Analytical Queries on Road Networks: An Experimental Evaluation of Two System Architectures

Shangfu Peng
shangfu@cs.umd.edu

Hanan Samet
hjs@cs.umd.edu

Department of Computer Science
University of Maryland
College Park, MD 20742, USA

SIGSPATIAL 2015

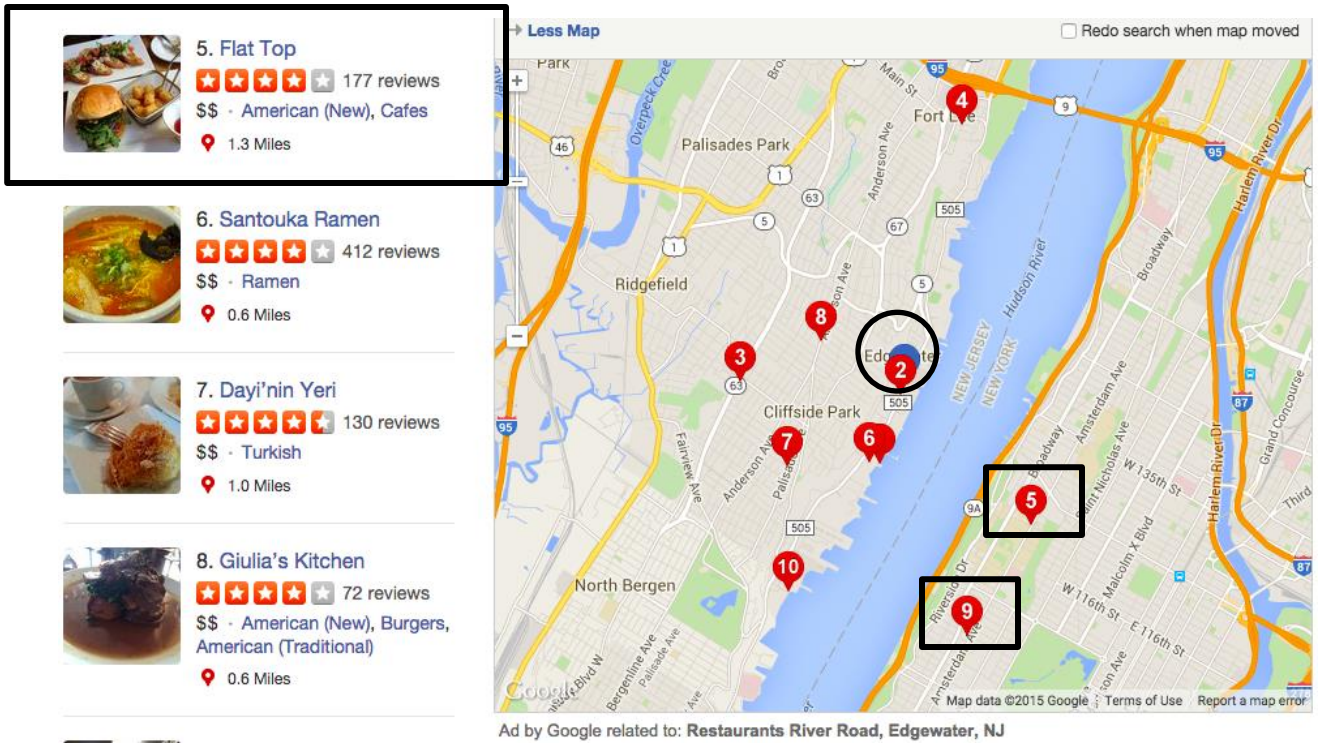
Outline

- Background & Motivation
  - Spatial analytical queries
  - Access patterns & existing methods
- Two architectures
  - Hybrid architecture
  - Integrated architecture (inside a database)
    - ϵ-distance oracle
    - SQL examples
- Evaluation
  - Throughput
  - Region query, KNN query and Density
- Conclusions

Motivation

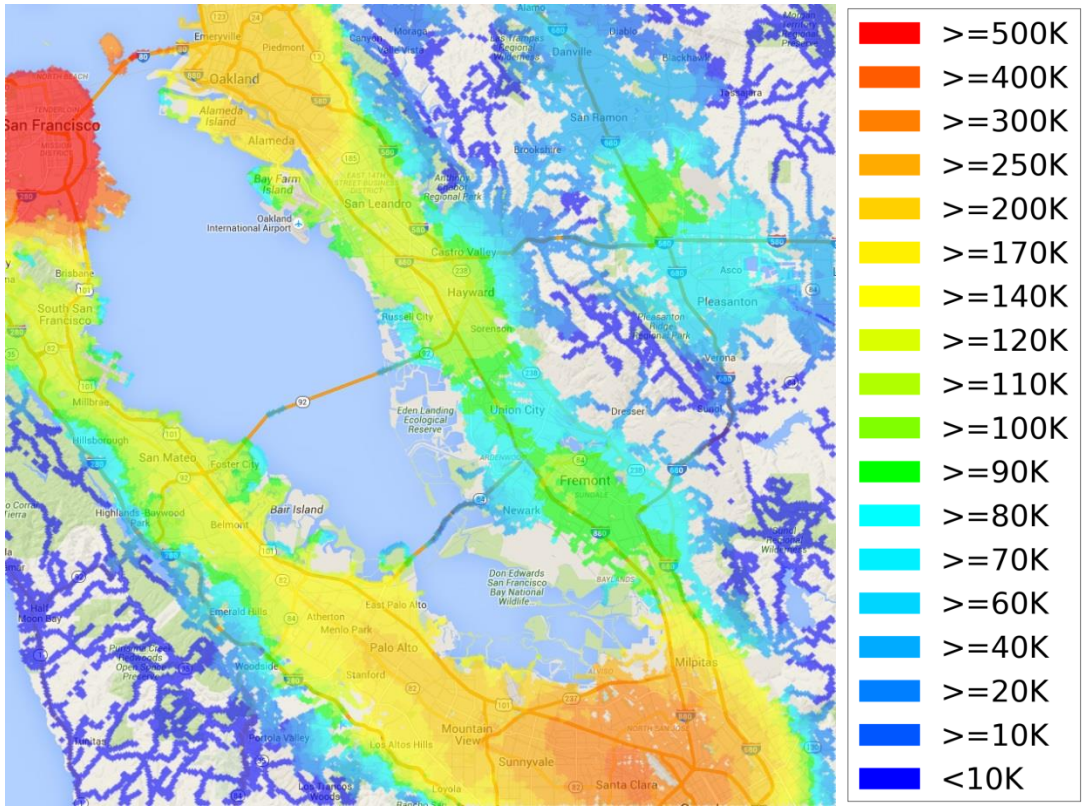
- Analytical queries on data embedded in a road network
  - Requires computing distances using along the road network
  - Much more than simple routing queries
- Ex: Answering queries such as finding all restaurants within a two mile biking distance of River Road in Edgewater, NJ

Yelp results for the query: find the restaurants within a 2 mile biking distance.



Example: Accessibility to Jobs

- Popular among transportation planners
- LODES dataset from census
- Number of jobs within 10KM by car
- Require millions distance computations to generate such heat map



Motivation

- Analytical queries on data embedded in a road network
  - Requires computing distance using along the road network
- Ex: Answering queries such as finding all restaurants within a two mile biking distance of River Road in Edgewater, NJ
- Challenge lies in realization that each such query involves making hundreds to even millions of distance computations along a road network instead of “as the crow flies” (i.e., Euclidean distance)
- Most methods aim at decreasing the latency time for one source-target query
  - Don't take into account factors such as multi-users, multi-threads, query optimization, etc
  - Our aim is to increase the *throughput* which means the total amount of time for the entire query
  - Throughput: number of distance computations per second

Example of Queries from Esri Message Board

1. I am a taxi operator running a fleet of taxis. I have a dataset of taxi trips such that each trip has a latitude and longitude values for both a pickup and a drop off point, as well as for way points at irregular intervals. I want to obtain the total number of miles travelled by each taxi this month.

2. I am an operator of a large hospital and have the geocoded address of my patients and the locations of my clinics. Our hospital has more than 500 clinics across the country. I want to get the average drive time for patients per clinic. This is an important metric in healthcare. This distance also informs us of the need to open new clinics or relocate existing ones to better serve our patients.

3. I have a trucking company with 10 trucks that deliver thousands of packages for a popular retailer. A common operation that I run several times during each day is determining which packages should be loaded on to which truck and the order in which they should be delivered. For this purpose, the input is a network distance matrix between the delivery locations of all current packages. An optimization program would decide how to assign packages to trucks and the order in which to deliver them.

Access Patterns & Algorithms for Distance Queries

- Analytical queries perform the following access patterns on the road network
  - One-to-One: a batch of source-target pairs
  - One-to-Many: KNN
  - Many-to-Many: distance matrix

Decompose

- Algorithms

Algorithm Type	Access Pattern	
	One-to-One	One-to-Many
Scan-based (in memory)	Contraction Hierarchies TNR etc	Dijkstra's algorithm etc
Lookup-based (in database)	ϵ-Distance Oracle Hub labeling	-

Access Patterns & Algorithms for Distance Queries

- Analytical queries perform the following access patterns on the road network
  - One-to-One: a batch of source-target pairs
  - One-to-Many: KNN
  - Many-to-Many: distance matrix

Decompose

- Algorithms

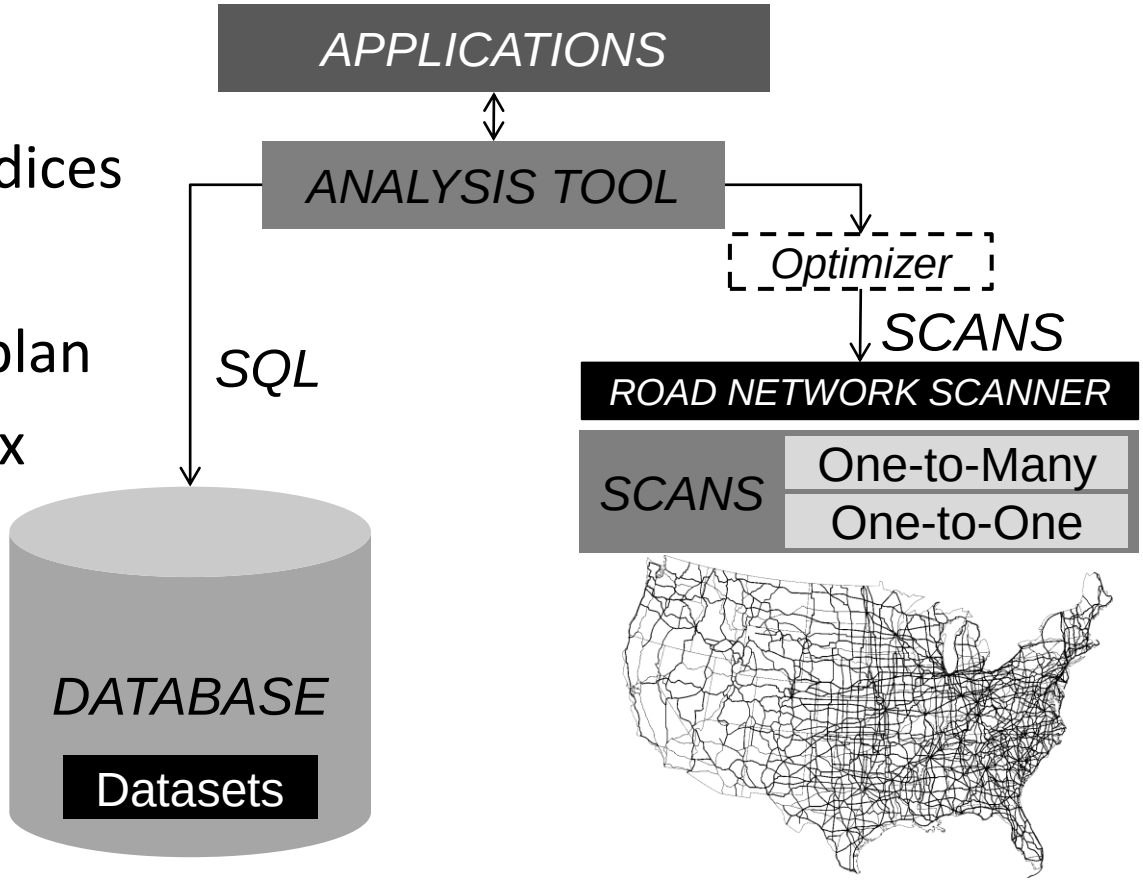
Algorithm Type	Access Pattern	
	One-to-One	One-to-Many
Scan-based (in memory)	Contraction Hierarchies TNR etc	Dijkstra's algorithm etc
Lookup-based (in database)	ϵ-Distance Oracle Hub labeling	-

Outline

- Background & Motivation
  - Spatial analytical queries
  - Access patterns & existing methods
- **Two architectures**
  - Hybrid architecture
  - Integrated architecture (inside a database)
    - ϵ-distance oracle
    - SQL examples
- Evaluation
  - Throughput
  - Region query, KNN query and Density
- Conclusions

Hybrid Architecture

- Decoupling the database from the distance computations
  - Retrieving location datasets from a database
  - Running distance computations in scanner module
- Scanner module
  - In memory
  - Contains spatial indices
- Optimizer module
  - Improve the scan plan
  - E.g, distance matrix $N \times M, N \gg M$
  - Users dispense with this step



Outline

- Background & Motivation
  - Spatial analytical queries
  - Access patterns & existing methods
- Two architectures
  - Hybrid architecture
  - **Integrated architecture (inside a database)**
    - ϵ-distance oracle
    - SQL examples
- Evaluation
  - Throughput
  - Region query, KNN query and Density
- Conclusions

ϵ-Distance Oracle (ϵ-DO)

- Assumes existence of well-separated pair decomposition (WSPD) on the road network
- Can represent the distance between any two points in A and B by one value with an epsilon error tolerance

$$\forall s \in A, t \in B, (1-\epsilon)d_\epsilon(A,B) \leq d_G(s,t) \leq (1+\epsilon)d_\epsilon(A,B)$$

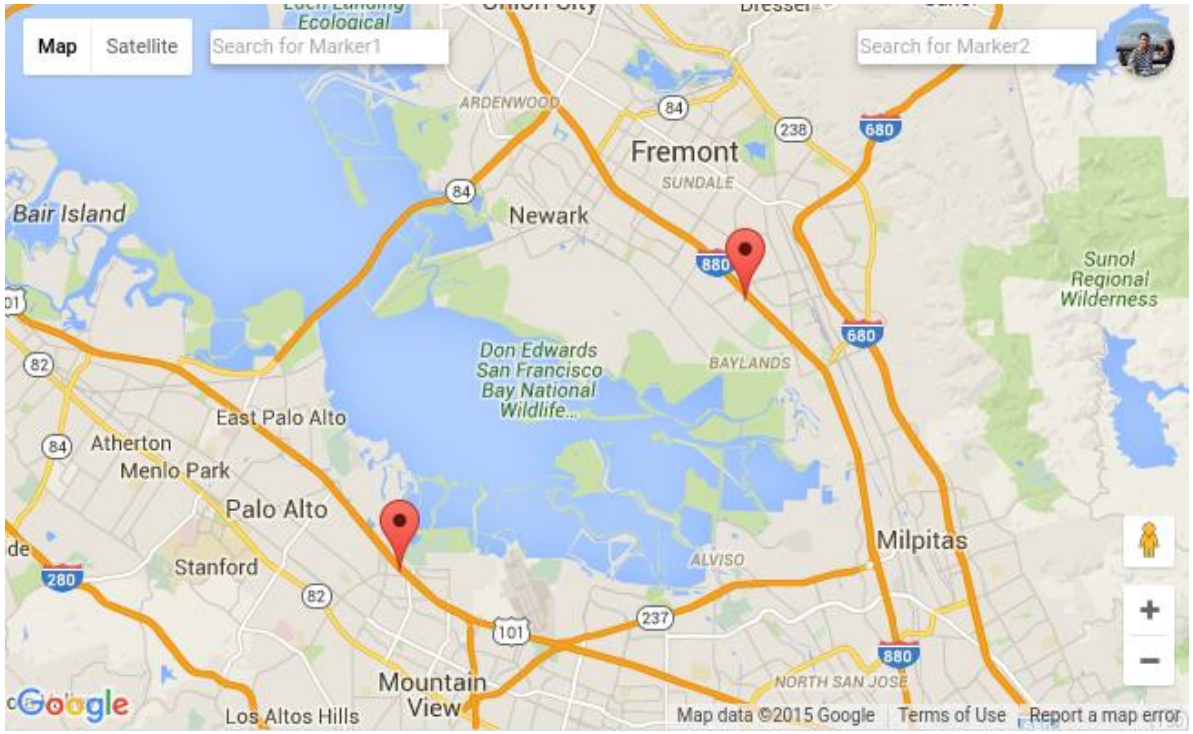
- $O\left(\frac{n}{\epsilon^2}\right)$ well-separated pairs, each well-separate pair can be represented as a key-value pair
  - USA road network with 24 million vertices
  - $\epsilon = 0.25$, 4.6 billion key-value pairs, 55 GB

Example of ϵ-Distance Oracle (ϵ-DO)

- Is ϵ-approximate result good enough?
- In average, the error is less than $\epsilon/10$

Road networks from OpenStreetMap

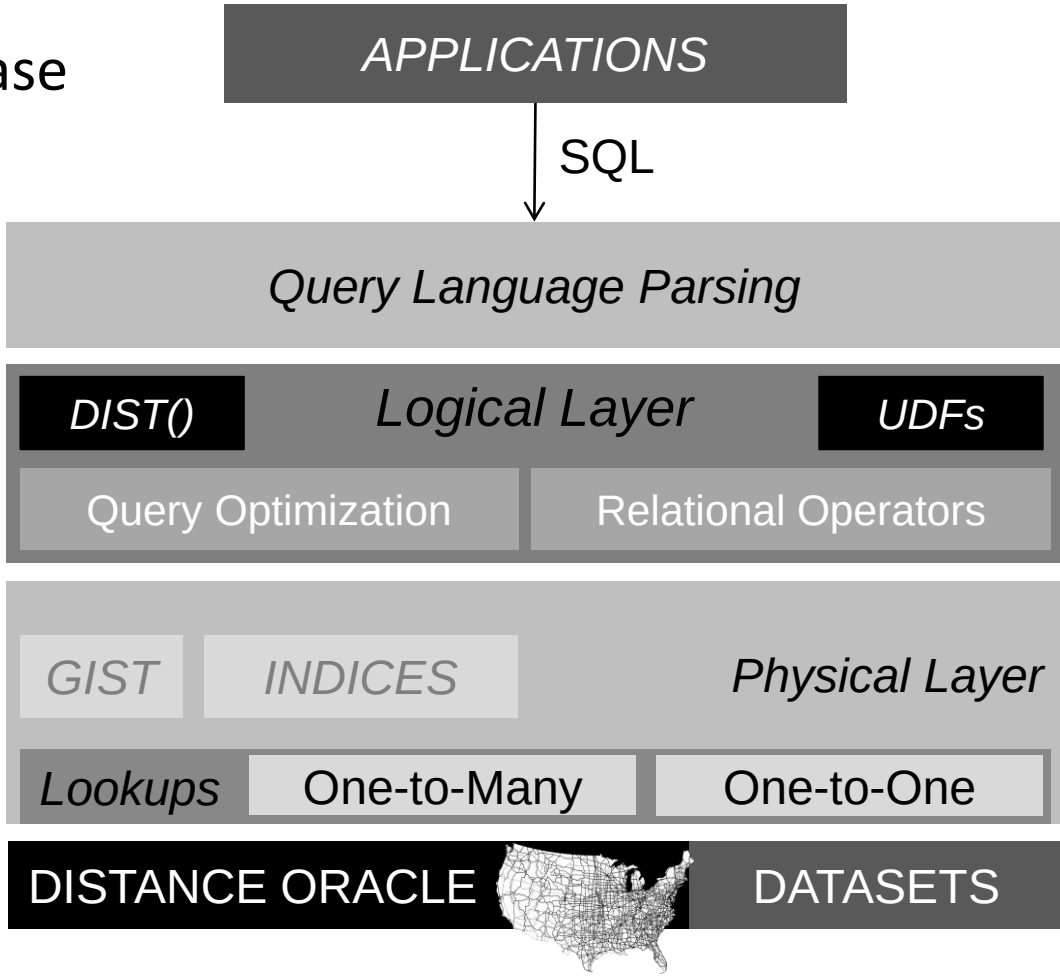
Method	Result
ϵ-DO, $\epsilon = 0.25$	27.802 KM
Exact Distance	28.546 KM
Google Maps	29.3 KM
Euclidean	14.649 KM



$http://sametnginx.umiacs.umd.edu/oracle$

Integrated Architecture

- Embed ϵ-DO into a database
- *Dist(lat1, lon1, lat2, lon2)*
  - A B-tree lookup
- Leverage
  - GiST and Indices
  - Query Optimization
  - SQL language




```
-- Road distance between White House and US Capitol
SELECT DIST(38.8977, -77.0366, 38.8898, -77.0091)
-- This produces 2144.7 (meters)
```

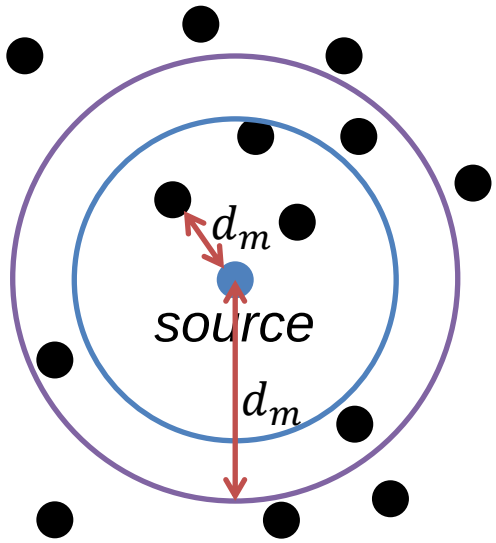
Example Query Using Integrated Architecture

- Find up to 100 houses with the maximum number of parks that lie within 0.5 km of road distance from the houses sorted by the number of such parks.
  - Tables: *houses(id, lat, lon)* and *parks(id, lat, lon)*

```
SELECT id, count(*) as count
FROM ( SELECT houses.id as id,
              DIST(houses.lat, houses.lon,
                   parks.lat,  parks.lon) as dist
       FROM houses, parks
      ) as foo
WHERE dist < 500  -- 0.5km in meters
GROUP BY id
ORDER BY count DESC
LIMIT 100;
```

KNN Example Using Integrated Architecture

- *K* nearest neighbors (One-to-One version)
  - *POI(id, lat, lon)* and a source location
  - Need to avoid computing all pairs of the source and *POI*



K=3

Algorithm:

1. Obtain the *K nearest* Euclidean distance neighbors using GiST
2. Compute their network distances and select the maximum one, denoted as d_m
3. Obtain all nearest objects whose Euclidean distance is less than or equal to d_m , and then compute their network distances
4. Retain the *K* closest ones

- Correctness: Euclidean distance is a lower bound on the network distance
  - Network distance of any point outside the red circle > d_m

Outline

- Background & Motivation
  - Spatial analytical queries
  - Access patterns & existing methods
- Two architectures
  - Hybrid architecture
  - Integrated architecture (inside a database)
    - ϵ-distance oracle
    - SQL examples
- **Evaluation**
  - Throughput
  - Region query, KNN query and Density
- Conclusions

Evaluation

- Setting
  - Integrated architecture using ϵ-DO in PostgreSQL
  - Hybrid architecture using Dijkstra's algorithm
    1. Single thread version
    2. Multi-thread version (7 scanning threads in a 8-core machine)
- Dataset
  - USA road network containing 23,947,347 vertices and 58,333,344 edges
  - 49,573 locations of fast food restaurants in USA
  - 5,964 locations of universities and schools in USA
  - 11,220,058 GPS entries for 537 taxis in a one month in San Francisco

Throughput Evaluation

- How many distance computations per second?
- One-to-one access pattern
  - Integrated is far superior to Hybrid
- One-to-many access pattern
  - Hybrid (Dijkstra's algorithm) amortizes the costs of scans from a single source to multiple destinations
  - Therefore, the Hybrid throughput for a distance matrix is much better than the one for random pairs

Query	Metric	Integrated	Hybrid - 7
10K random pairs	Time	0.327 sec	2026 sec
	Throughput	**30581 dist/sec**	**4.9 dist/sec**
Distance Matrix (5,964×49,573)	Time	8853.9 sec	20139 sec
	Throughput	**33392 dist/sec**	**14680 dist/sec**

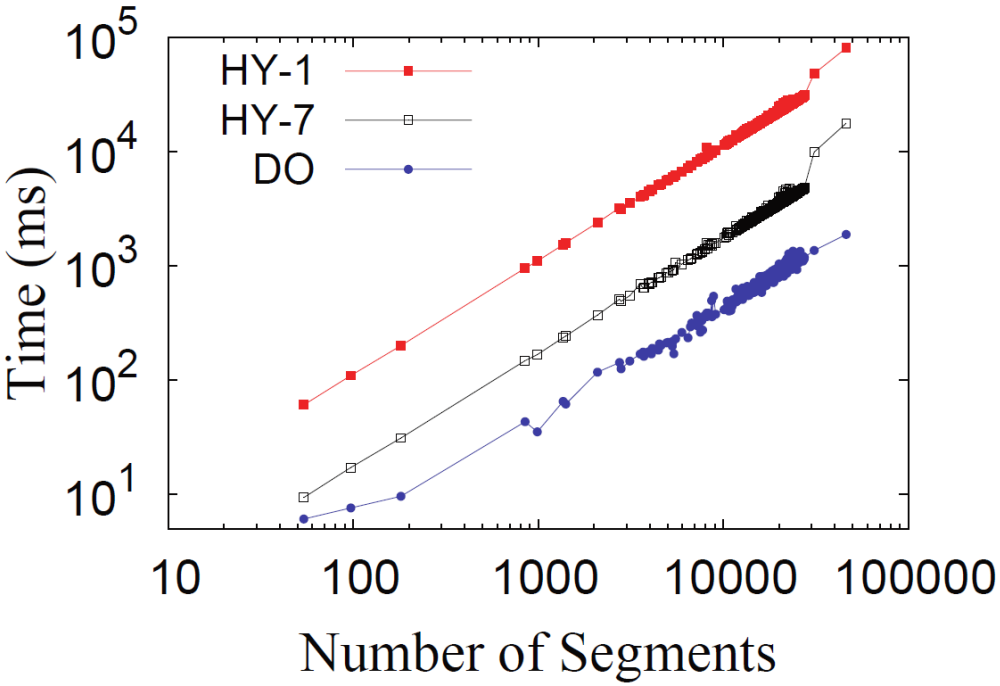
Evaluation of the Effect of Distance

- Distance between points influences the time cost in the one-to-one access pattern
  - Execution time increases with distance between points for most scan-based methods (Hybrid)
  - Execution time is not related to distance for most lookup-based methods (Integrated)

Travel distance of each taxi

The endpoints of each segment are near to each other

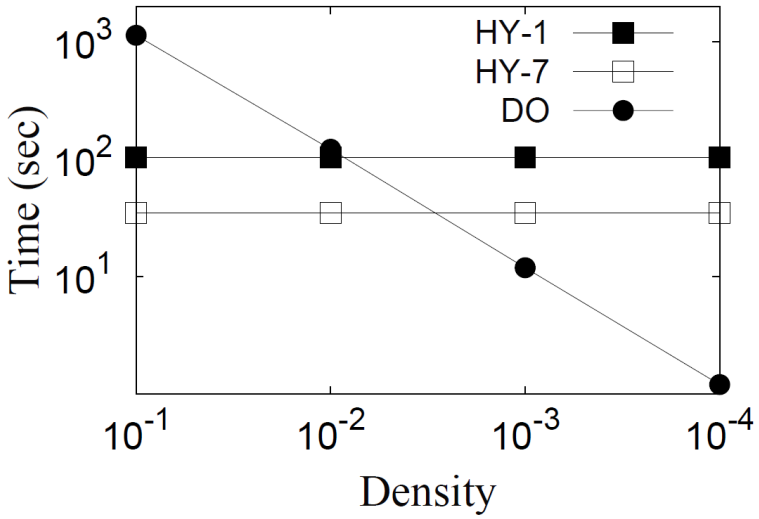
See *http://roadsindb.com/*



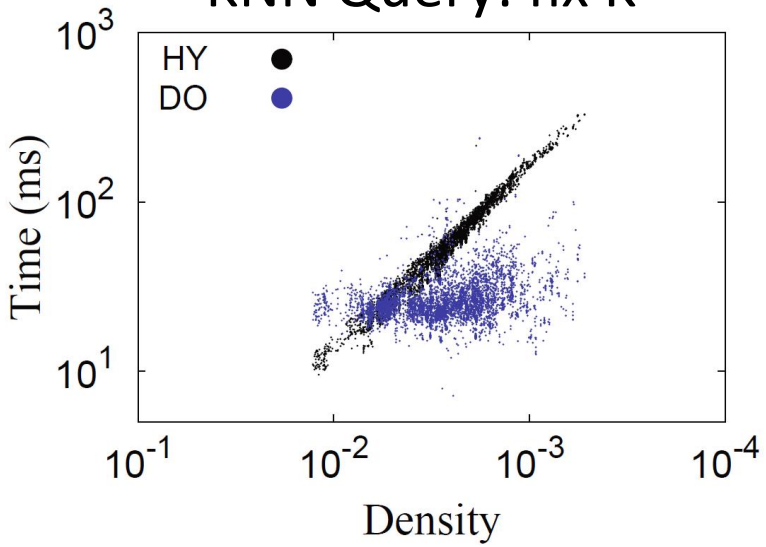
Evaluation of the Effect of Density

- Density influences time cost in the one-to-many access pattern
  - Ratio of number of destinations to number of vertices of the road network visited for a given region
  - Time cost increases with density in the algorithms that are good for one-to-one (Integrated)
  - Time cost is not related to density in the algorithms that are good for one-to-many (Hybrid)

Region Query: fix search space



KNN Query: fix K



Outline

- Background & Motivation
  - Spatial analytical queries
  - Access patterns & existing methods
- Two architectures
  - Hybrid architecture
  - Integrated architecture (inside a database)
    - ϵ-distance oracle
    - SQL examples
- Evaluation
  - Throughput
  - Region query, KNN query and Density
- **Conclusions**

Conclusions

- One-to-One:
  - Integrated is much better than Hybrid in time and throughput
- One-to-Many
  - Depends on the density of targets
  - Integrated is faster than Hybrid in most real applications
- Ease of use
  - Integrated is much better than Hybrid

- Future work
  - Embed it into a distributed memory system, e.g., Spark
  - Aware of traffic changes for different periods of time


Thanks

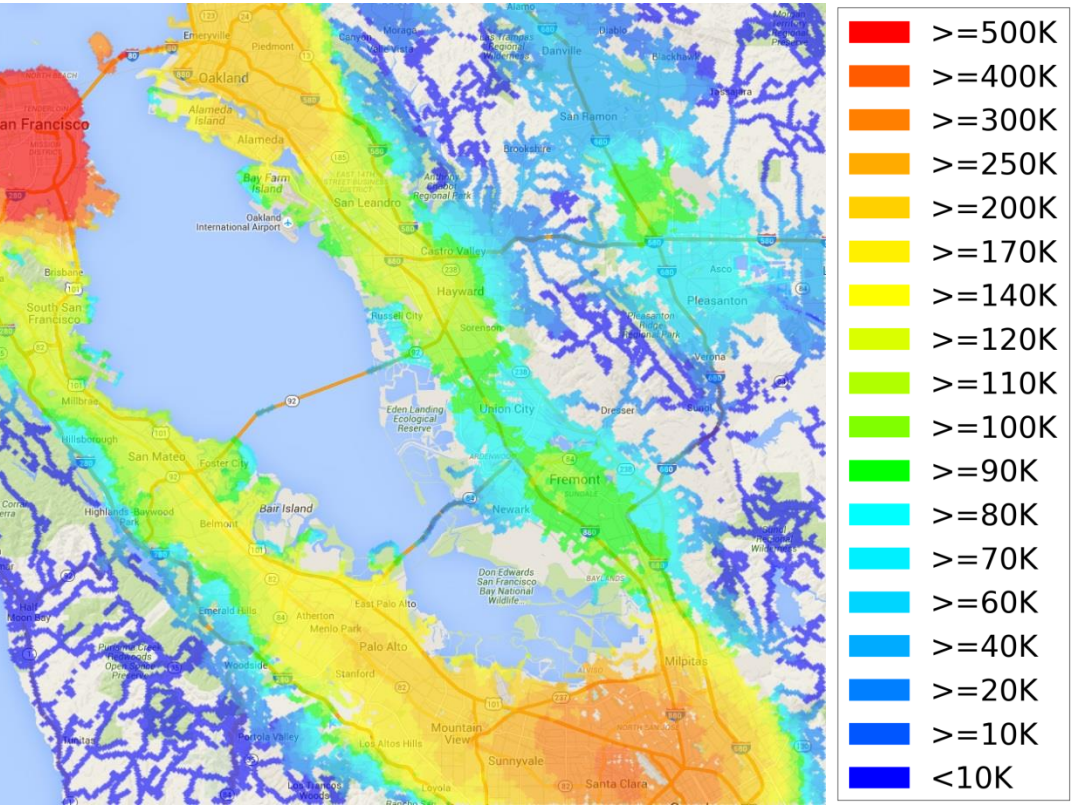
Motivation

- Throughput
  - How many distance computations per second
- Existing Solutions
  - Focus on decreasing the latency time
  - Google Distance Matrix
    - » Limit non-paying users to submit 100 shortest distances per query
    - » Limit paying customers to submit 625 shortest distances per query
  - ArcGIS Network Analyst extension in Esri
    - » Using Dijkstra's algorithm
- Require a high throughput solution!

ϵ-Distance Oracle

- Accessibility to jobs in the Bay Area
  - Jobs that are accessible within 10KM by car

ϵ-DO, $\epsilon = 0.25$



Dijkstra's algorithm