Similarity Group-by Operators for Multi-dimensional Relational Data

Mingjie Tang[1], Ruby Y. Tahboub[1], Walid G. Aref[1], Mikhail J.Atallah[1], Qutaibah M. Malluhi[2],
Mourad Ouzzani[3] , Yasin N. Silva[4]

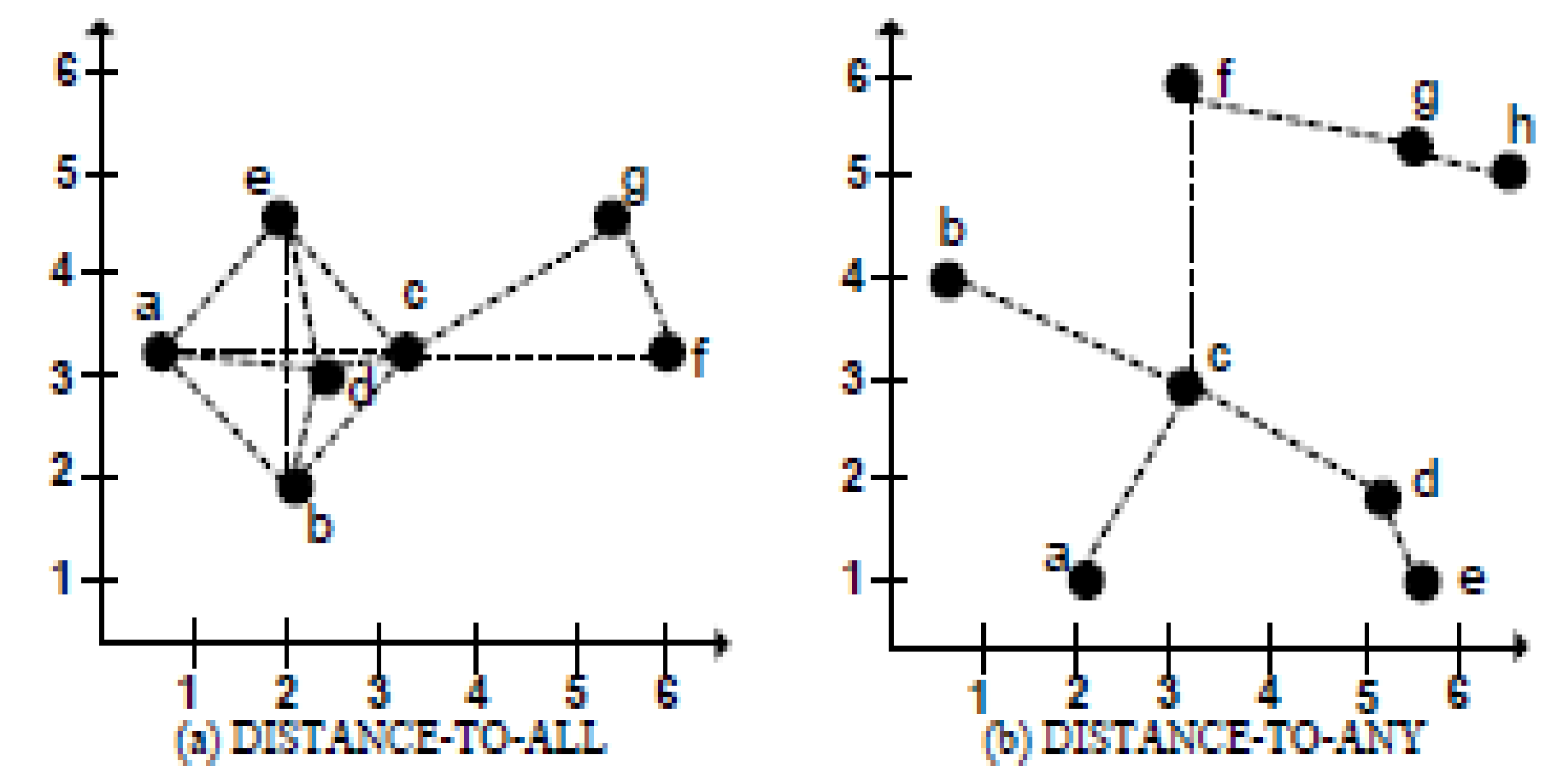
[1]Purdue University, [2]Qatar University, [3]Qatar Computing Research Institute,
[4]Arizona State University

Motivation

- Similarity search is everywhere, so is searching for database elements that are similar or close to a given query element.
- There is a need to group n-dimensional data tuples together that have similar (≈) values.
- We need to **extend the SQL Group-By operator to support similarity-based grouping**.

Semantics of Similarity Group-By (SGB)

- Given 2D data tuples T, and distance parameter ε, return groups of tuples from T that satisfy the predefined distance predicates: *Distance-to-All (SGB-All), Distance-to-Any (SGB-Any)*
- ***Distance-to-All**:* All the tuples in a group are within certain distance threshold ε from each other
- ***Distance-to-Any**:* A tuple belongs to a group if the tuple is within distance ε from any other tuple in the group
- ***ON-OVERLAP***: To decide on a course of action when a point *p is* within Distance ε from more than one group.
- Possible actions:
  - **- ON-OVERLAP JOIN-ANY:** Data point *p* is inserted into any one of the overlapping groups.
  - **- ON-OVERLAP ELIMINATE**: Discard data point *p* if p overlaps more than one group.
  - **- ON-Overlap FORM-NEW-GROUP**: Insert *p* into a separate new group that contain all the overlapping points.

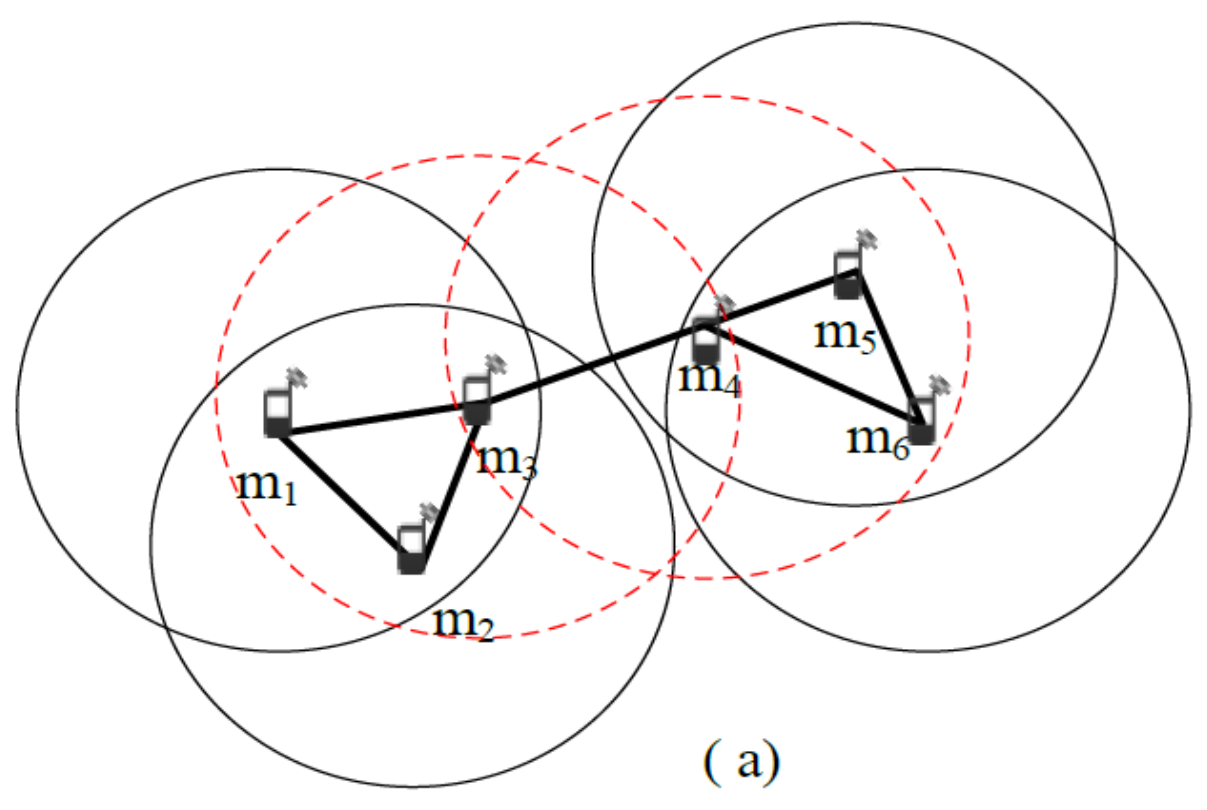


(a) DISTANCE-TO-ALL

(b) DISTANCE-TO-ANY

Example Queries

- Table Mobile Devices: (MDID, Latitude, Longitude) maintains the geographic locations of mobile devices



(a)

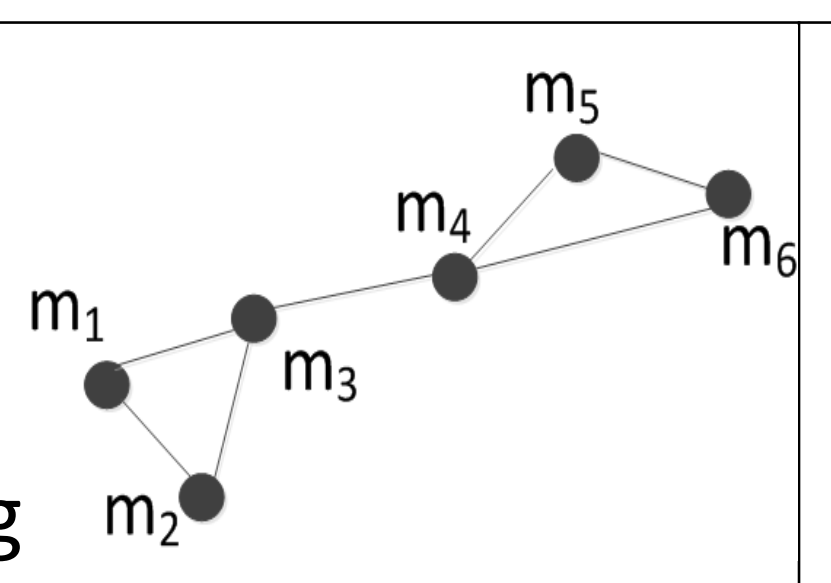
Mobile Devices Table

MDID	Latitude	Longitude	…
m_1	lat_1	lon_1	
m_2	lat_2	lon_2	
…			

(b)

Application of SGB-Any:

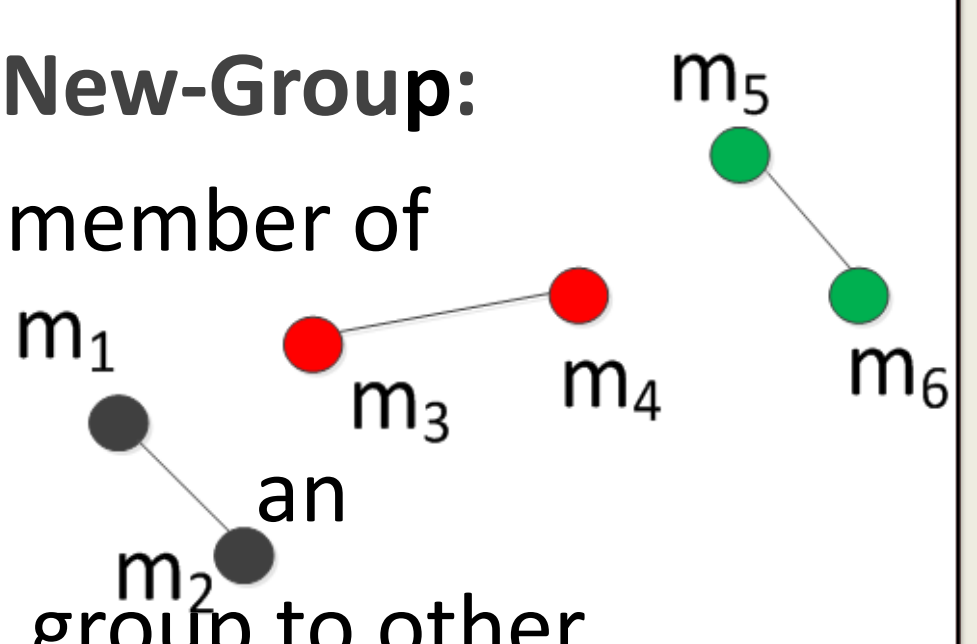
- Identify groups of connected mobile devices using signal range as a similarity grouping threshold



SELECT ST_Polygon (Device-lat, Device-long)
FROM MobileDevices
GROUP BY Device-lat, Device-long
***DISTANCE-TO-ANY L2 WITHIN** SignalRange*

Application of SGB-All-Form-New-Group:

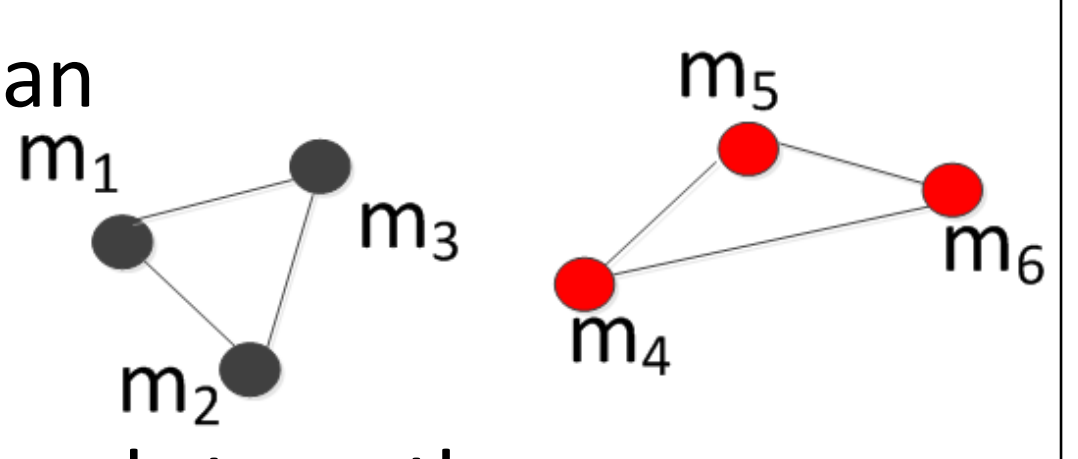
- Identify gateway devices (member of multiple groups)
- A gateway device acts as an entrance from one group to other groups



SELECT List-ID (Device-ID) FROM MobileDevices
GROUP BY Device-lat , Device-long
***DISTANCE-TO-ALL L2 WITHIN** SignalRange*
ON-OVERLAP FORM-NEW-GROUP

Application of SGB-All-Join-Any

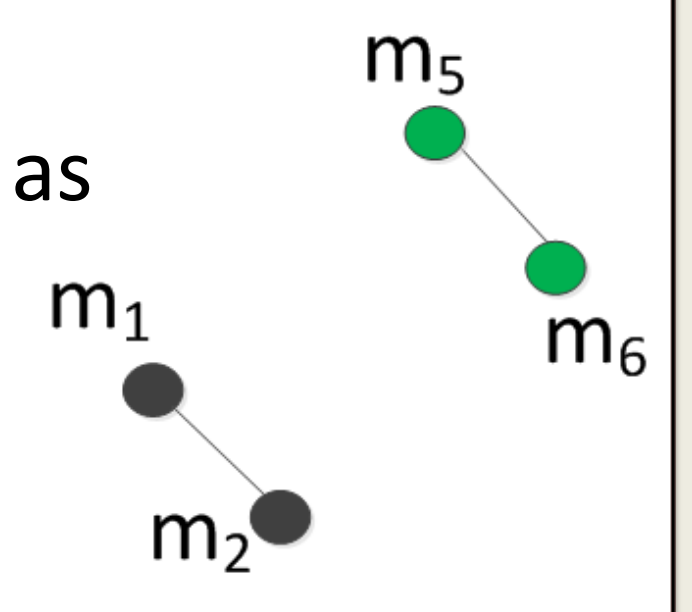
- Identify devices that can communicate with each other directly based on their own signal strength



SELECT List-ID (Device-ID) FROM MobileDevices
GROUP BY Device-lat, Device-long
***DISTANCE-TO-AALL L2 WITHIN** SignalRange*
ON-OVERLAP JOIN-ANY

Application of SGB-All-Eliminate

- Identify devices that cannot serve as a gateway, and devices from different group that cannot communicate without a gateway

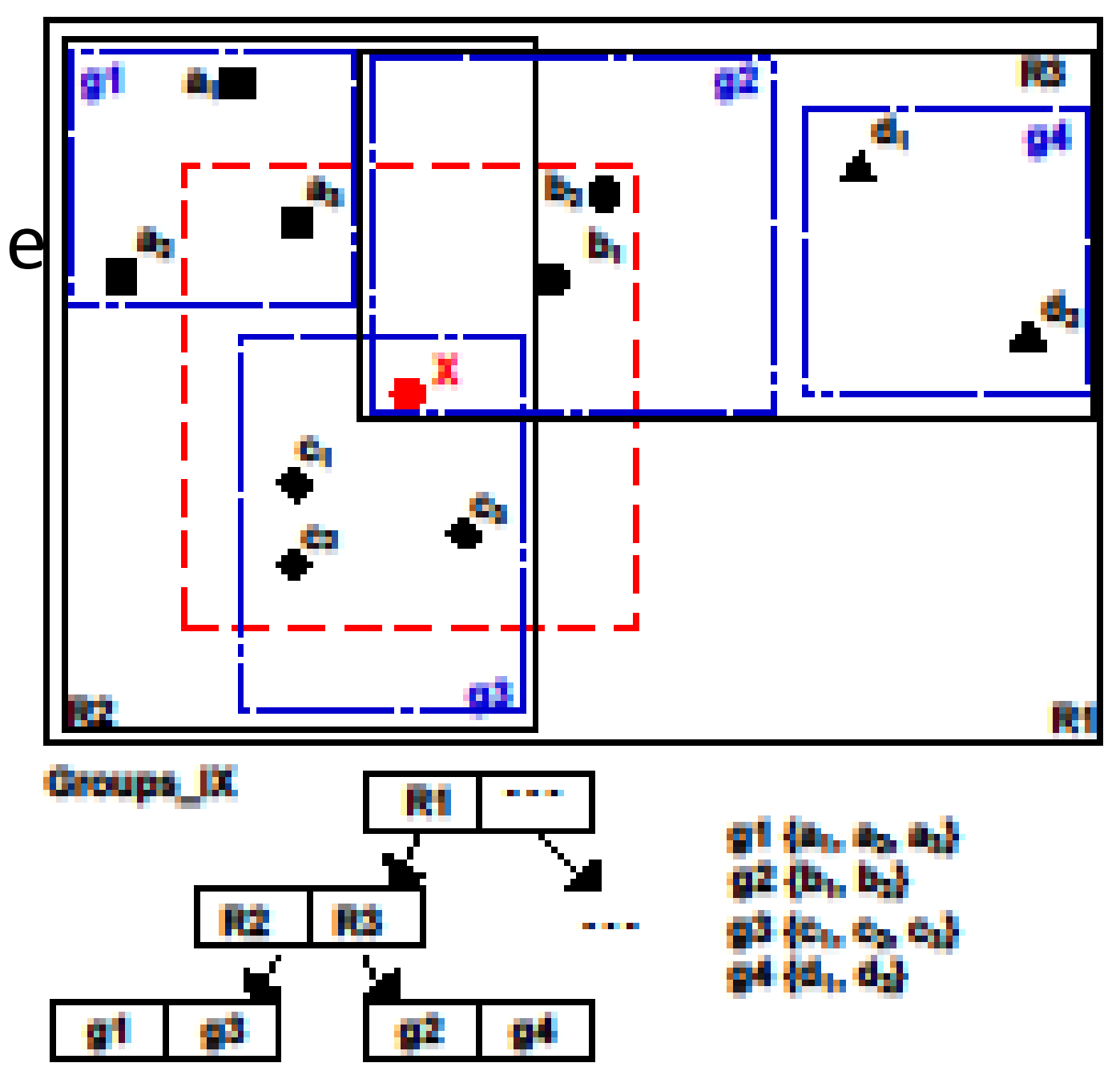


SELECT List-ID (Device-ID) FROM MobileDevices
GROUP BY Device-lat , Device-long
***DISTANCE-TO-ALL L2 WITHIN** SignalRange*
ON-OVERLAP ELIMINATE

Query Optimization

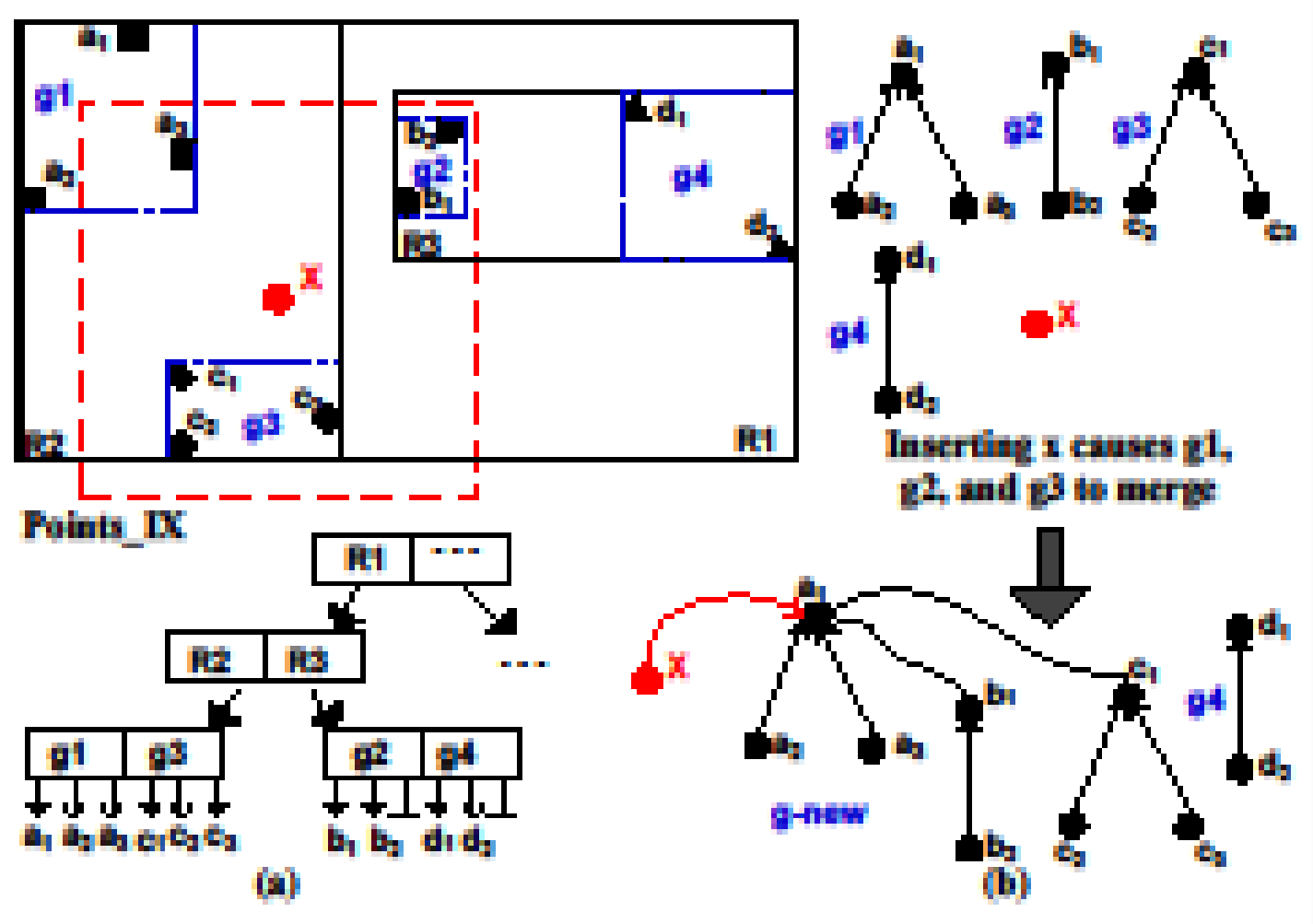
- **SGB-All**

Bounding-rectangle
+ convex hull
+ spatial index
+ disk-based hash tables

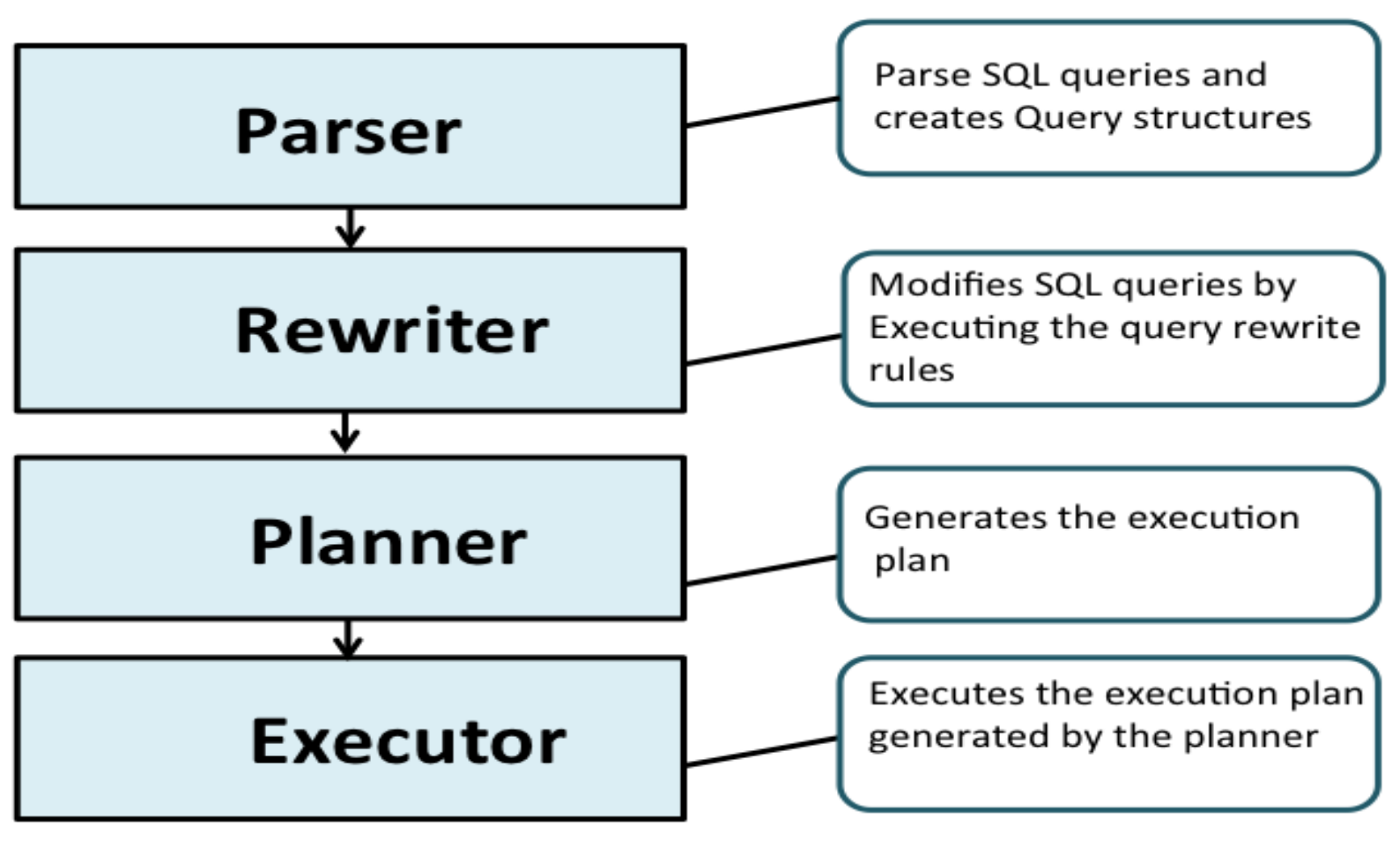


- **SGB-Any**

Spatial index
+ union-find
+ disk-based hash tables



Implementation



Parser — Parse SQL queries and creates Query structures

Rewriter — Modifies SQL queries by Executing the query rewrite rules

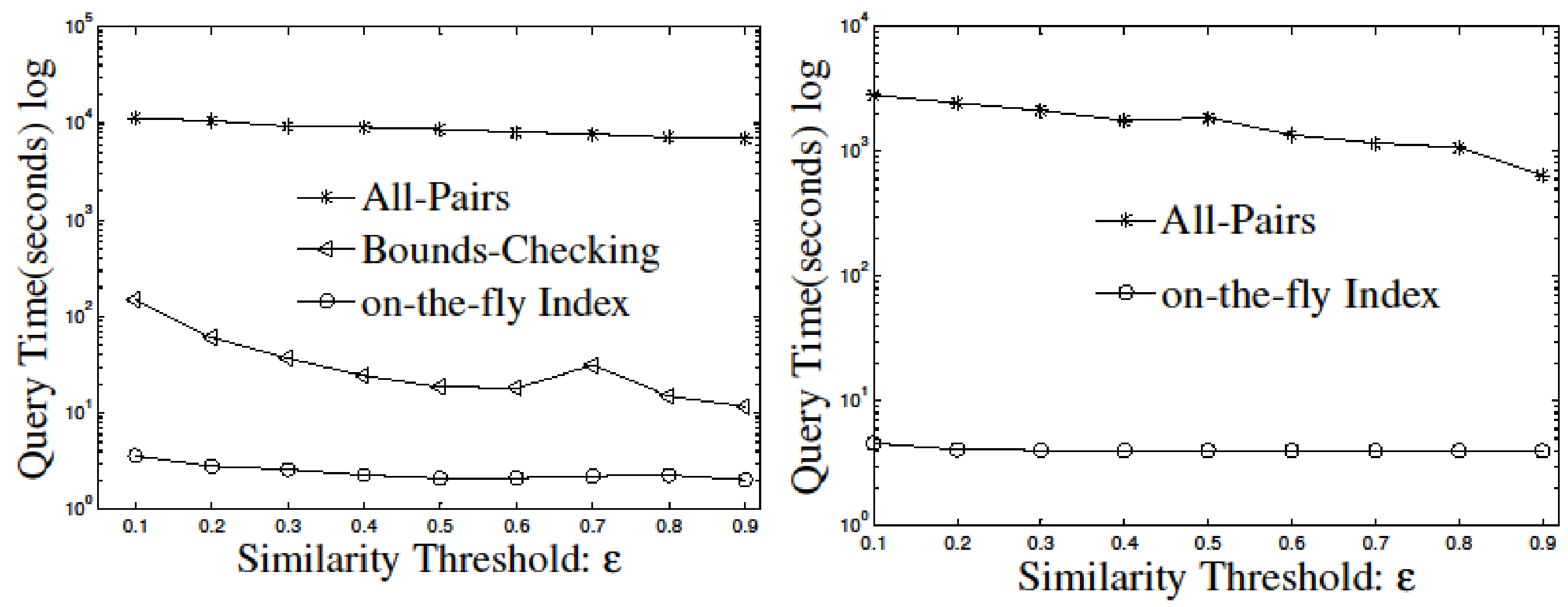
Planner — Generates the execution plan

Executor — Executes the execution plan generated by the planner

- Developed inside PostgreSQL 8.2
  - > 8k lines of codes
  - Uses an in-memory R-tree index inside query executor
  - Memory protection
  - Transaction consistency
  - Fault recovery

Experiments

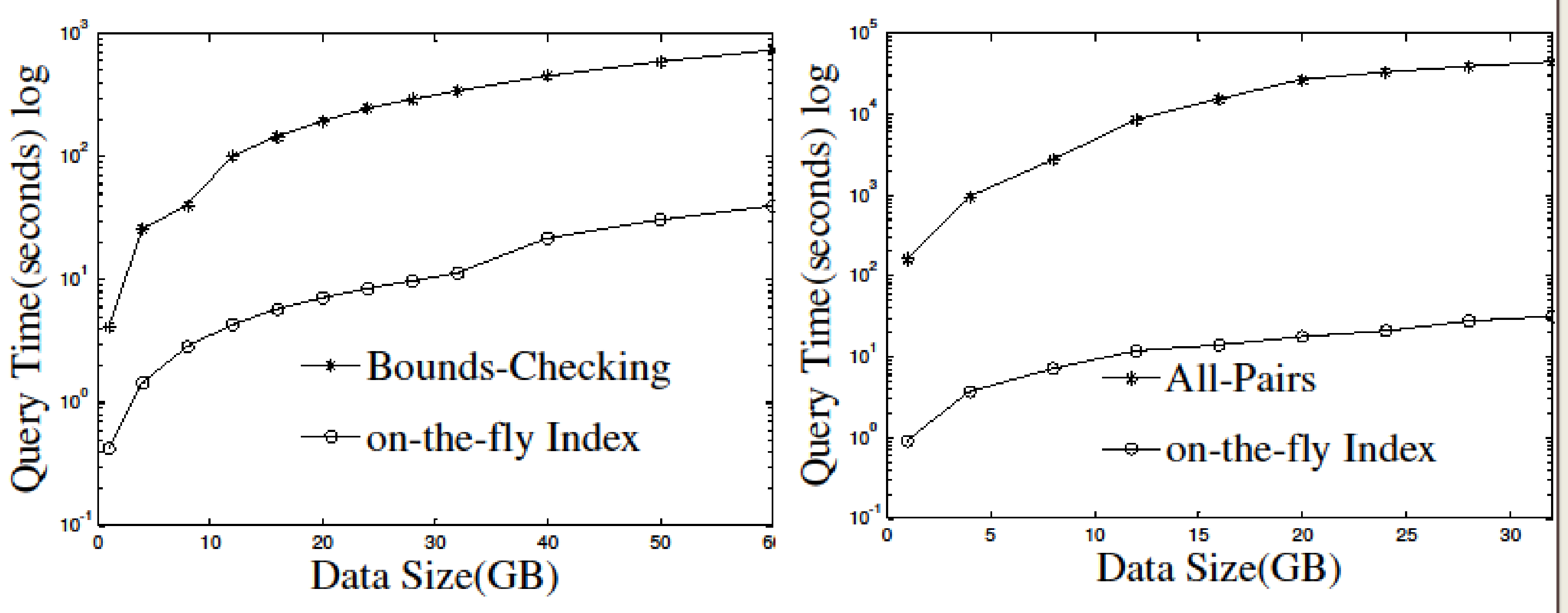
- Tested using TPC-H and social network check-in dataset (Gowalla, Brightkite)
- SGB operators implemented inside PostgreSQL 8.2.4
- Code is available at https://github.com/merlintang/sgb
- Tested query performance against straightforward realization of SGB, various other cluster algorithms, and standard Group-by of PostgreSQL



SGB-All-Join-Any

SGB-Any

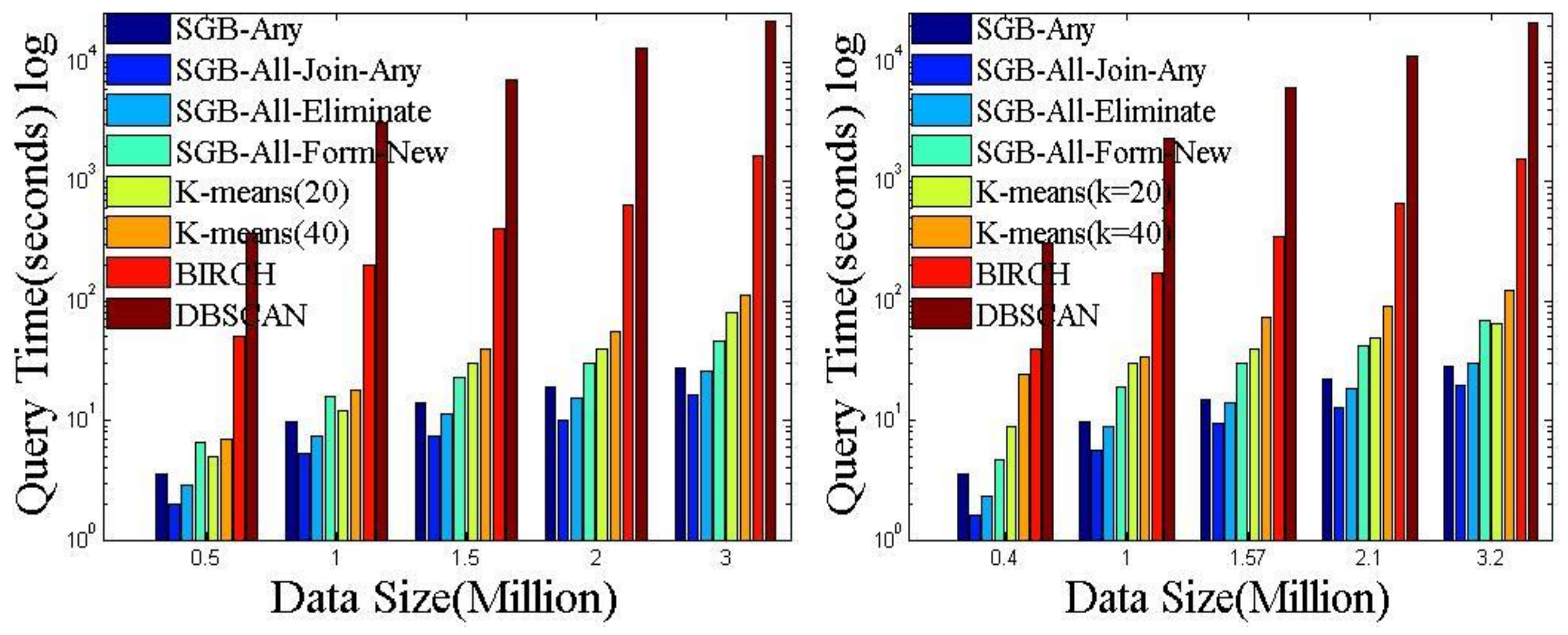
The effect of the similarity threshold ε



SGB-All-Join-Any

SGB-Any

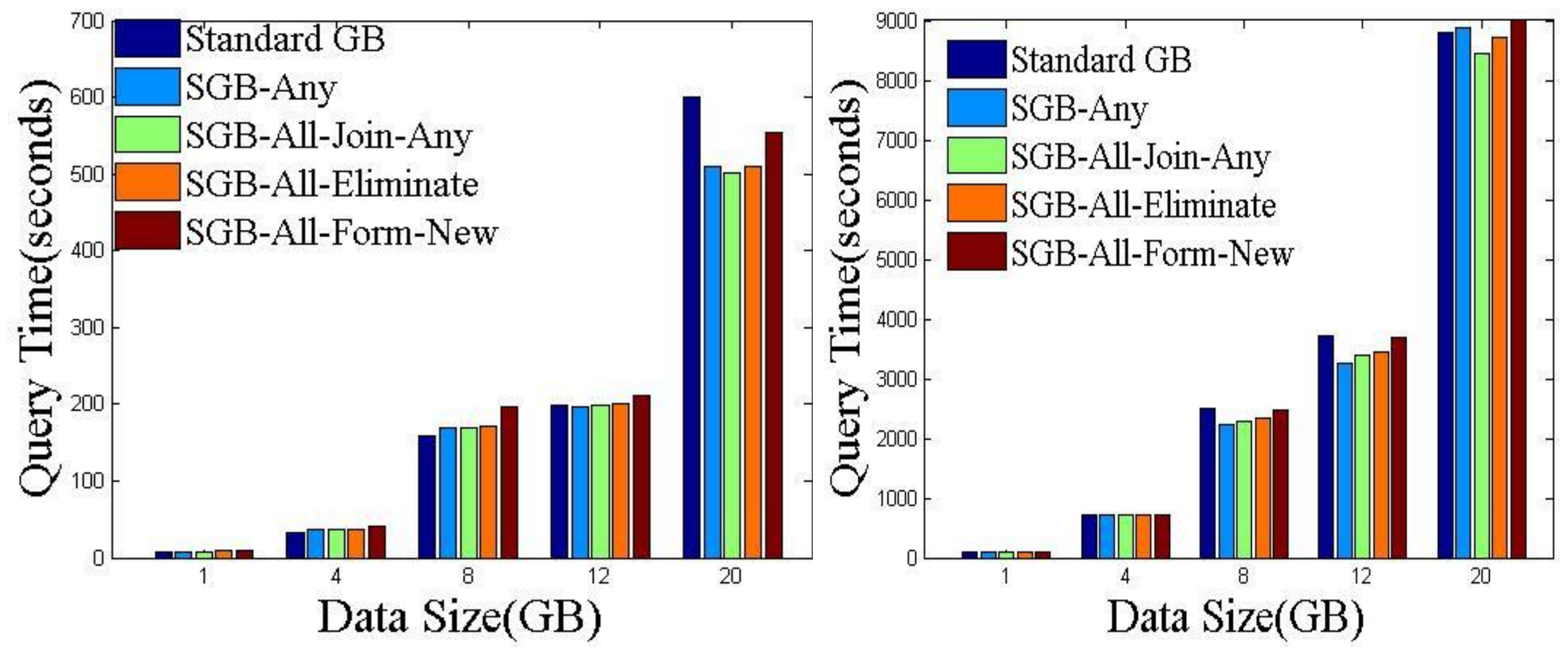
The effect of increasing data size



Brightkite

Gowalla

Comparing with Clustering Algorithm



GBY2 VS SGB3 and SGB4

GBY3 vs SGB5 and SGB6

Comparing with Standard Group By

Related work

- Data cluster algorithms
  - Developed on top of the DBMS
  - Takes the DBMS as a black box
  - Suffers from the extraneous I/O due to impedance mismatch with data in the DB
- Similarity query processing algorithms
  - Well studied, but no previous work on multi-dimensional similarity-group-by

ACKNOWLEDGEMENTS

This work is supported by QNRF Grant No. 4-1534-1-247 and National Science Foundation under Grant III-1117766.