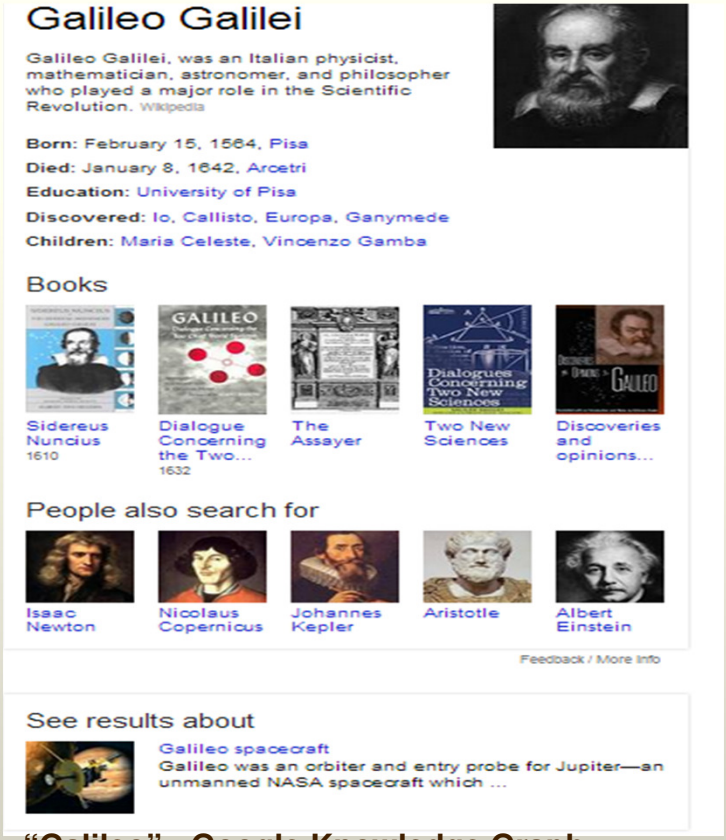
Towards Advanced Search in Complex Graphs

Yinghui Wu

University of California, Santa Barbara

Searching complex graphs



"Galileo" : Google Knowledge Graph Search (May, 2012)

"People who like things I like" – an example of Facebook graph search query (January, 2013)

Main challenges

- ✓ Computationally efficient query models
  - Enabling graph search with semantic similarity (ontology-based search)
  - Enhancing approximate subgraph querying (Nema, simulation-based graph pattern matching)
- ✓ Distributed graph searching: partition management (Sedge)
- ✓ Vision of a distributed, complex graph search engine
- ✓ Conclusion

Towards Advanced Search in Big and Complex Graphs:

computationally efficient query models

Computational efficient graph querying

shape

- **G-ray** [Tong et. al., KDD '07]: approximate matches by preserving the query shape
- **Dual-simulation** [Shuai et.al, VLDB '12]: preserving edge-edge matching

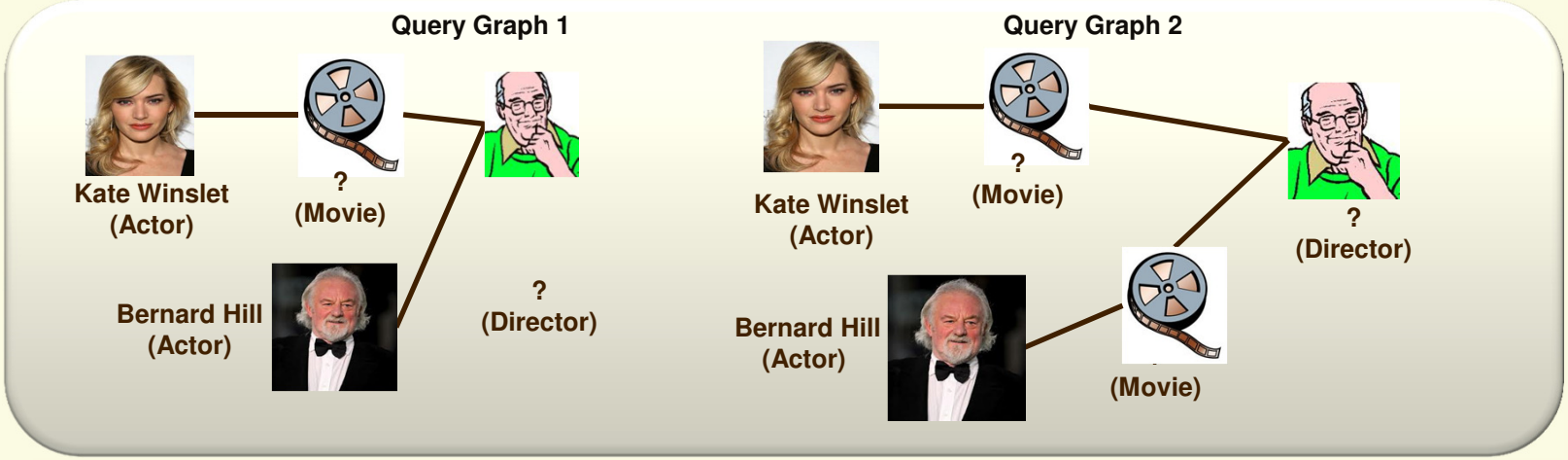
Edit distance

- **TALE[**Tian et. al., ICDE '08]
- **SIGMA** [Mongiovi et. al., J. Bioinformatics & Computational Biology '10]
- **SAPPER** [Zhang et. al., VLDB '12]: two-hop neighborhood indexes
- **Trust** [Sambhoos et al., Information Fusion '10]: - approximate graph matching with number of edge misses and node label mismatch
- **P-Hom** [W.Fan et.al., VLDB '10]: edge-path matching

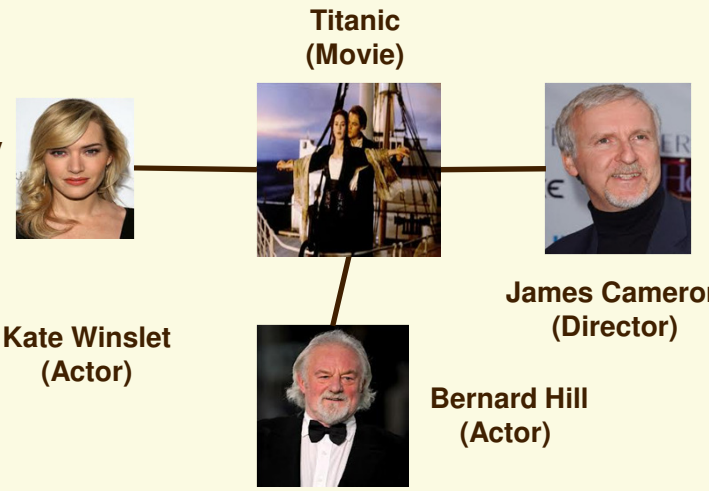
Proximity+label similarity

- **SAGA** [Tian et. al., Bioinformatics '06]: both structure and node label similarity
- **NESS, Nema** [Arijit et.al, SIGMOD '12, VLDB '13]
- **Bounded simulation** [W.Fan et.al, VLDB '10]: edge-path matching

Example: a simple IMDB query



- Multiple representation of a simple query
- Matches may look very different from queries (e.g., large edit distance)
- relax rigid matching constraints of subgraph isomorphism



Searching big graphs

NESS and Nema (SIGMOD'11, VLDB'13)

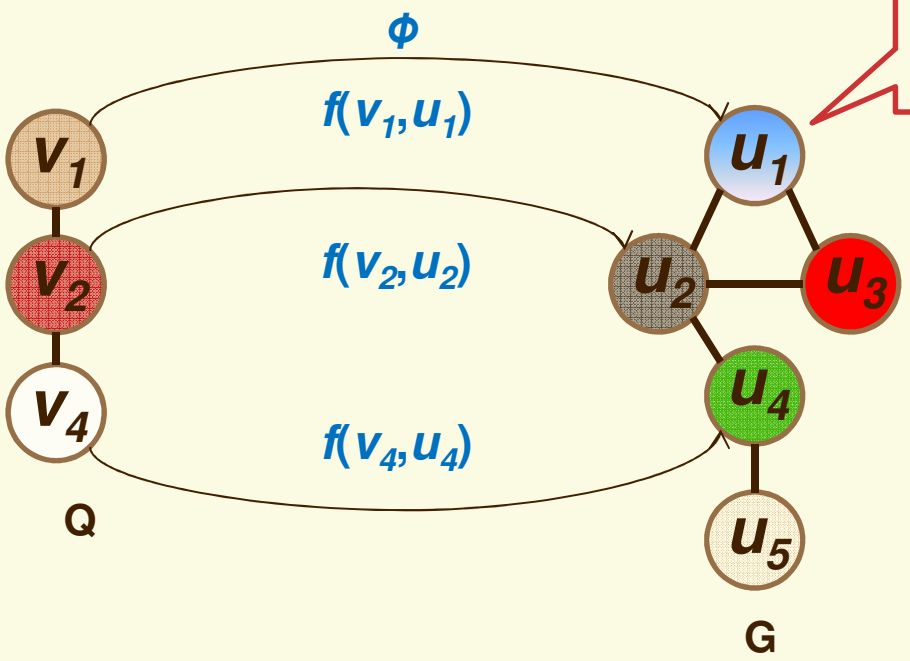
- Goal: find a matching function Φ with minimum matching cost $C(\Phi)$

$$C(\phi) = \sum_{v \in V_Q} f(v, \phi(v)).$$

Individual node matching cost : Sum of label similarity and neighbourhood (vector) similarity of v and Φ(v)

$\{\langle u_2, 0.5\rangle, \langle u_3, 0.5\rangle, \langle u_4, 0.25\rangle\}$

(2 hop with propagation factor 0.5

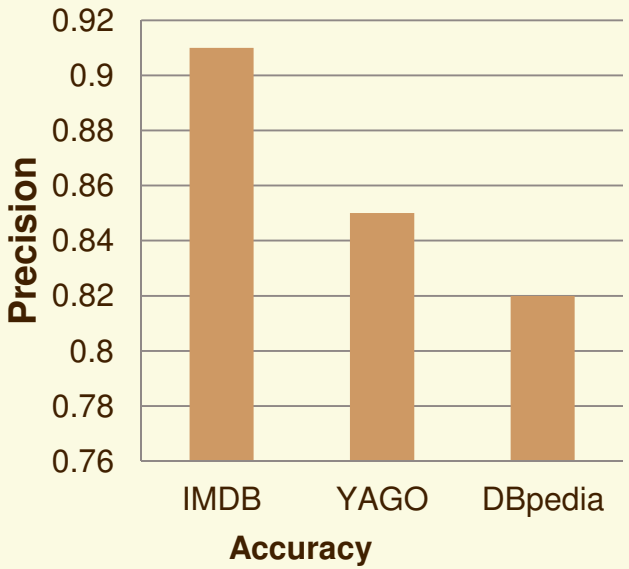


Iterative Inference algorithm: a Max-Sum Inference in Random Markov Field
- Time complexity: $O(n_G \cdot n_Q + T \cdot n_Q \cdot m_Q \cdot d_Q)$

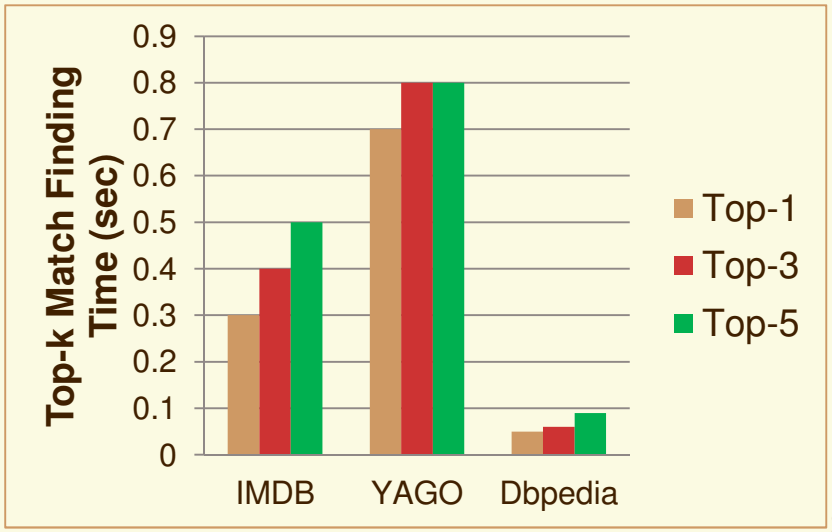
Effectiveness & Efficiency

	BLINKS [SIGMOD '07]	SAGA [Bioinfo '06]	IsoRank [PNAS '08]	gStore [VLDB '11]	NeMa [VLDB '13]
Precision	0.52	0.75	0.63	0.59	**0.91**
Recall	0.52	0.75	0.63	0.59	**0.91**
Time (top-1)	1.92 sec	15.95 sec	4882.0 sec	0.92 sec	**0.97 sec**

IMDB - 3M, 11M; YAGO - 13M, 18M; DBPedia – 5M, 20M



Query graph contains 7 nodes, structural noise 30%, and label noise 50%.



Towards Advanced Search in Complex Graphs:

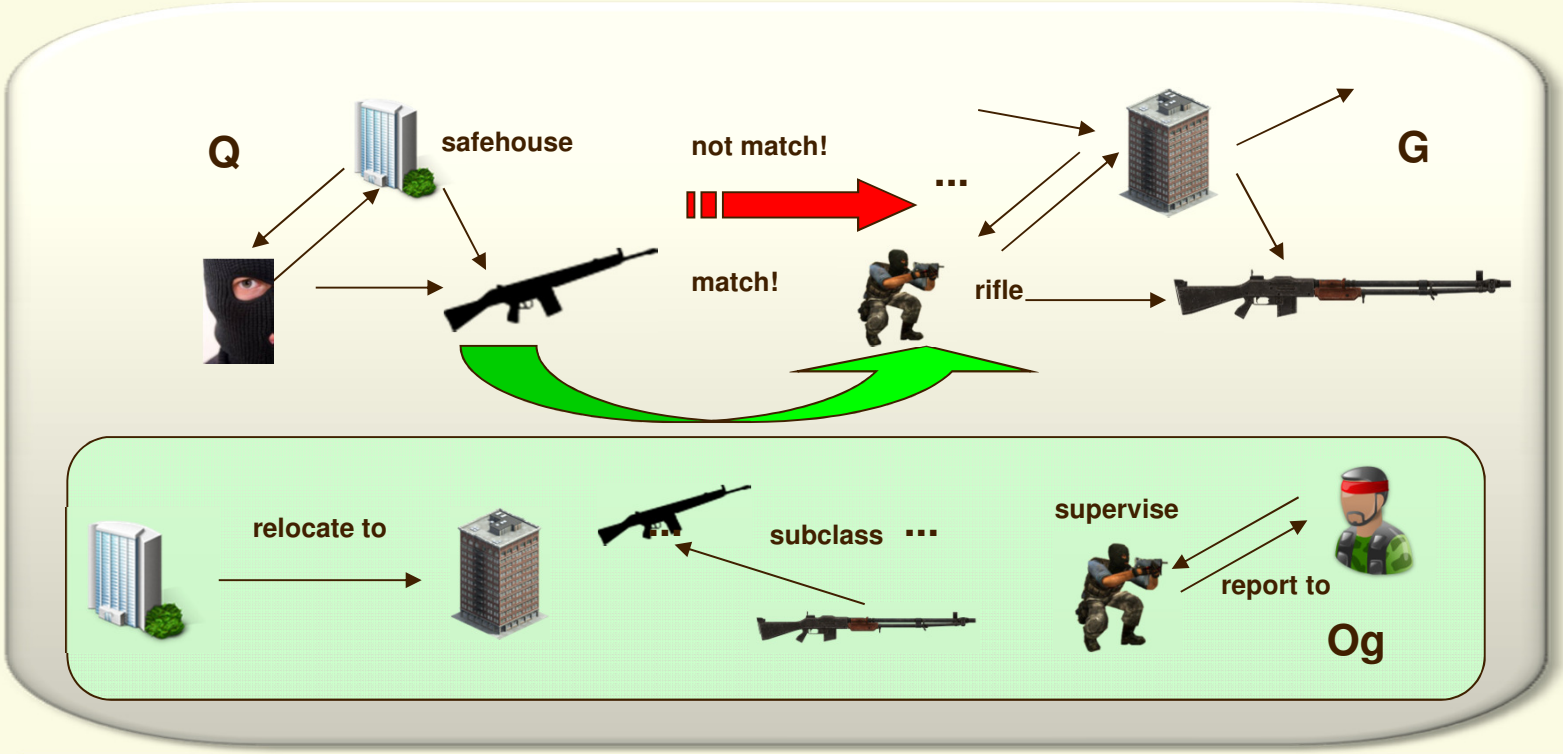
querying noisy graphs

Searching in noisy graphs (ICDE '13)

- Finding semantically related matches for Q in G

Q: "*find suspect A using class I guns in a safehouse*"

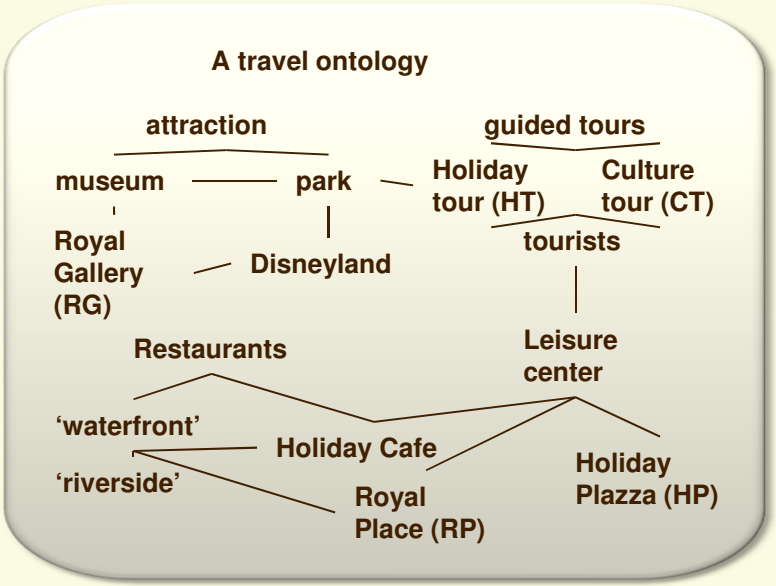
A:"we found a match [who report to A] using class II guns [a type of class I guns] in a [relocated] safehouse"



Using ontology-information to capture semantically similar matches

Ontology-based searching

- Ontology graphs

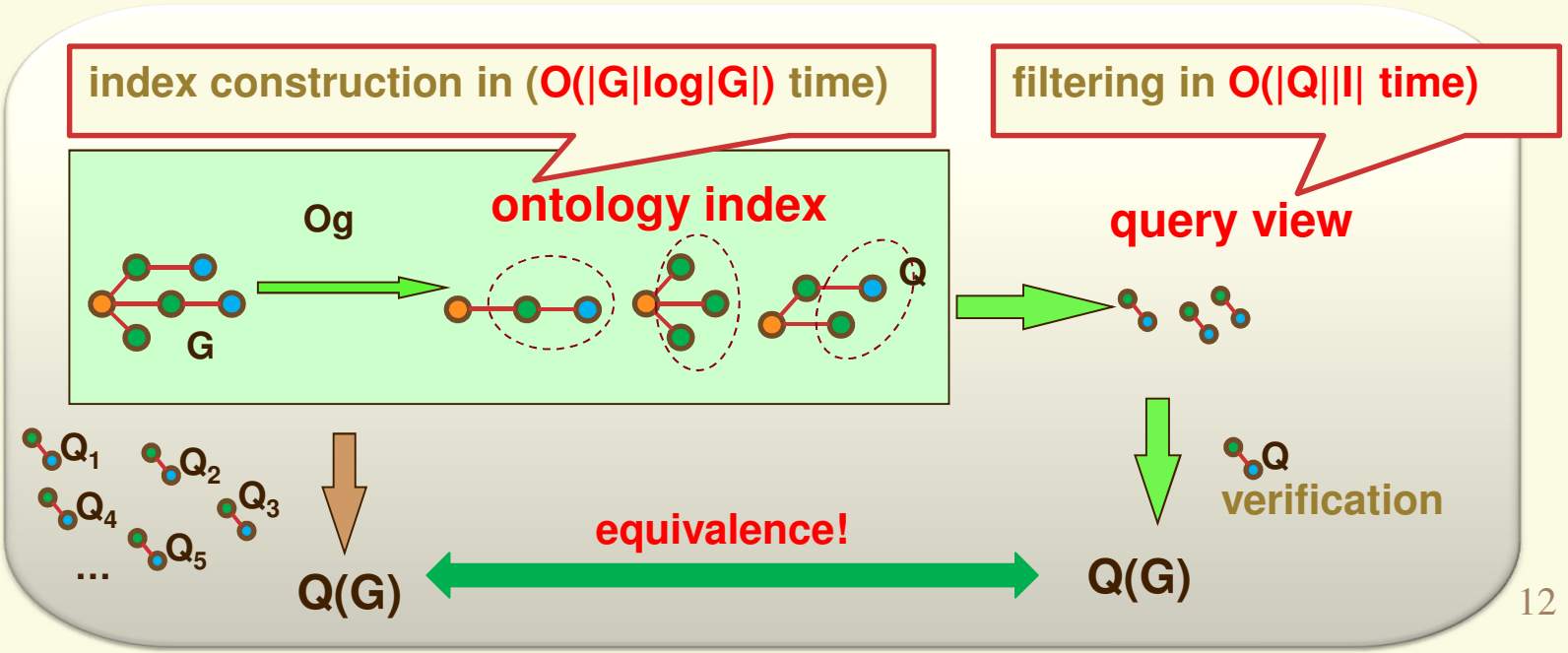


- **Ontology-based subgraph querying**: given a data graph G, a query graph Q and an ontology graph Og, identify best matches that are semantically close to Q

Querying framework

- A querying framework based on filtering-and-verification
  - (1) offline ontology indexing: develop "ontology views" of G as an ontology index, by summarizing G using Og
  - (2) online ontology-based filtering-and-verification

A query evaluation framework (comparing with query enumeration):



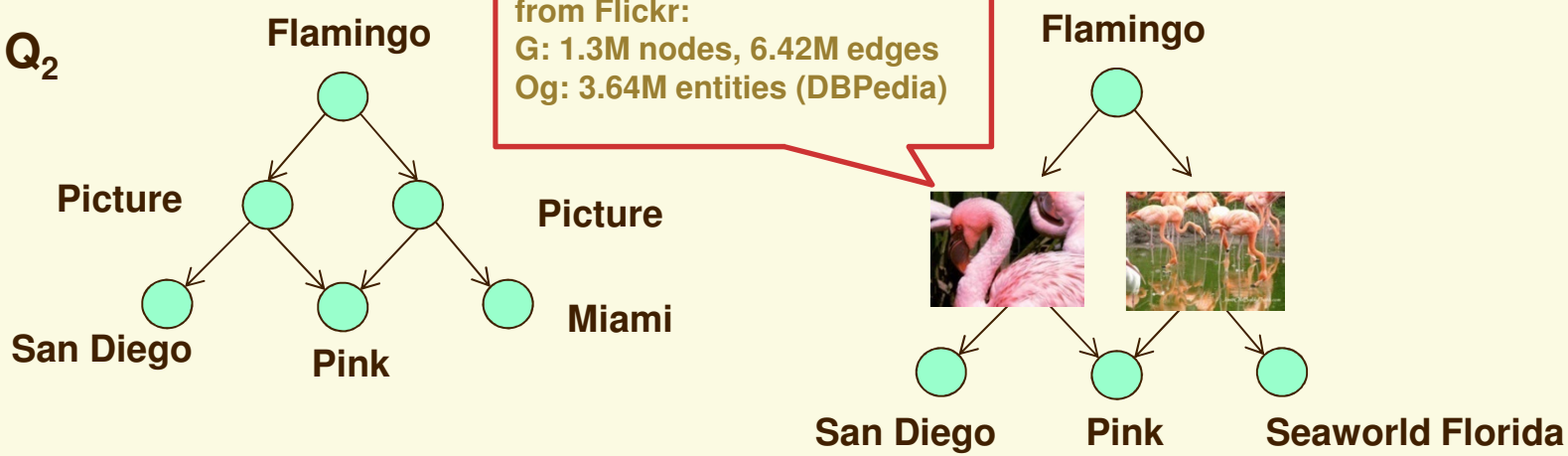
Experimental results: effectiveness

from CrossDomain:
G: 1.07M nodes, 3.86M edges
Og: 1.44M nodes, 5.3M edges

Q_1

from Flickr:
G: 1.3M nodes, 6.42M edges
Og: 3.64M entities (DBPedia)

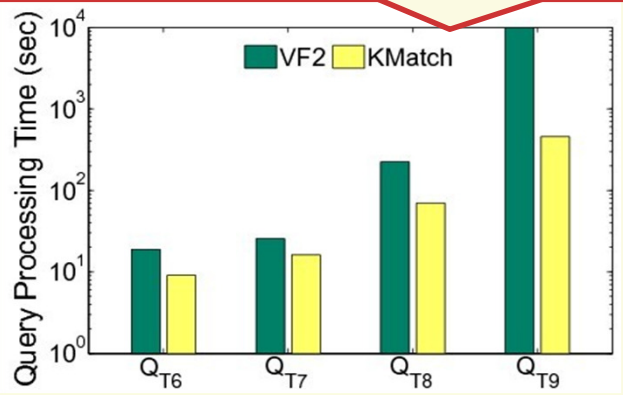
Q_2



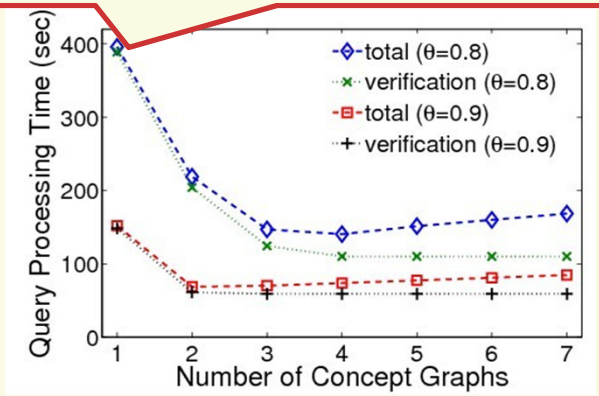
Ontology matching identifies much more meaningful "hidden" matches

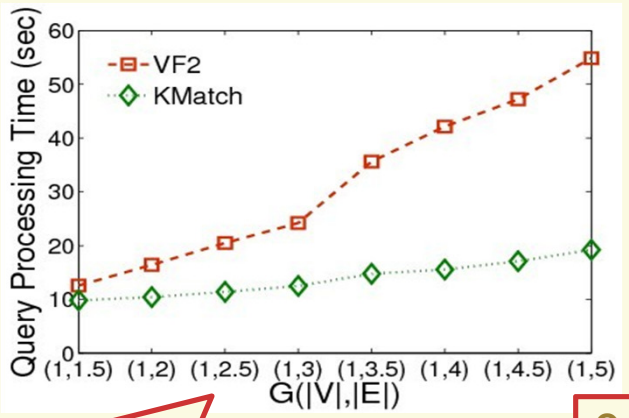
Experimental results: efficiency

30% of the running time of traditional subgraph querying algorithm, e.g., VF2

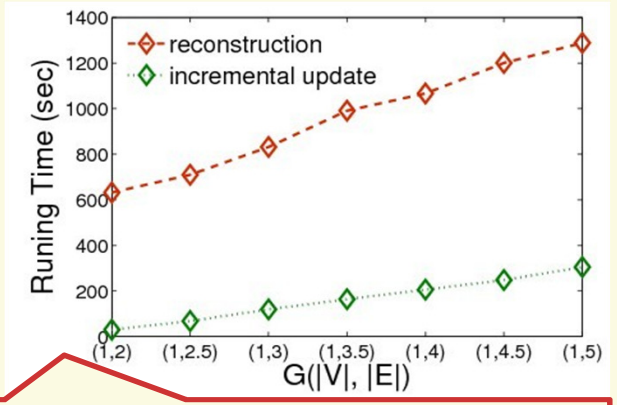


Effective even with a single concept graph





Scale well with data size



Ontology index can be efficiently updated upon changes to data graphs

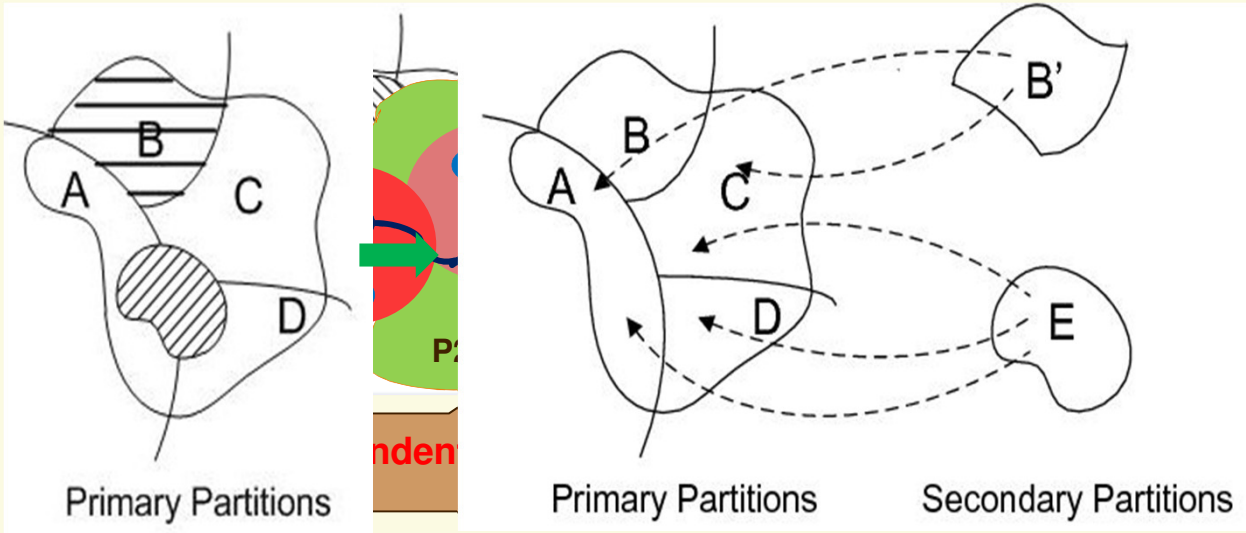
Ontology matching outperforms traditional graph querying in efficiency

Towards Advanced Search in Complex Graphs:

partition strategy and management

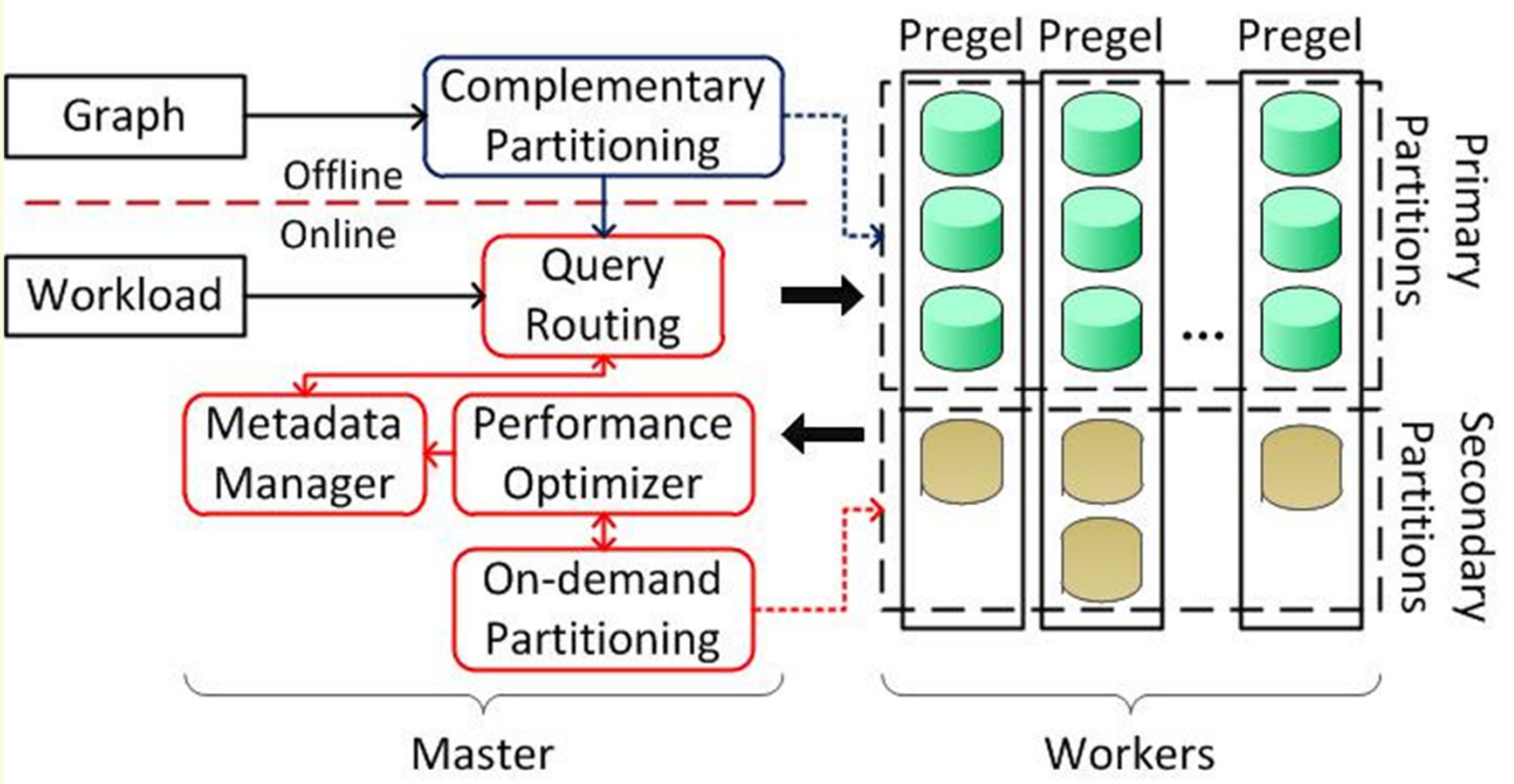
Graph partition and management(SIGMOD '12)

- ✓ Sedge: partition strategy
  - Complementary partitioning
    - **repartition the graph with region constraint**
  - On-demand partitioning: "envelop" grouping for cross queries
  - Two-level partition management

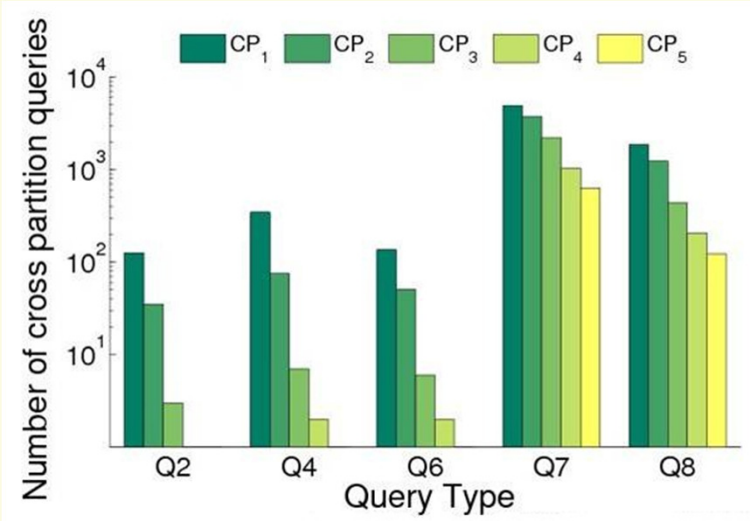


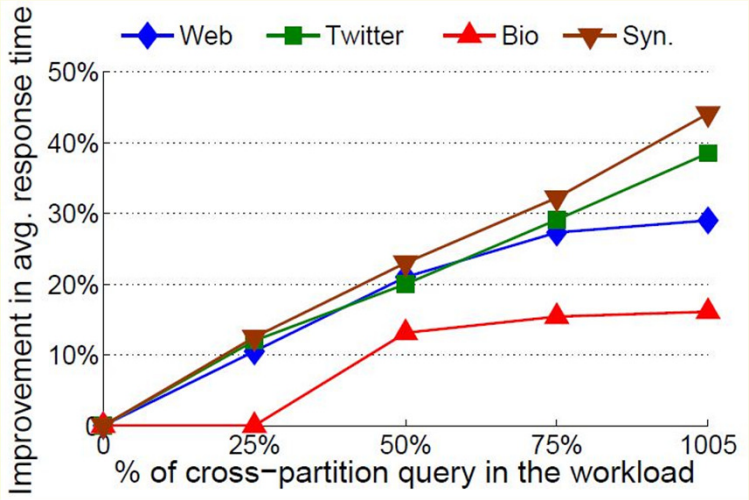
Graph partition management

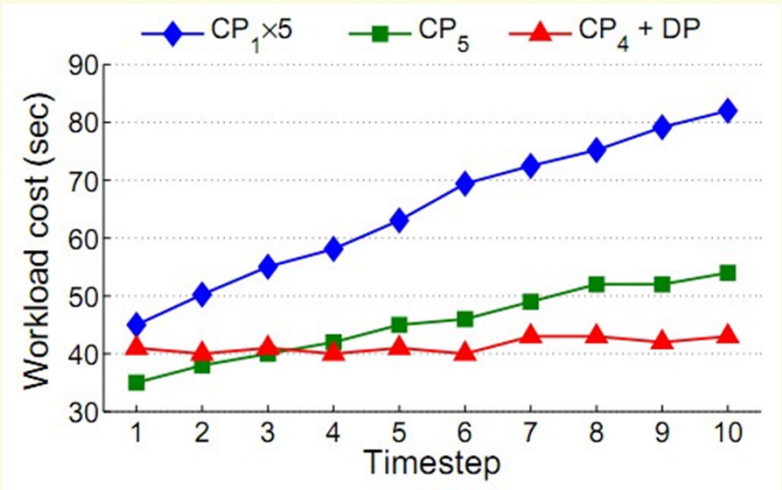
- Sedge: system architecture



Sedge: performance evaluation







- **SP²Bench (DBLP-based)**
- **Significant cross query reduction (complete removal of cross queries for several cases)**
- **Complementary and on demand partition is effective**
- **effectively improve efficiency for cross queries**

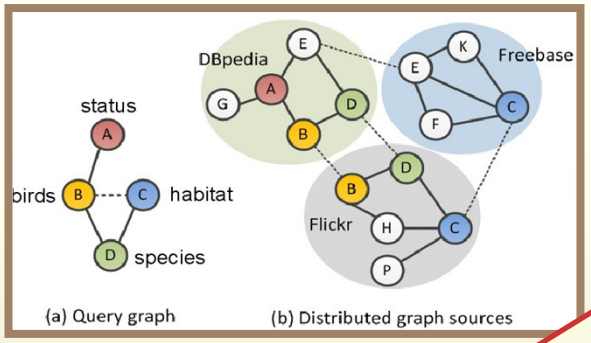
Towards Advanced Search in Big Graphs:

dealing with decentralized world

Dealing with decentralized world

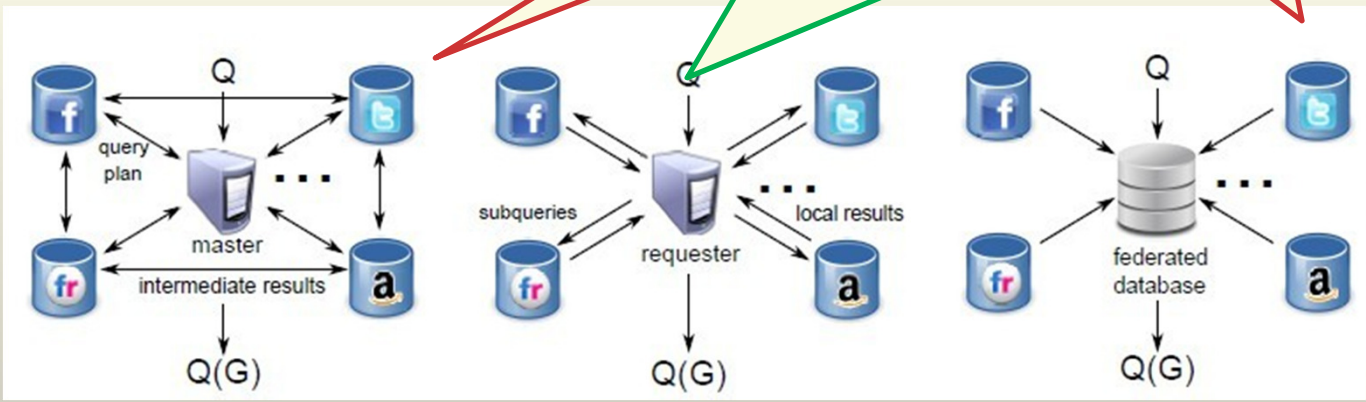
- ✓ Real-life graphs are distributed

Our vision: a dispatching-local evaluation-assembling framework



High communication cost; inter-source communication is expensive/not allowed

Local evaluation+assembling



Distributed querying over independent graph sources

Distributed graph querying framework (rejected, still trying)

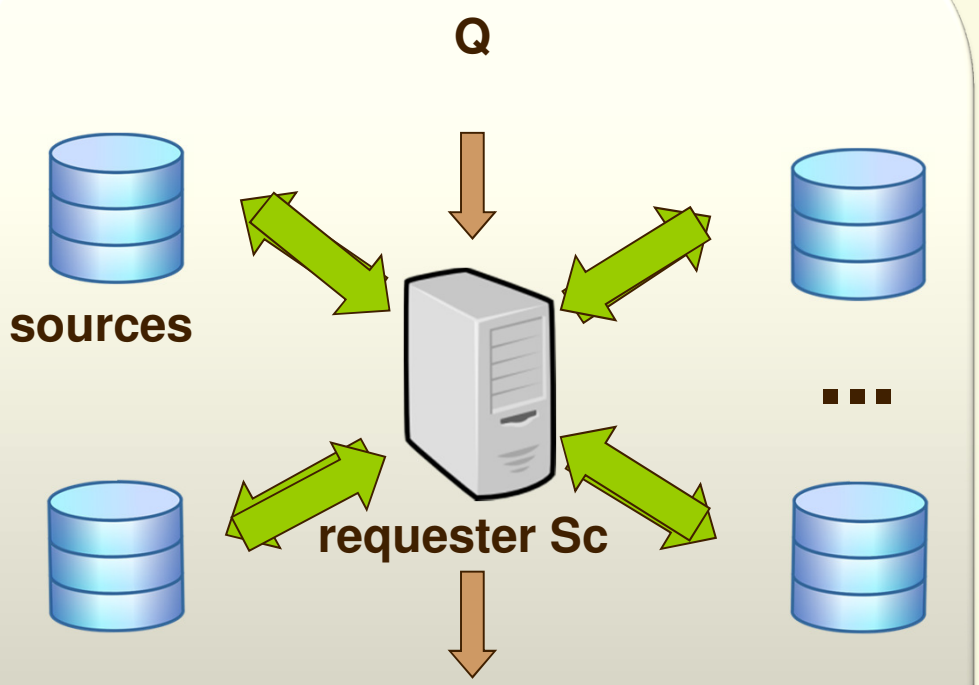
Requester site Sc

and a set of graph Sources F1, …, Fn

distributing at Sc: source filtering and query decomposition

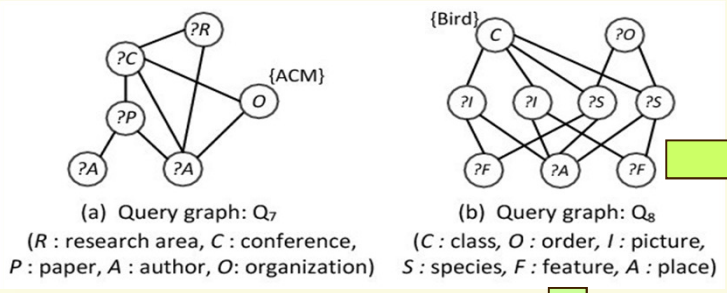
local evaluation: generate partial results

Assembling at Sc



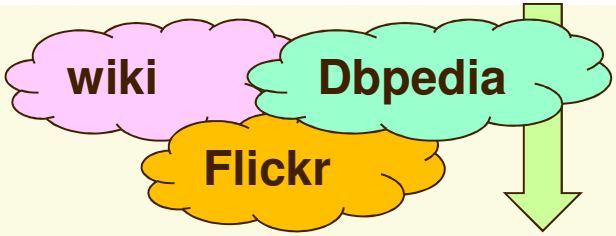
dispatching + local evaluation + assembling

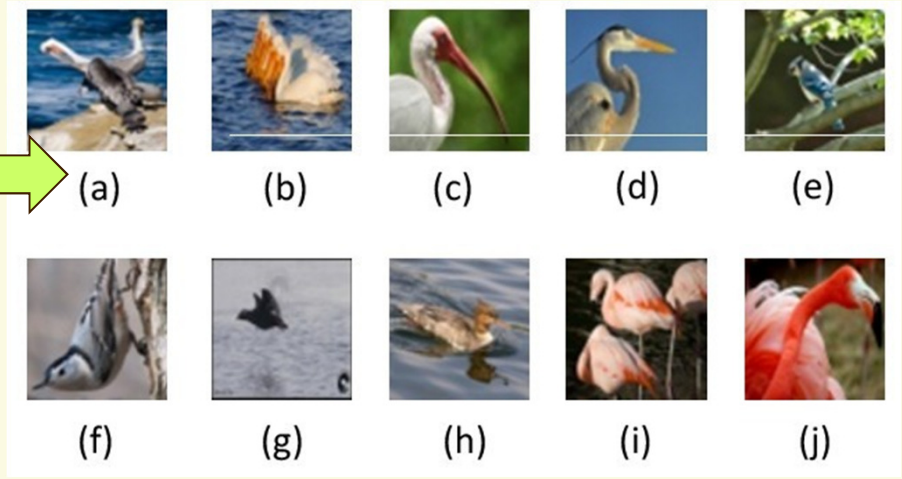
Distributed graph querying: real-life queries



(a) Query graph: Q_7 (*R* : research area, *C* : conference, *P* : paper, *A* : author, *O*: organization)

(b) Query graph: Q_8 (*C* : class, *O* : order, *I* : picture, *S* : species, *F* : feature, *A* : place)

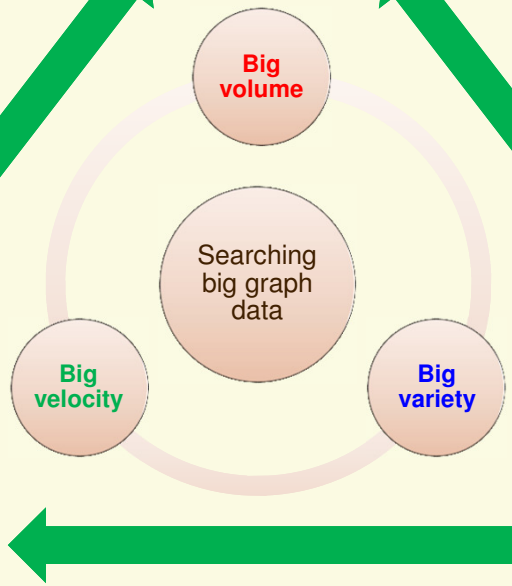




$?I_1$(picture)	$?S_1$(species)	$?F_1$(feature)	$?I_2$(picture)	$?S_2$(species)	$?F_2$(feature)	$?A$(place)
(a)	Brown Pelican	Brown body	(b)	American White Pelican	White Plumage	Pacific Coast
(c)	American White Ibis	Red-billed	(d)	Great Blue Heron	Blue Features	Florida Coast
(e)	Blue Jay	Blue crest	(f)	White-Breasted Nuthatch	White-breasted	Woodland
(g)	Black Scoter	All Black	(h)	Red-breasted Merganser	Red-breasted	Water Coast
(i)	Chilean Flamingo	Pink Plumage	(j)	American Flamingo	Red-necked	California Bay

Searching complex graph: a “big graph” issue

• computationally efficient query models
• partition strategy & management/ distributed querying
• compression/summarization
• view-based querying
•…



•incremental/dynamic graph querying and maintenance
•Spatial-temporal /stream graph querying
•…

• semantic searching e.g., ontology-based indexing and querying
• usability-expressive power: query suggestion/transformation/rewriting/refinement
• knowledge inference
•…

A great source of research topics and promising search tools

resources

- All of our software and data will be announced in this link: http://grafia.cs.ucsb.edu/
- Ness and Nema: source code
  - http://habitus.cs.ucsb.edu/SIGMOD11_Ness.tar.gz
  - http://habitus.cs.ucsb.edu/VLDB13_NeMa.tar.gz
- Sedge: project homepage (docs, source code and dataset)
  - http://grafia.cs.ucsb.edu/sedge/
- Ontology-based subgraph matching
  - http://grafia.cs.ucsb.edu/ontology

Thank you!

- Acknowledgement:
  - Information Network Science CTA
  - Our group: Xifeng Yan, Shengqi, Arijit …