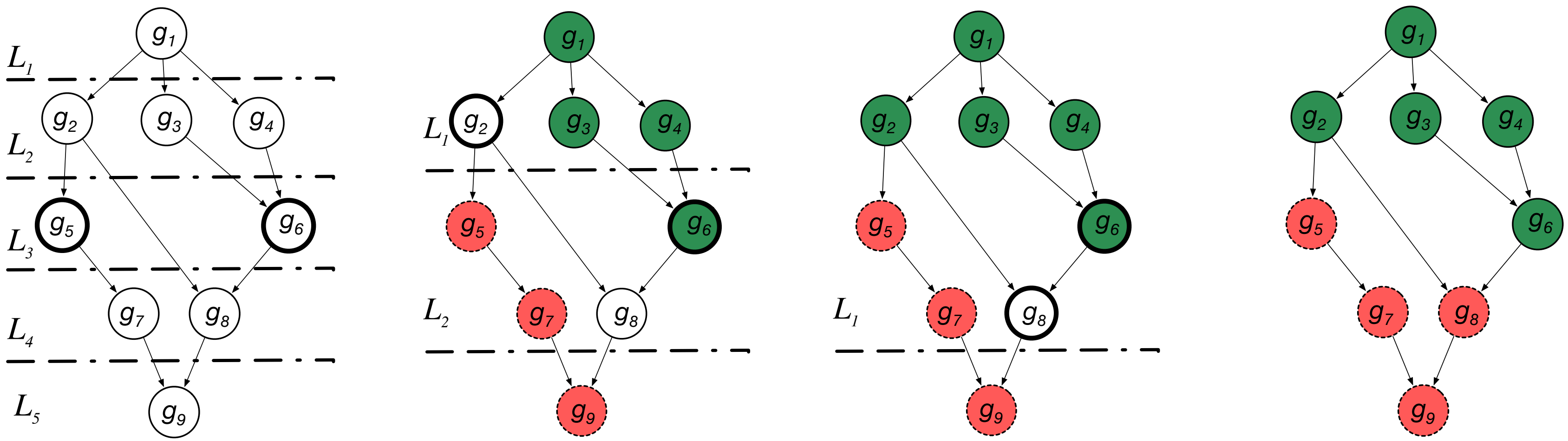
Parallel Algorithm

Select multiple vertices and ask them together in each iteration

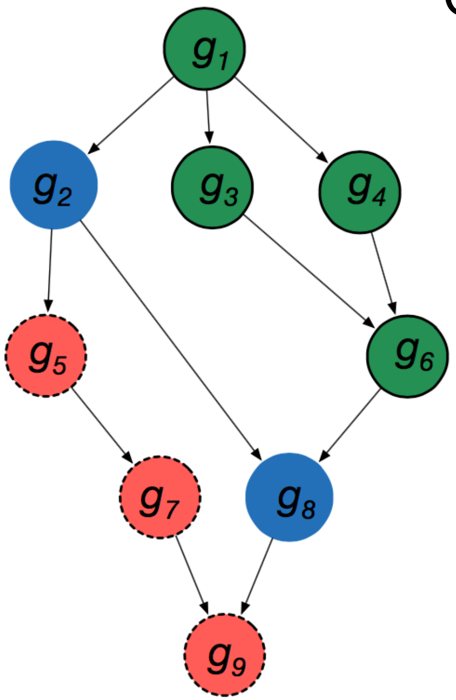
Multi-Path Algorithm



Topology-Sorting-Based Algorithm



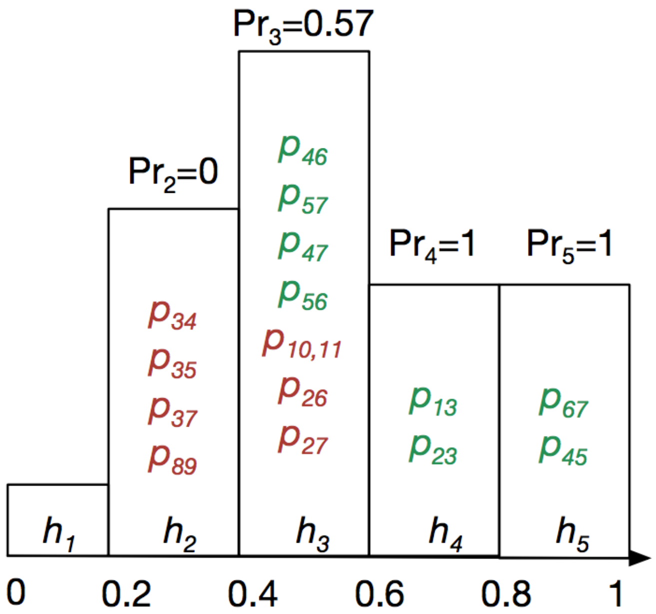
Refinement



Overall weighted similarities of pairs

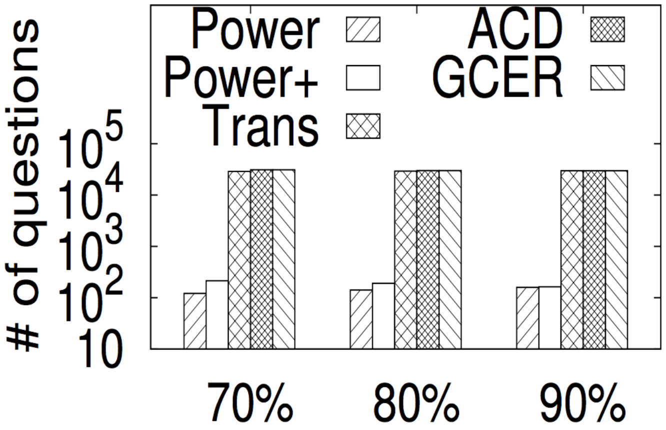
p_{ij}	$\hat{s}_{ij}$	p_{ij}	$\hat{s}_{ij}$
p_{12}	0.72	p_{37}	0.21
p_{13}	0.68	p_{45}	0.97
p_{23}	0.60	p_{46}	0.43
p_{24}	0.28	p_{47}	0.42
p_{25}	0.29	p_{56}	0.41
p_{26}	0.40	p_{57}	0.44
p_{27}	0.41	p_{67}	0.98
p_{34}	0.39	p_{89}	0.37
p_{35}	0.39	$p_{10,11}$	0.44

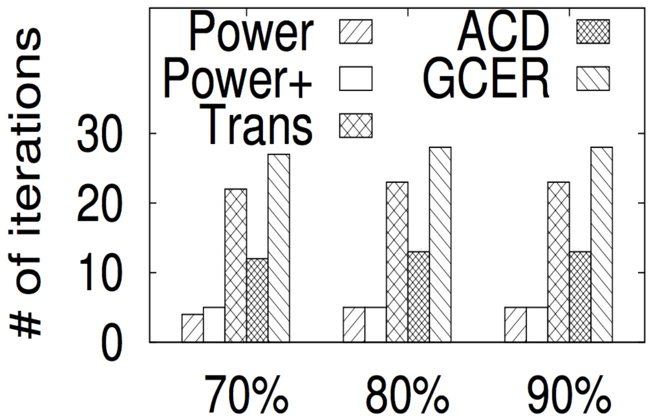
Equi-depth histograms

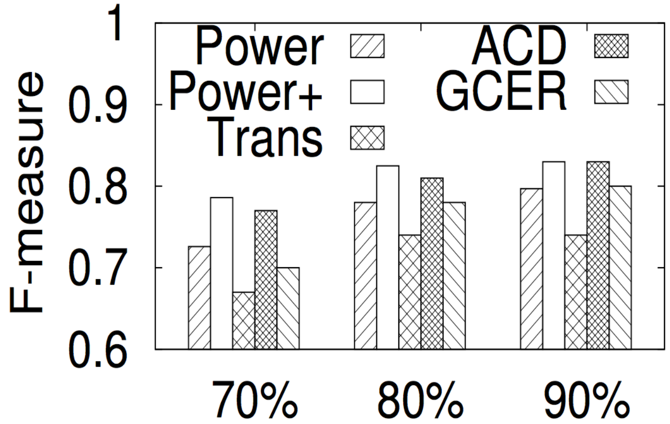


$$\omega_k = \frac{\sum_{p_{ij} \in P^g} s_{ij}^k}{\sum_{p_{ij} \in P^g} \sum_{1 \le t \le m} s_{ij}^t}. \quad \hat{s}_{ij} = \sum_{k \in [1,m]} \omega_k \cdot s_{ij}^k.$$

Results







Cost 100×

Latency 10×

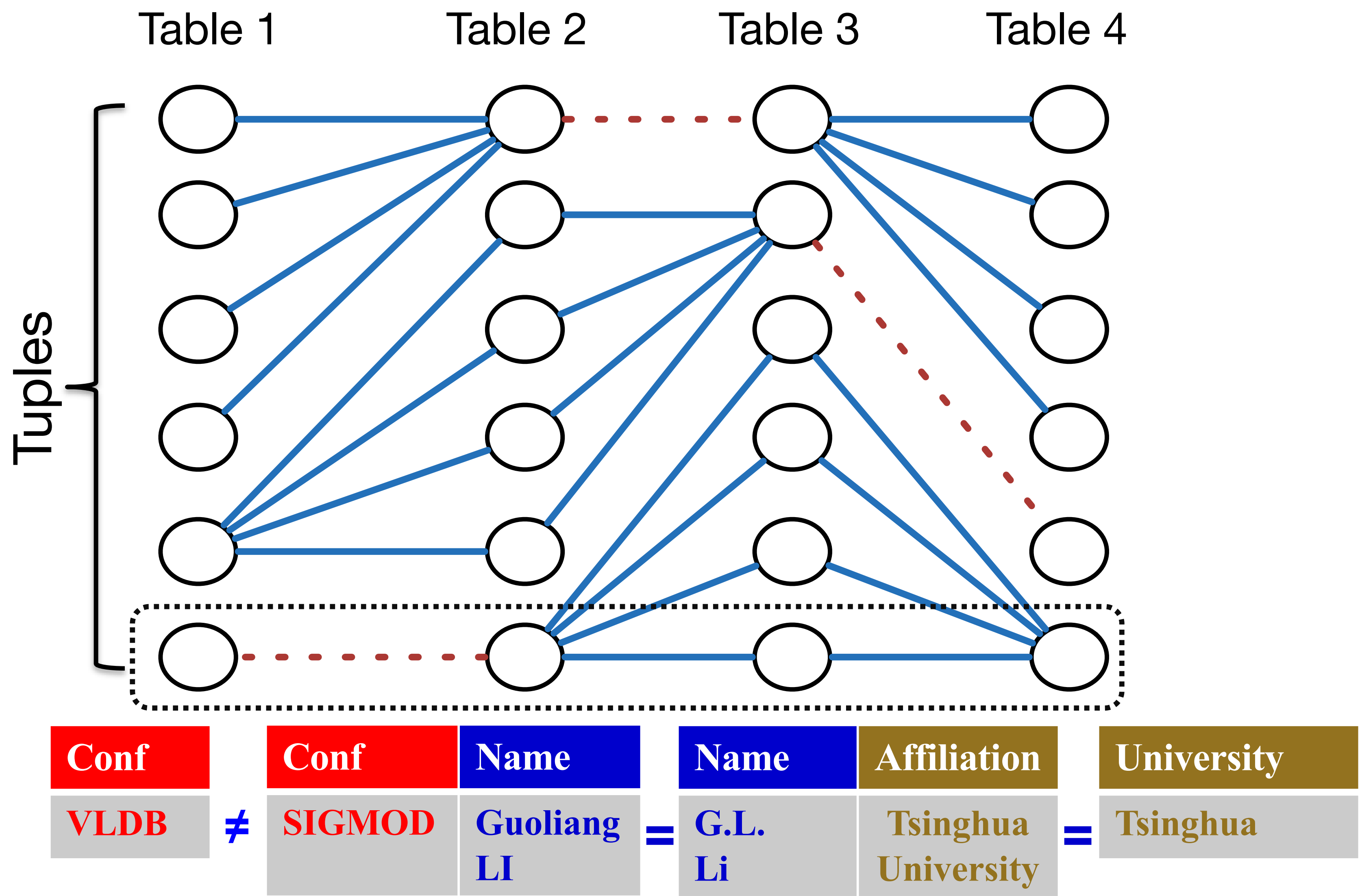
Quality 5%

Crowd-based Method - CDB

- A crowd-powered database system
  - Users require to write code to utilize crowdsourcing platforms
  - CDB encapsulates the complexities of interacting with the crowd
- Limitations of existing systems
  - Coarse-grained optimization - Tree model – Table Level
  - Single-goal optimization - Cost
- Highlights of CDB
  - **Fine-grained optimization - Graph Model – Tuple Level**
  - **Multi-goal optimization – Cost, Latency, Quality**

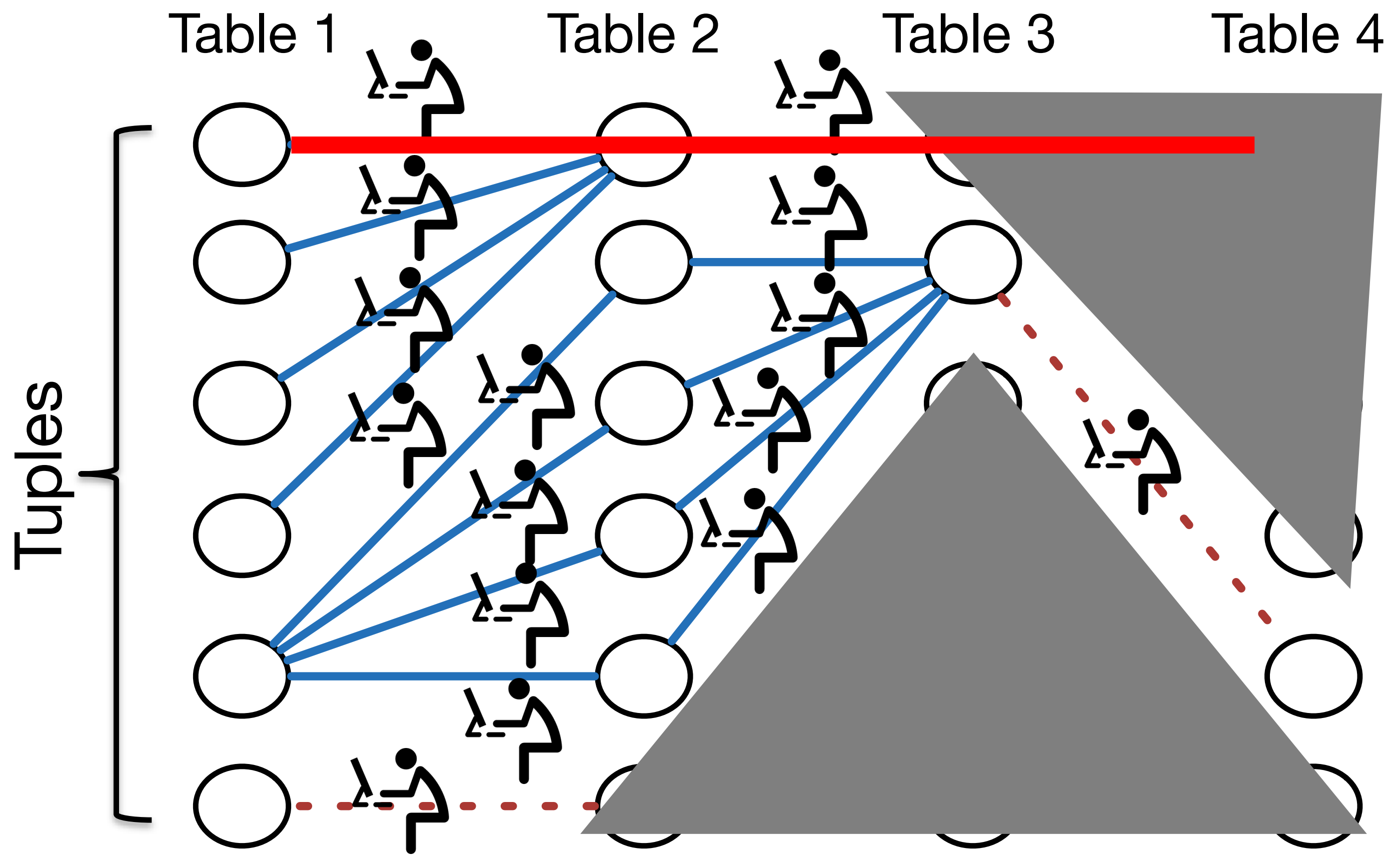
Tree Model vs Graph Model

- Find connected paths with 3 solid edges



Tree Model vs Graph Model

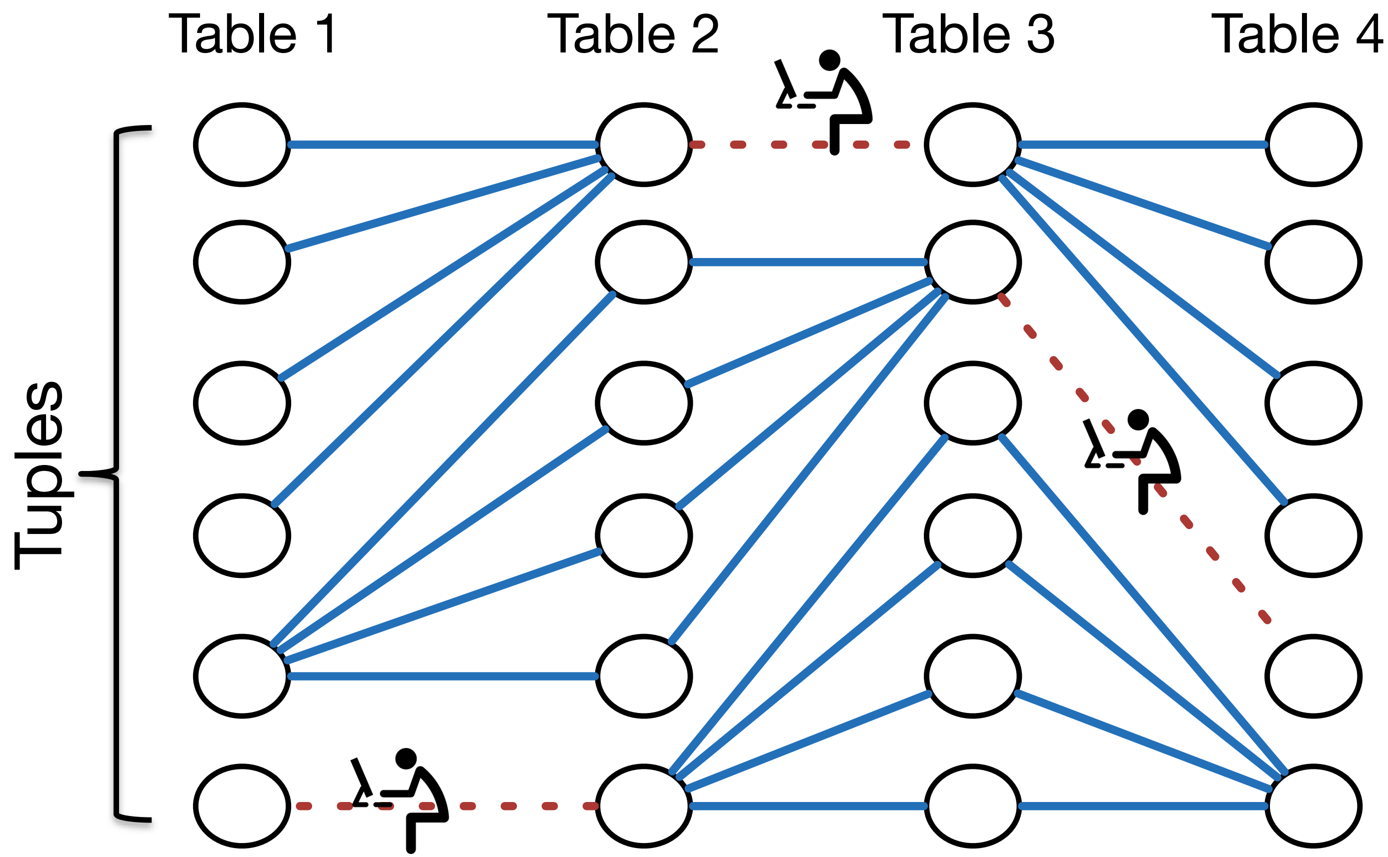
- Find connected paths with 3 solid edges



Optimal Tree Model: 9+5+1=15 tasks
Graph Model: 3 tasks

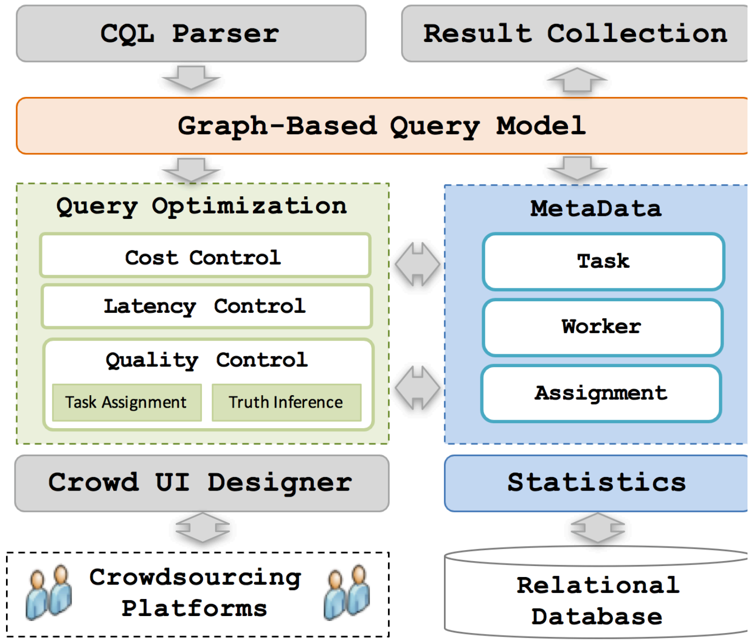
Tree Model vs Graph Model

- Find connected paths with 3 solid edges



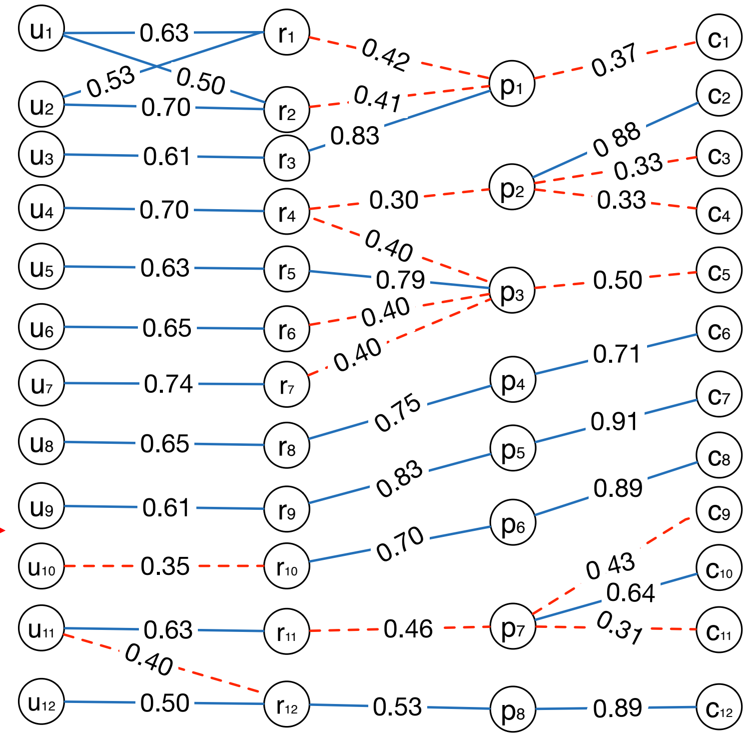
Optimal Tree Model: 9+5+1=15 tasks
Graph Model: 3 tasks

CDB: Graph Model



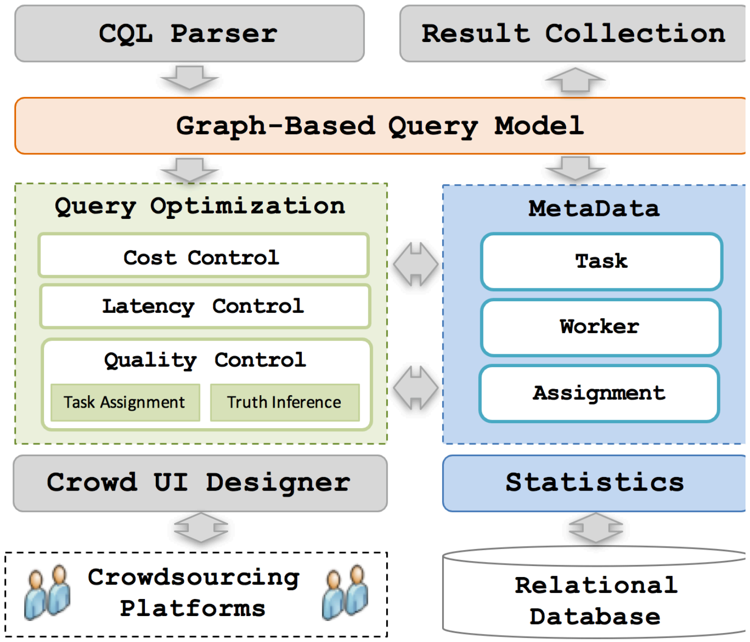
- **Graph Model**
  - Vertices
    - Tuples
  - Edges
    - Join predicate
  - Weight
    - Similarity

```
SELECT * FROM Paper, Researcher, Citation, University
WHERE Paper.Author CROWDEQUAL Researcher.Name AND
Paper.Title CROWDEQUAL Citation.Title AND
Researcher.Affiliation CROWDEQUAL University.Name
```

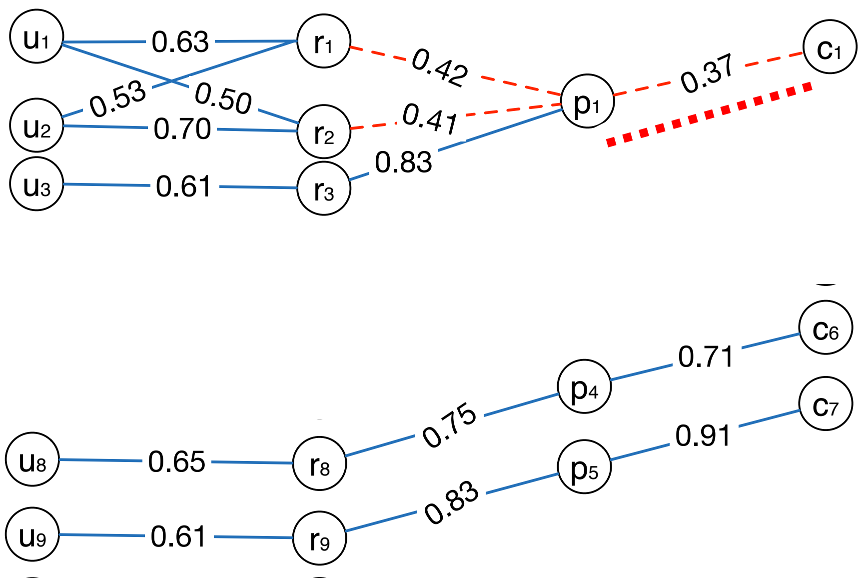


Chengliang Chai, **Guoliang Li**, Jian Li, Dong Deng. Cost-Effective Crowdsourced Entity Resolution: A Partial-Order Approach . **SIGMOD** 2016.

CDB: Cost Control

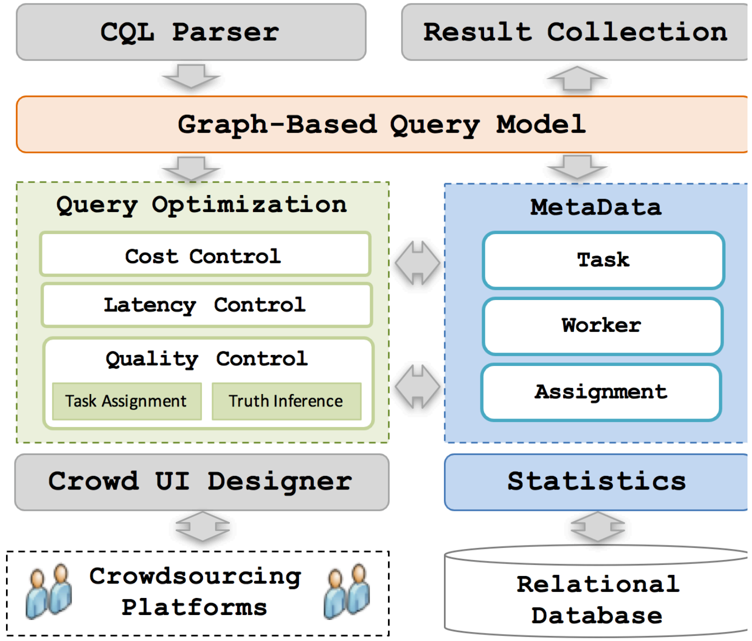


- **Minimize Cost**
  - Find all the results with the minimal cost
- **Budget (pay as you go)**
  - Find the most results with a given budget (B tasks)
- **Expectation-based Method**
  - Pruning ability to cut the graph

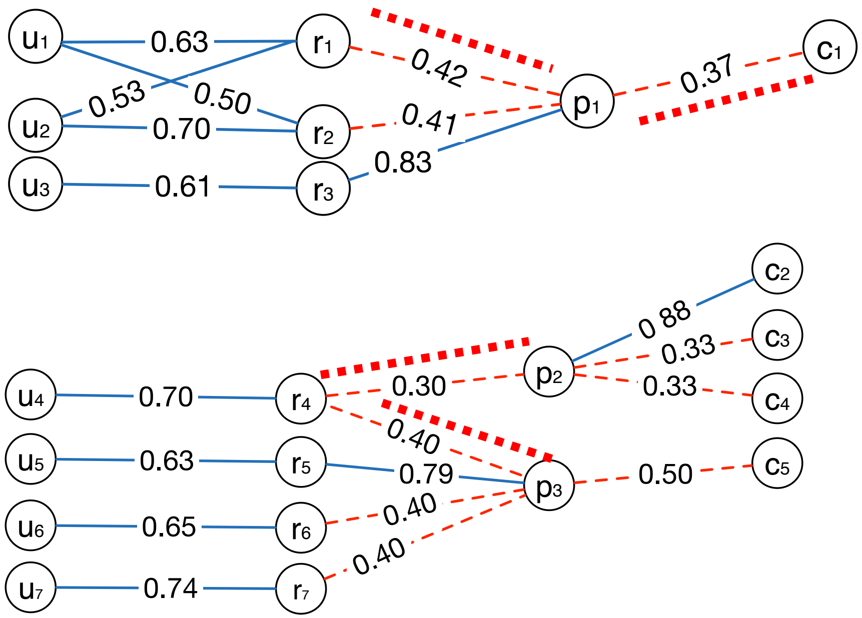


$$\mathbb{E}(t,t') = \frac{\prod_{i=1}^{x}(1-\omega(t,t_i))}{x}\alpha + \frac{\prod_{i=1}^{y}(1-\omega(t_i,t'))}{y}\beta$$

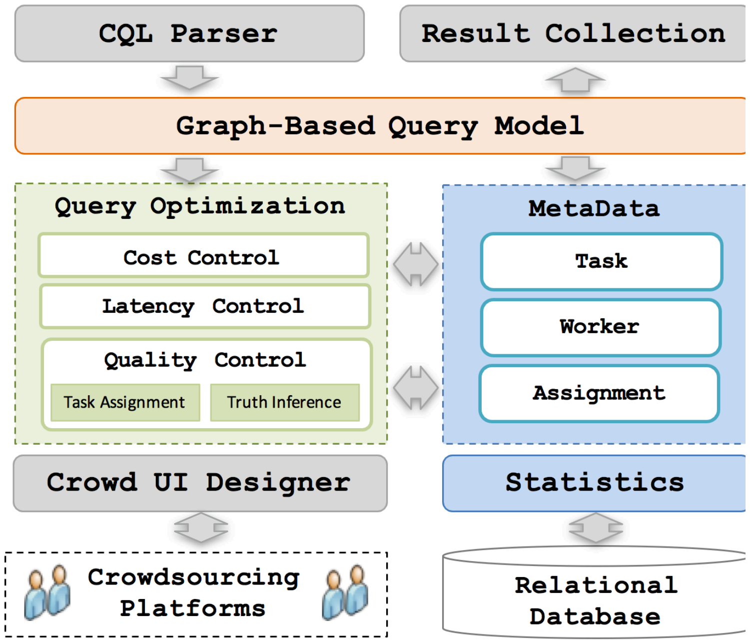
CDB: Latency Control



- Which tasks can be asked in parallel
  - Tasks have correlations
  - Tradeoff: cost and latency
  - Minimize the number of rounds without increasing the cost
- Task batching
  - Connected components
  - Edges containing tuples from the same table



CDB: Quality Control



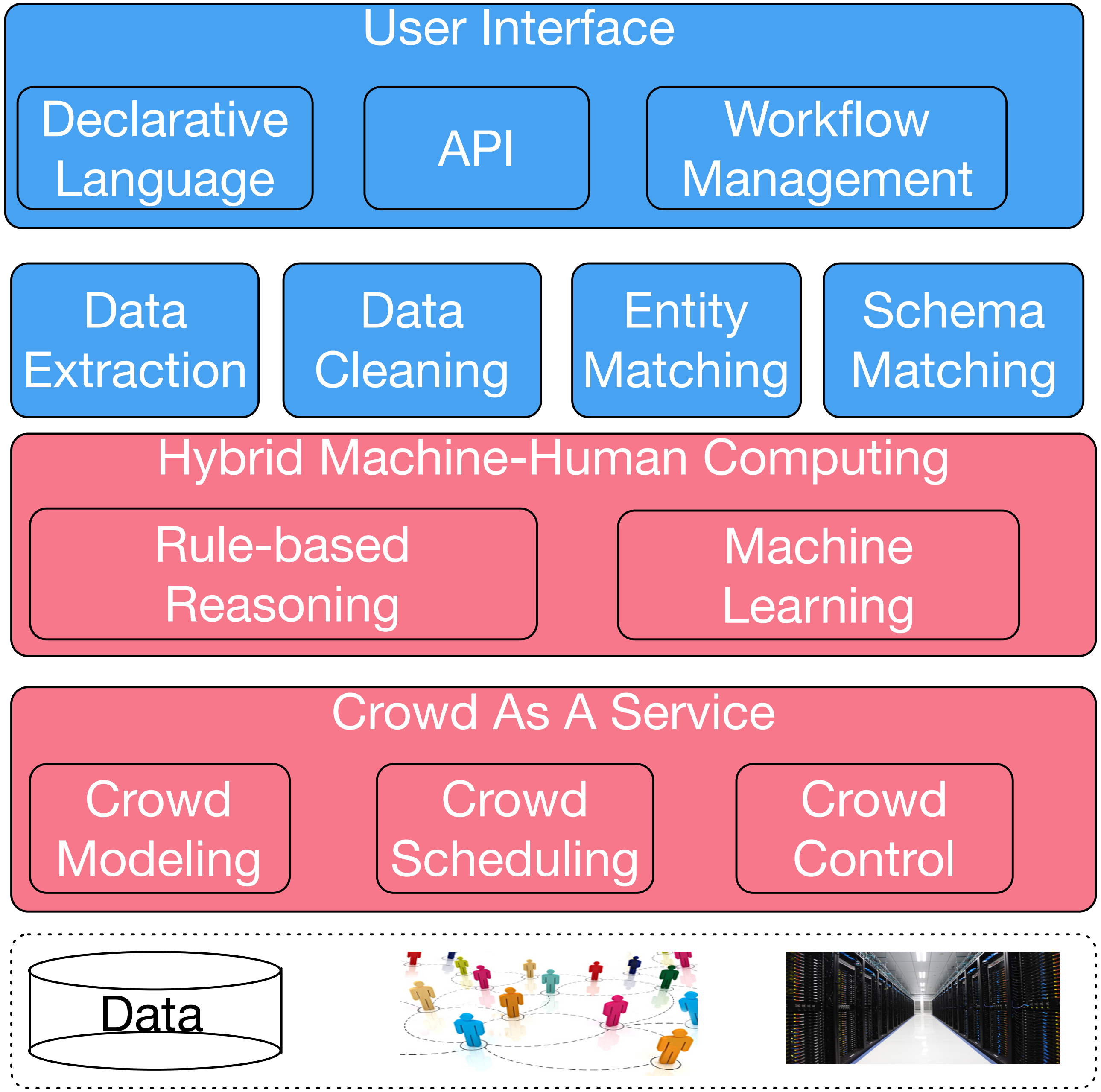
- **Truth Inference**
  - A unified inference model
    - Single choice
    - Multiple choice
    - Fill/Collection
- **Online Task Assignment**
  - Worker Model
  - Task Model
  - Worker answer prediction

$$p_i = \frac{\prod_{(w,a)\in V_t} (q_w)^{\mathbb{1}\{i=a\}} \cdot \left(\frac{1-q_w}{\ell-1}\right)^{\mathbb{1}\{i\neq a\}}}{\sum_{j=1}^{\ell} \prod_{(w,a)\in V_t} (q_w)^{\mathbb{1}\{j=a\}} \cdot \left(\frac{1-q_w}{\ell-1}\right)^{\mathbb{1}\{j\neq a\}}}$$

$$\mathcal{C}(t) = \sum_{\{(w,a)\in V_t\}\wedge\{(w',a')\in V_t\}\wedge\{w\neq w'\}} \frac{sim(a,a')}{\binom{|V_t|}{2}}$$

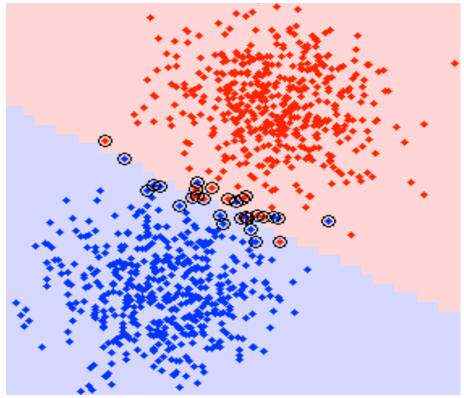
$$\mathcal{I}(t) = \mathcal{H}(\vec{p}) - \sum_{i=1}^{\ell} \left[\, p_i \cdot q_w + (1-p_i)\cdot \frac{1-q_w}{\ell-1} \,\right] \cdot \mathcal{H}(\vec{p'})$$

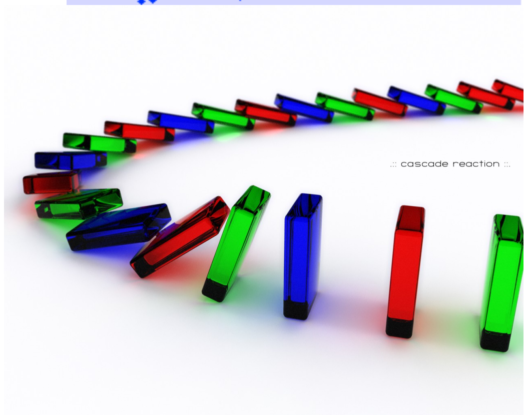
Data Integration As A Service



Lessons Learned

- Human is important in data integration
- Machine step
  - Rules are important
  - Require high-quality examples
- Crowd step
  - Crowd is double-edged sword
    - high quality for easy tasks
    - low quality for hard tasks
  - Inference is important
    - Can reduce the cost significantly
    - But may sacrifice quality





All the codes are open-sourced at
https://github.com/TsinghuaDatabaseGroup/

Thank You!