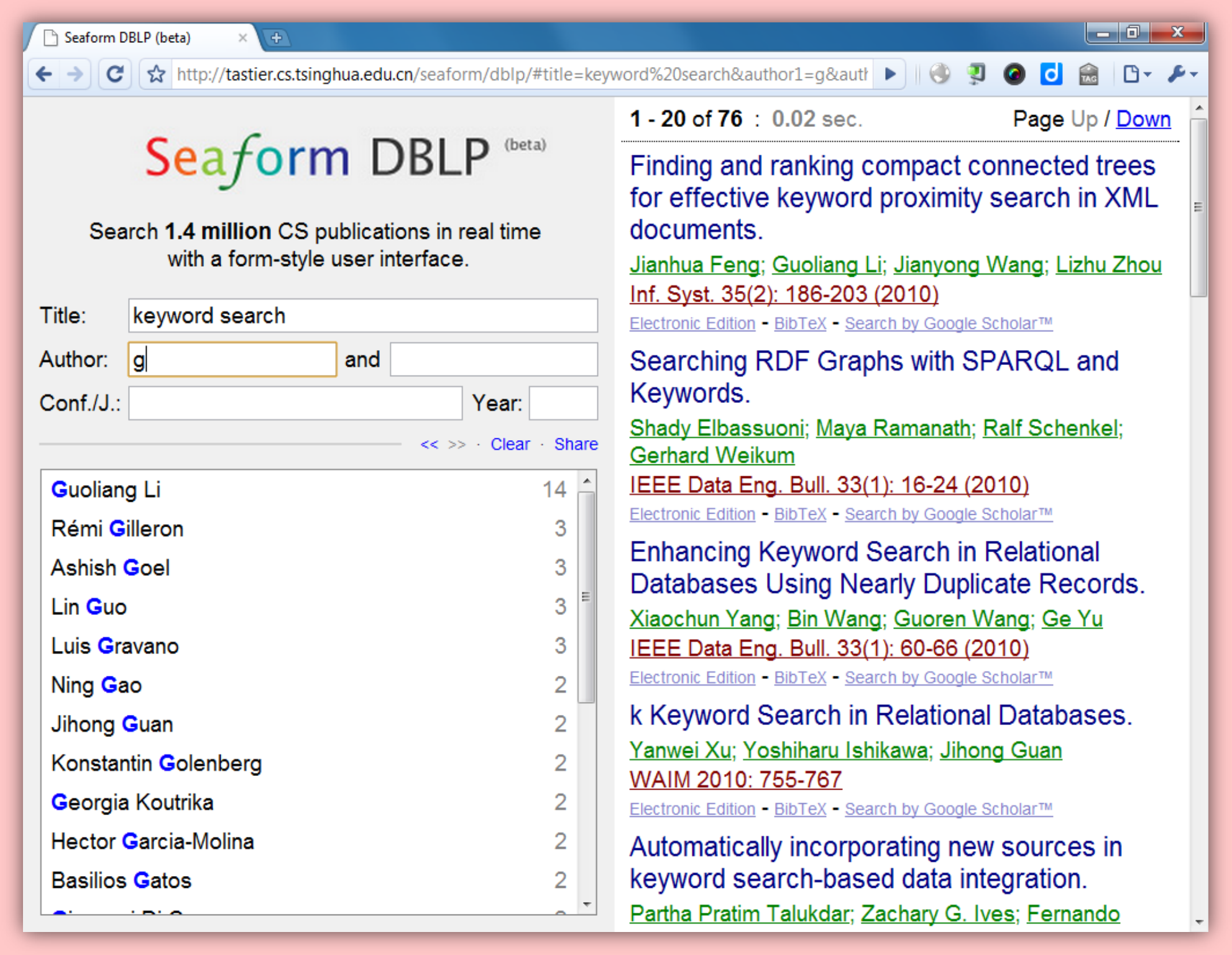
# Seaform: Search-As-You-Type in Forms

Hao Wu[1], Guoliang Li[1], Chen Li[2], Lizhu Zhou[1]
[1]Department of Computer Science, Tsinghua University, Beijing, China
[2]Department of Computer Science, University of California, Irvine, USA

## What is Seaform?

Seaform is a keyword search system that can search a relational table in real time using a form-style user interface.



Seaform DBLP (beta) — http://tastier.cs.tsinghua.edu.cn/seaform/dblp/#title=keyword%20search&author1=g&autl

Seaform DBLP (beta)

Search **1.4 million** CS publications in real time with a form-style user interface.

Title: keyword search
Author: g and
Conf./J.: Year:

<< >> · Clear · Share

| | |
|---|---|
| Guoliang Li | 14 |
| Rémi Gilleron | 3 |
| Ashish Goel | 3 |
| Lin Guo | 3 |
| Luis Gravano | 3 |
| Ning Gao | 2 |
| Jihong Guan | 2 |
| Konstantin Golenberg | 2 |
| Georgia Koutrika | 2 |
| Hector Garcia-Molina | 2 |
| Basilios Gatos | 2 |

**1 - 20 of 76** : 0.02 sec. Page Up / Down

Finding and ranking compact connected trees for effective keyword proximity search in XML documents.
Jianhua Feng; Guoliang Li; Jianyong Wang; Lizhu Zhou
Inf. Syst. 35(2): 186-203 (2010)
Electronic Edition - BibTeX - Search by Google Scholar™

Searching RDF Graphs with SPARQL and Keywords.
Shady Elbassuoni; Maya Ramanath; Ralf Schenkel; Gerhard Weikum
IEEE Data Eng. Bull. 33(1): 16-24 (2010)
Electronic Edition - BibTeX - Search by Google Scholar™

Enhancing Keyword Search in Relational Databases Using Nearly Duplicate Records.
Xiaochun Yang; Bin Wang; Guoren Wang; Ge Yu
IEEE Data Eng. Bull. 33(1): 60-66 (2010)
Electronic Edition - BibTeX - Search by Google Scholar™

k Keyword Search in Relational Databases.
Yanwei Xu; Yoshiharu Ishikawa; Jihong Guan
WAIM 2010: 755-767
Electronic Edition - BibTeX - Search by Google Scholar™

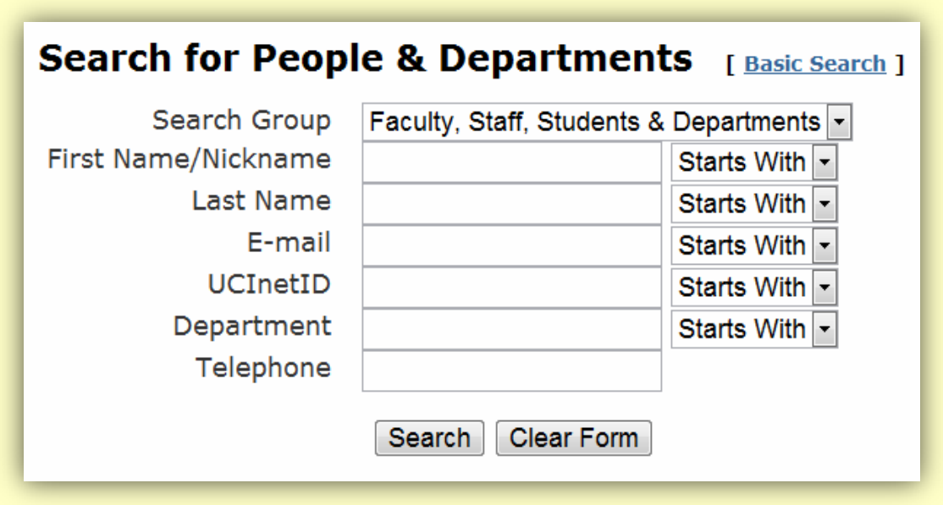
Automatically incorporating new sources in keyword search-based data integration.
Partha Pratim Talukdar; Zachary G. Ives; Fernando

## Why Seaform?

#1: SQL is complex.

```
SELECT *
FROM   Author A, Autor_Paper AP, Paper P
WHERE  title   LIKE 'keyword' AND
       title   LIKE 'search'  AND
       authors LIKE 'g%'      AND
       A.id    =    AP.aid    AND
       P.id    =    AP.pid
```

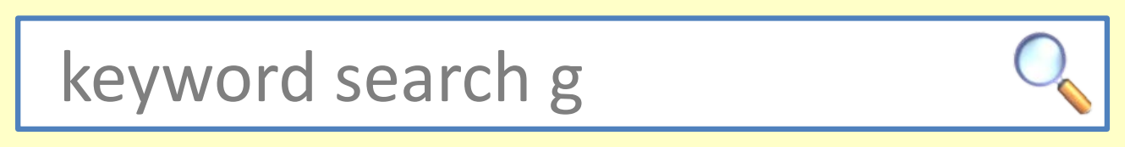
#2: Form is awkward.



**Search for People & Departments** [ Basic Search ]

Search Group: Faculty, Staff, Students & Departments
First Name/Nickname: Starts With
Last Name: Starts With
E-mail: Starts With
UCInetID: Starts With
Department: Starts With
Telephone:

#3: Traditional keyword search is imprecise.



keyword search g

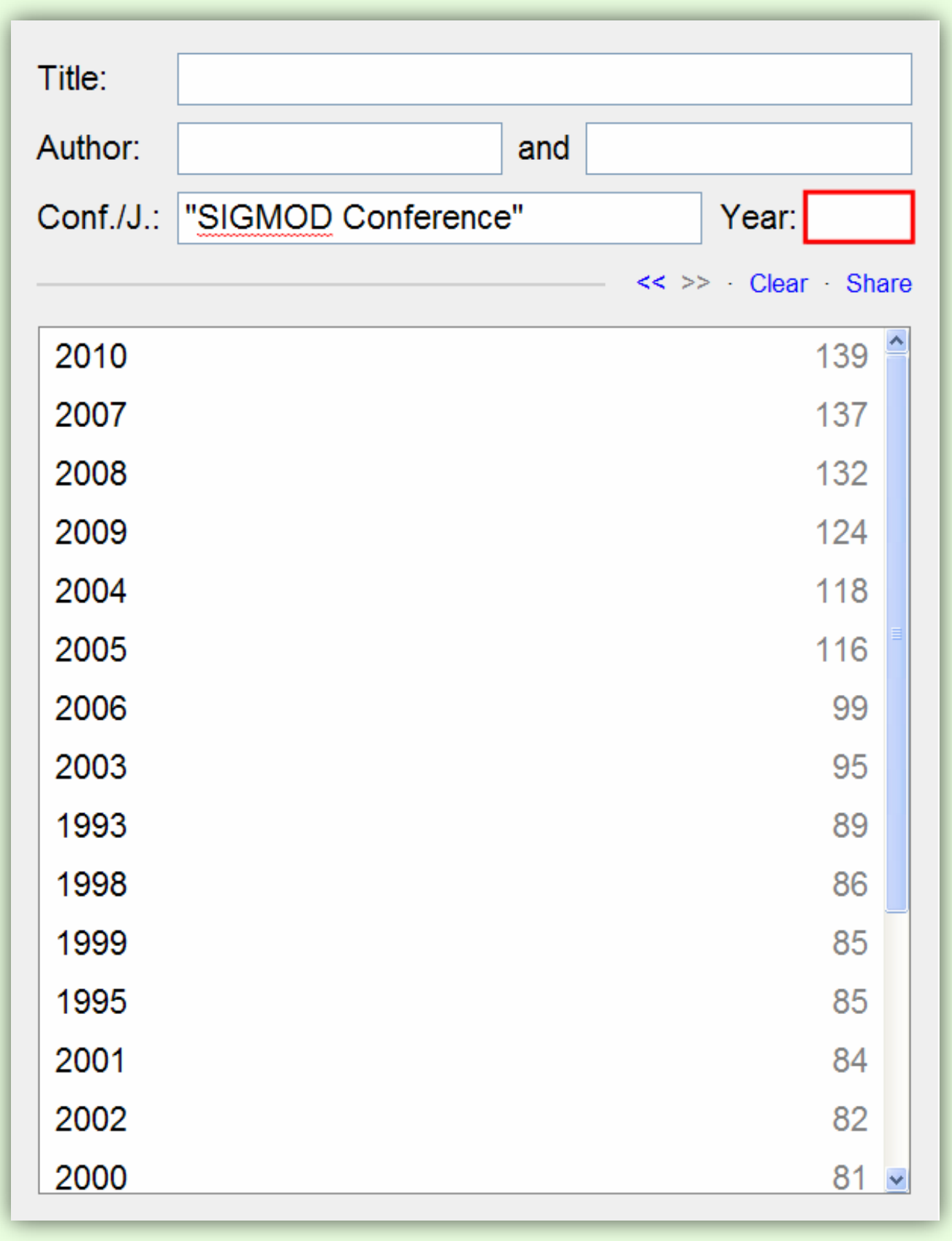
Title? Conf. name?

## Features

#1: Real-time response: less than 30 ms. for each keystroke in average.

#2: Precise search conditions.

#3: Faceted search.

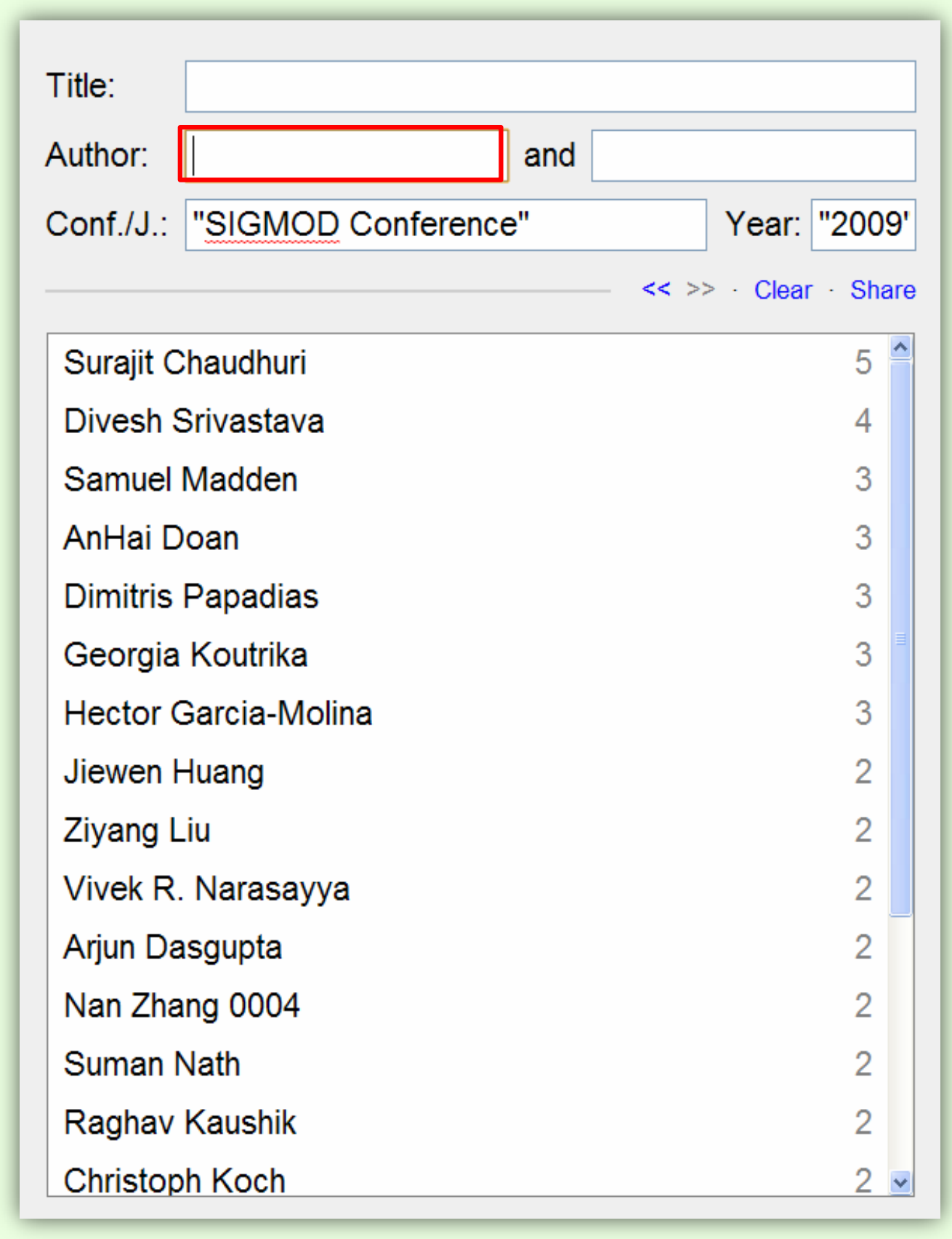


Title:
Author: and
Conf./J.: "SIGMOD Conference" Year:

<< >> · Clear · Share

| | |
|---|---|
| 2010 | 139 |
| 2007 | 137 |
| 2008 | 132 |
| 2009 | 124 |
| 2004 | 118 |
| 2005 | 116 |
| 2006 | 99 |
| 2003 | 95 |
| 1993 | 89 |
| 1998 | 86 |
| 1999 | 85 |
| 1995 | 85 |
| 2001 | 84 |
| 2002 | 82 |
| 2000 | 81 |

# of accepted papers over years.

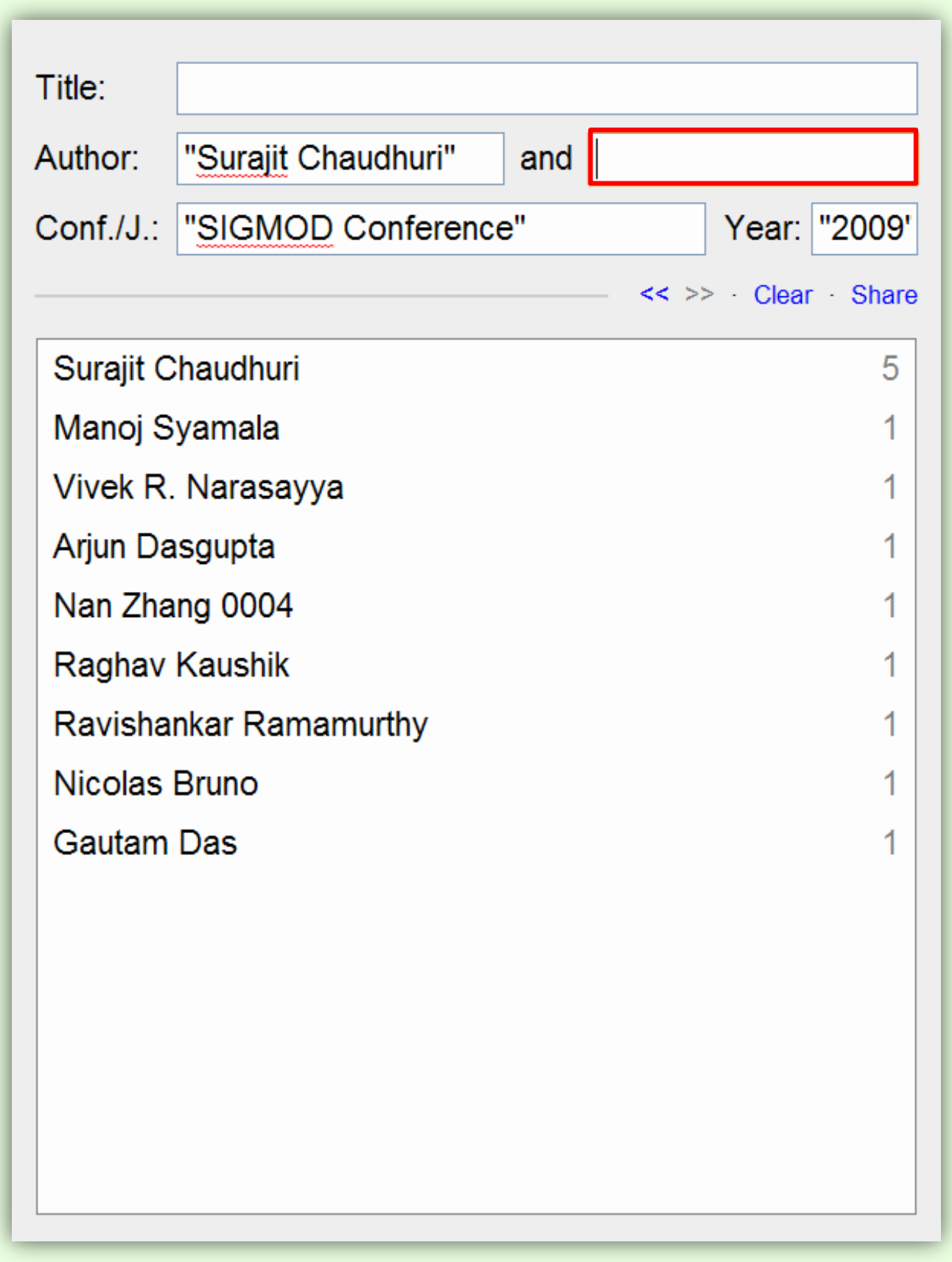


Title:
Author: and
Conf./J.: "SIGMOD Conference" Year: "2009'

<< >> · Clear · Share

| | |
|---|---|
| Surajit Chaudhuri | 5 |
| Divesh Srivastava | 4 |
| Samuel Madden | 3 |
| AnHai Doan | 3 |
| Dimitris Papadias | 3 |
| Georgia Koutrika | 3 |
| Hector Garcia-Molina | 3 |
| Jiewen Huang | 2 |
| Ziyang Liu | 2 |
| Vivek R. Narasayya | 2 |
| Arjun Dasgupta | 2 |
| Nan Zhang 0004 | 2 |
| Suman Nath | 2 |
| Raghav Kaushik | 2 |
| Christoph Koch | 2 |

All authors sorted by # of papers.



Title:
Author: "Surajit Chaudhuri" and
Conf./J.: "SIGMOD Conference" Year: "2009'

<< >> · Clear · Share

| | |
|---|---|
| Surajit Chaudhuri | 5 |
| Manoj Syamala | 1 |
| Vivek R. Narasayya | 1 |
| Arjun Dasgupta | 1 |
| Nan Zhang 0004 | 1 |
| Raghav Kaushik | 1 |
| Ravishankar Ramamurthy | 1 |
| Nicolas Bruno | 1 |
| Gautam Das | 1 |

Coauthors of Surajit Chaudhuri.

## System Overview

### Data:

| ID | Title | Conf. | Author |
|---|---|---|---|
| 1 | xml database | VLDB | albert |
| 2 | xml database | SIGMOD | bob |
| 3 | xml search | VLDB | albert |
| 4 | xml security | VLDB | alice |
| 5 | rdbms | SIGMOD | charlie |

Global table

| ID | Title |
|---|---|
| T1 | xml database |
| T2 | xml search |
| T3 | xml security |
| T4 | rdbms |

| ID | Conf. |
|---|---|
| C1 | VLDB |
| C2 | SIGMOD |

| ID | Author |
|---|---|
| A1 | albert |
| A2 | bob |
| A3 | alice |
| A4 | charlie |

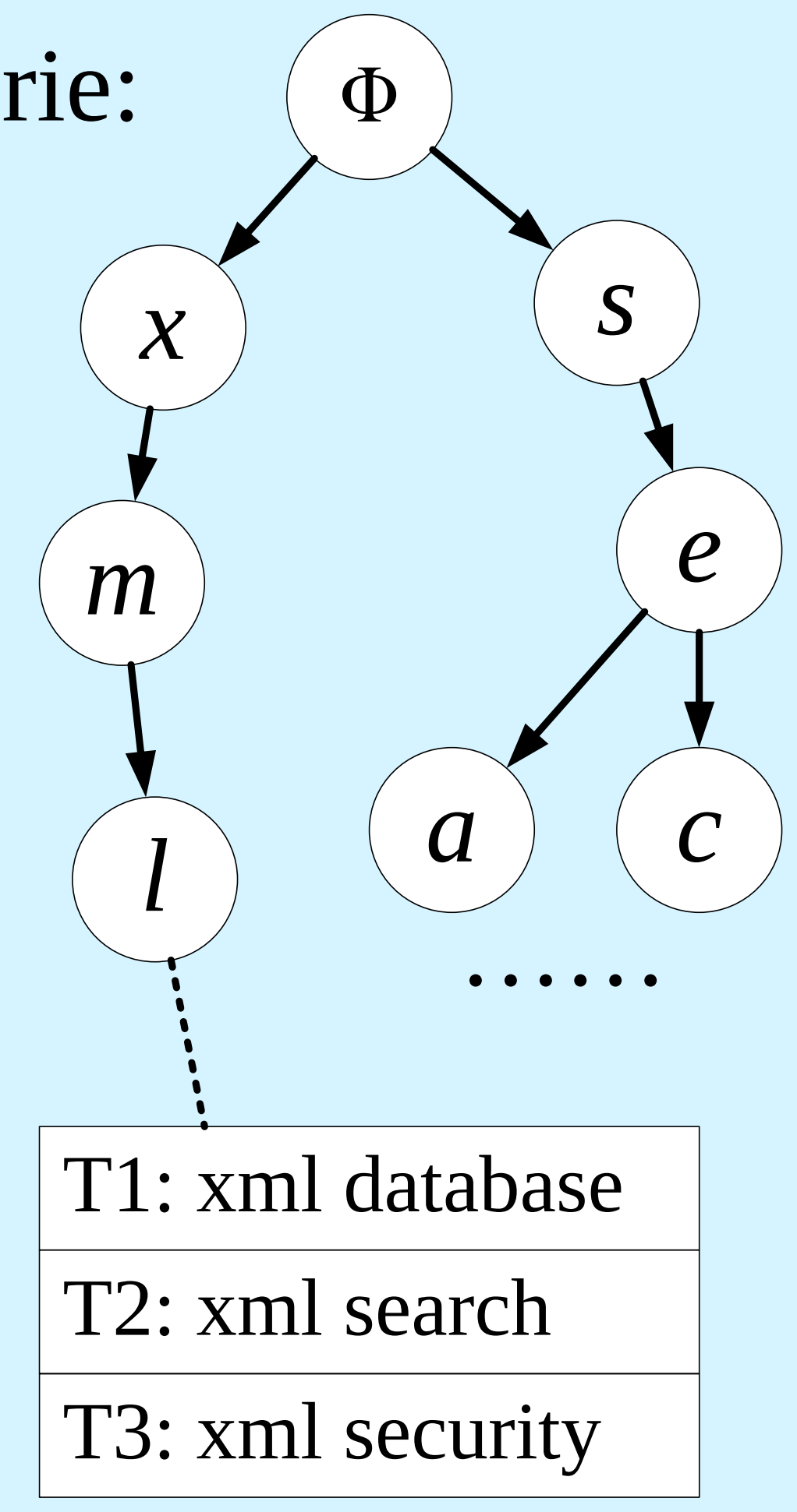
Local tables

### Index:

For each local table, we build:

1. A trie structure;
2. A local-global mapping table; and
3. A global-local mapping table.

Trie:



| |
|---|
| T1: xml database |
| T2: xml search |
| T3: xml security |

L-G Mapping Table:

| | |
|---|---|
| T1 | 1, 2 |
| T2 | 3 |
| T3 | 4 |
| T4 | 5 |

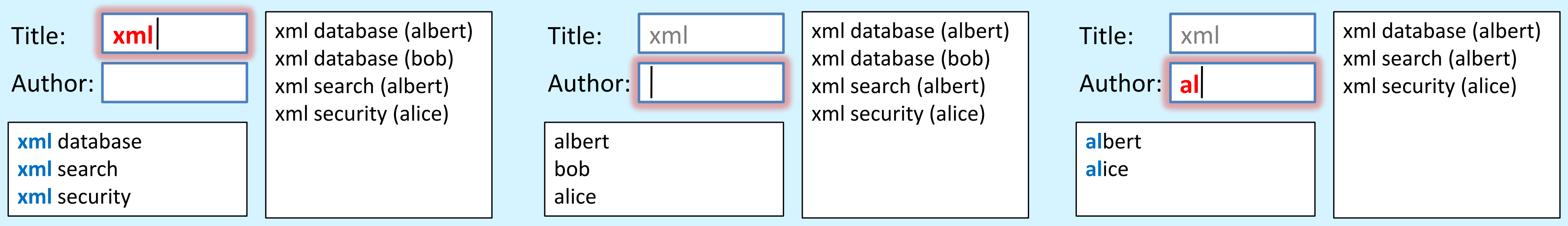
G-L Mapping Table:

| | |
|---|---|
| 1 | T1 |
| 2 | T1 |
| 3 | T2 |
| 4 | T3 |
| 5 | T4 |

### Search Algorithm:

1. Identify $\Delta q = q - q_0$ using cache;
2. Calculate $L_q^* = \mathrm{TrieSearch}(\Delta q) \cap L_{q_0}$ ;
3. Calculate $G_q = \mathrm{MapToGlobal}(L_q^*) \cap G_{q_0}$ ;
4. Calculate $L_q = \mathrm{MapToLocal}(G_q) \cap L_{q_0}$ .

*Synchronization*



Title: xml
Author:

xml database
xml search
xml security

xml database (albert)
xml database (bob)
xml search (albert)
xml security (alice)

Title: xml
Author:

albert
bob
alice

xml database (albert)
xml database (bob)
xml search (albert)
xml security (alice)

Title: xml
Author: al

albert
alice

xml database (albert)
xml search (albert)
xml security (alice)

$\Delta q$ = title:xml

$L_q^*$ = {xml database, xml search, xml security}

$G_q$ = {xml database (albert), xml database (bob), xml search (albert), xml security (alice)}

$L_q$ = {xml database, xml search, xml security}

$\Delta q$ = author:

$L_q^*$ = {albert, bob, alice, charlie}

$G_q$ = {xml database (albert), xml database (bob), xml search (albert), xml security (alice)}

$L_q$ = {albert, bob, alice}

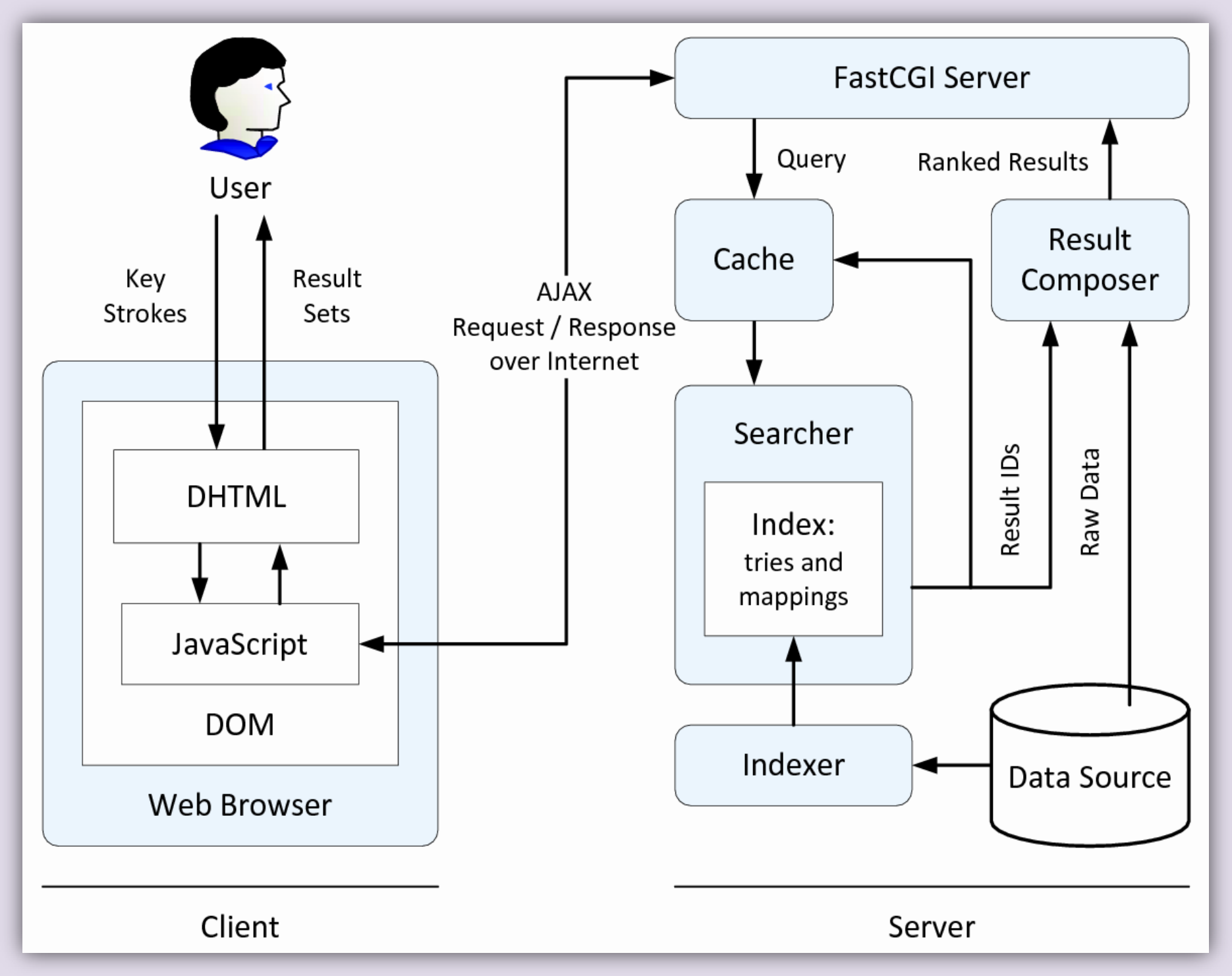
$\Delta q$ = author:al

$L_q^*$ = {albert, alice}

$G_q$ = {xml database (albert), xml search (albert), xml security (alice)}

$L_q$ = {albert, alice}

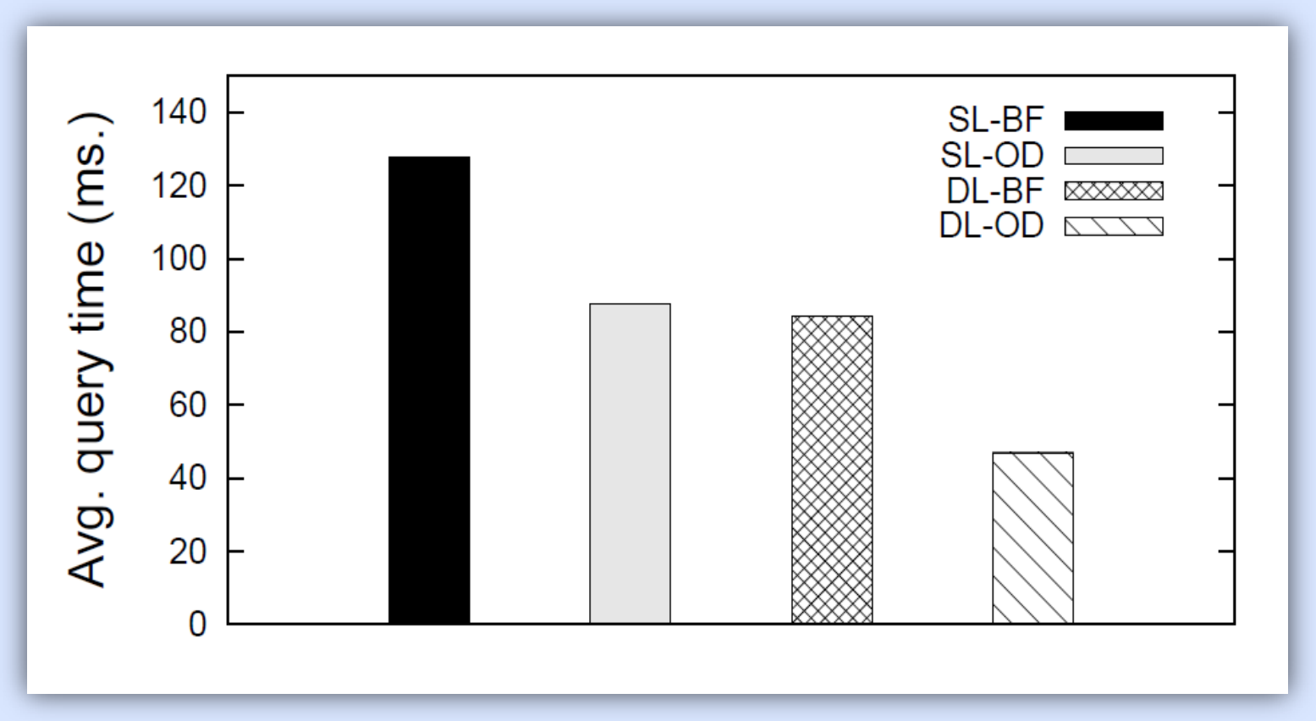
## Implementation



- Client/server architecture:
  1) Client: Javascript/Ajax,
  2) Server: FastCGI.
- Datasets:
  1) DBLP (1.4m publications),
  2) IMDB (507k movies).

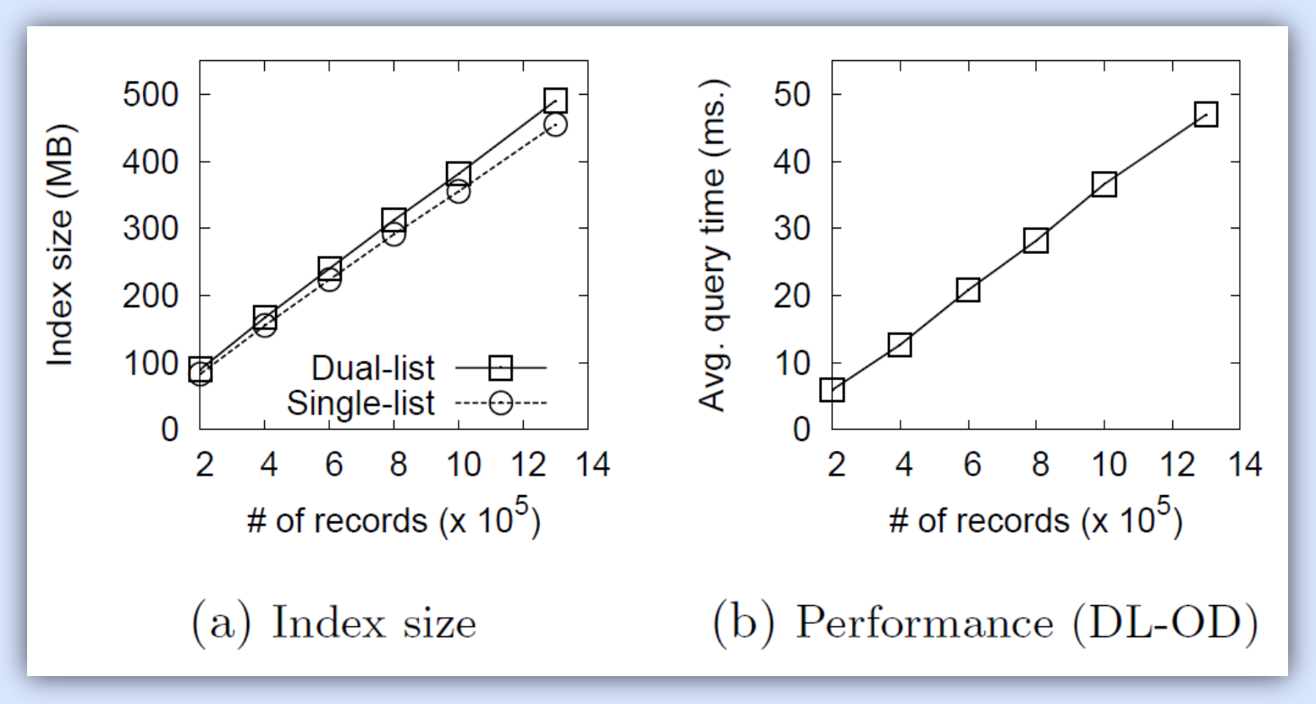
## Experiments

Comparison of four algorithms:



SL: single-list tire, DL: dual-list trie,
BF: brute-force sync., OD: on-demand sync.

Scalability of Seaform-DBLP:



(a) Index size (b) Performance (DL-OD)

http://tastier.cs.thu.edu.cn/seaform/

VLDB Demonstration Track, Singapore, Sept. 2010.