

# Performance evaluation of social networking services using a spatio-temporal and textual Big Data generator

*Diploma Thesis*

Thaleia-Dimitra Doudali

# Thesis contribution

---

1. Design and implementation of a parameterized generator of spatio-temporal and textual social media data
2. Creation of a large dataset using the generator
3. Storage of the dataset into an Hbase distributed database system
4. Scalability testing of the Hbase cluster

# Motivation

---

- Era of Big Data
- Polymorphic social media data
- Transition to distributed storage and processing tools
- Limited access to such data due to privacy restrictions
- Restricted evaluation of distributed data management tools

# Generator

---

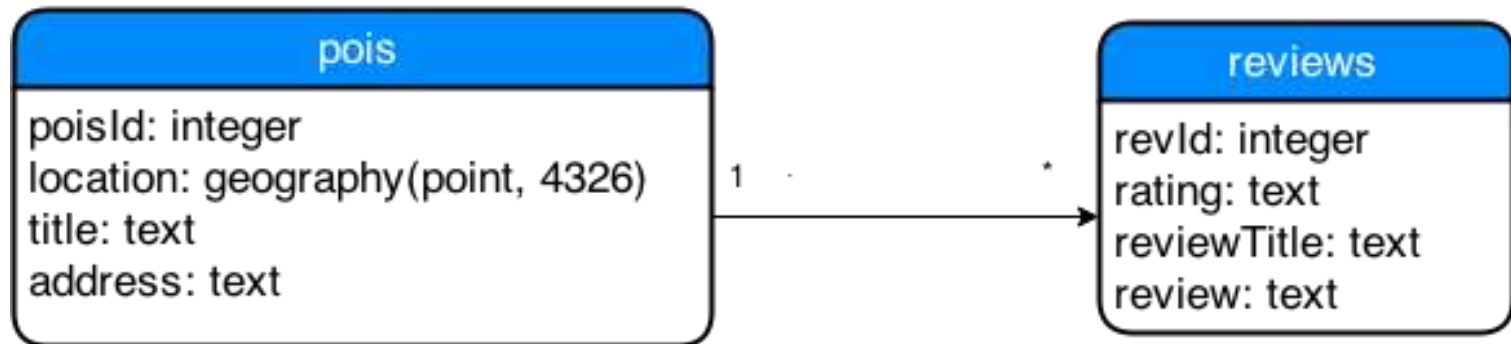
- Spatio-temporal and textual data
- Users of social networking service
- Daily Check-ins to Points of Interest leaving a review and rating
- GPS traces indicating the routes
- Static Map representation

# Source Data

---

- Real Points of Interest crawled from TripAdvisor
- 136409 points = 13 GB JSON file
- Storage in PostgreSQL
- PostGIS extension offers functions and indexes for geographic data types

# Source data schema



# Input Parameters

---

- `userIdStart`, `userIdEnd`
- `startTime`, `endTime`
- `startDate`, `endDate`
- `dist`, `maxDist`
- `chkNumMean`, `chkNumStDev`
- `chkDurMean`, `chkDurDev`

# Implementation

---

## Check-ins:

- Number of daily check-ins defined using a gauss distribution
- First ever check-in = home location
- First check-in randomly chosen using uniform distribution
- It should be in maxDist range from home
- Rest check-ins of the day should be in walking distance (parameter dist)
- Assign random rating and review using uniform distribution



# Implementation

---

Path between check-ins:

- Google Directions API
- JSON response file containing the path and duration
- Encoded polyline representation of the path
- Extracted geographical points as GPS traces

# Implementation

---

## Timestamps:

- First check-in of the day → `startTime`
- Duration of each visit → Gauss distribution
- Time of next check-in = time of previous one + duration of visit + duration of walk
- Should not exceed `endTime`
- GPS trace timestamp = splitted walk duration

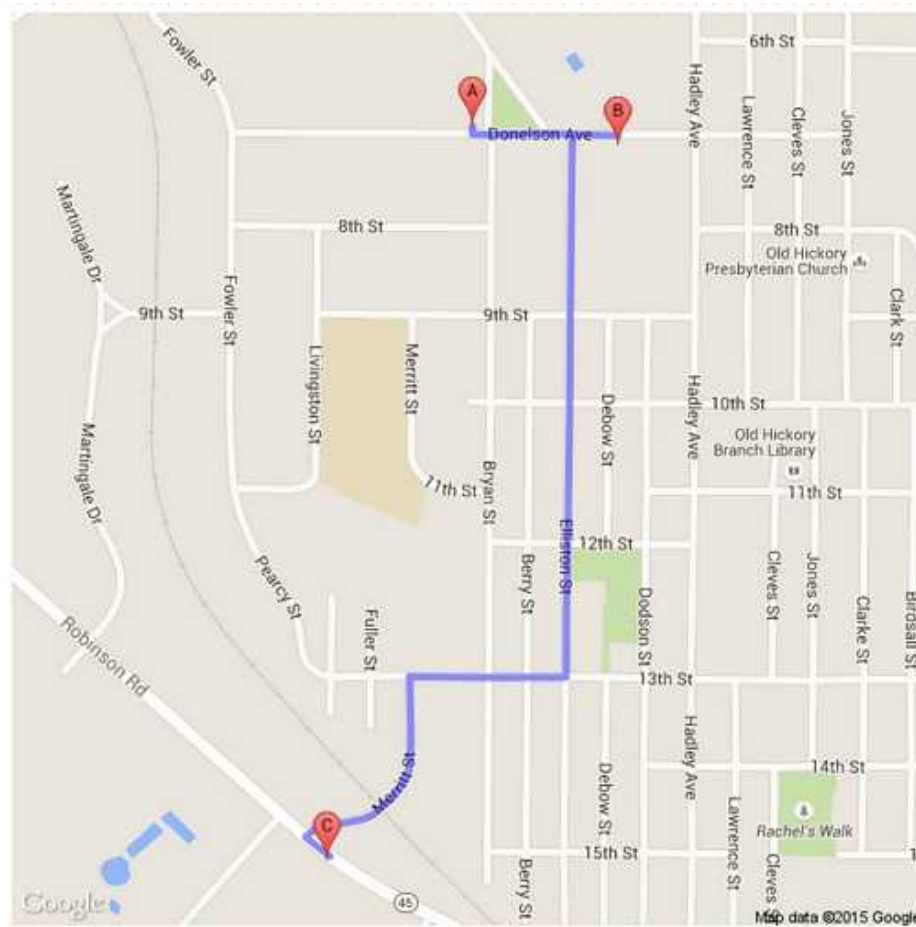
# Implementation

---

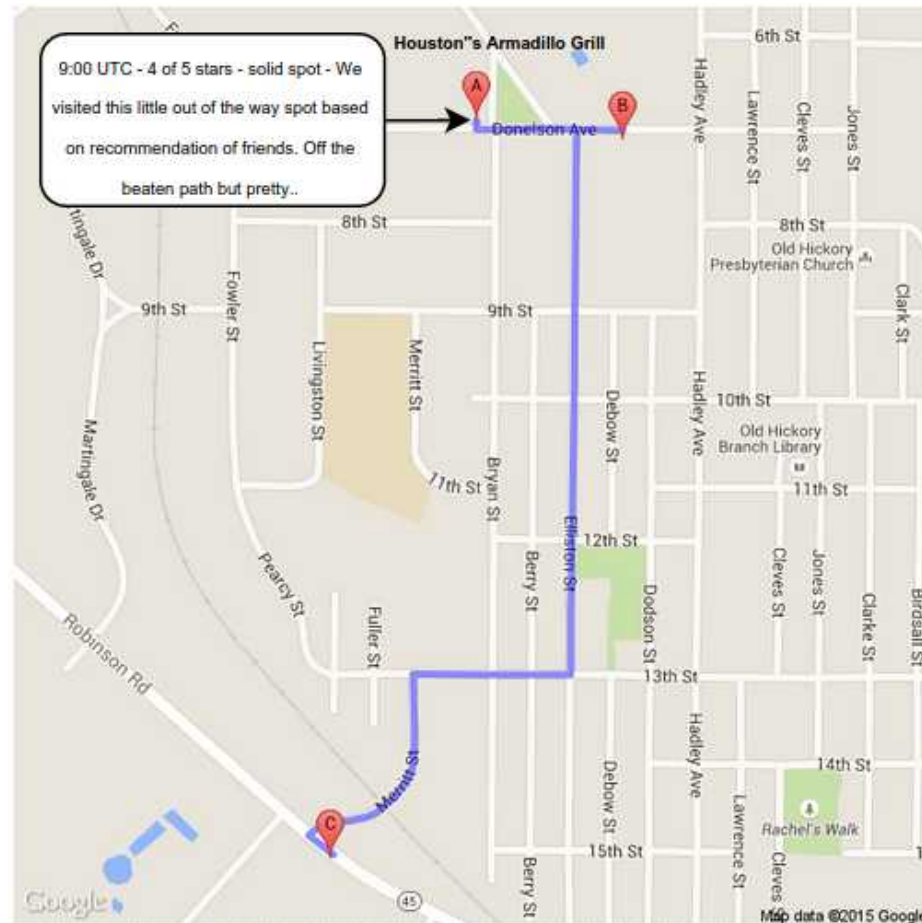
## Trips:

- Travel location equivalent to home
- Available travel days = 10% (endDate – startDate)
- Trip duration = Gauss with  $\mu = 5$  and  $\sigma = 2$
- Decision to start trip  $\rightarrow$  coin toss every day

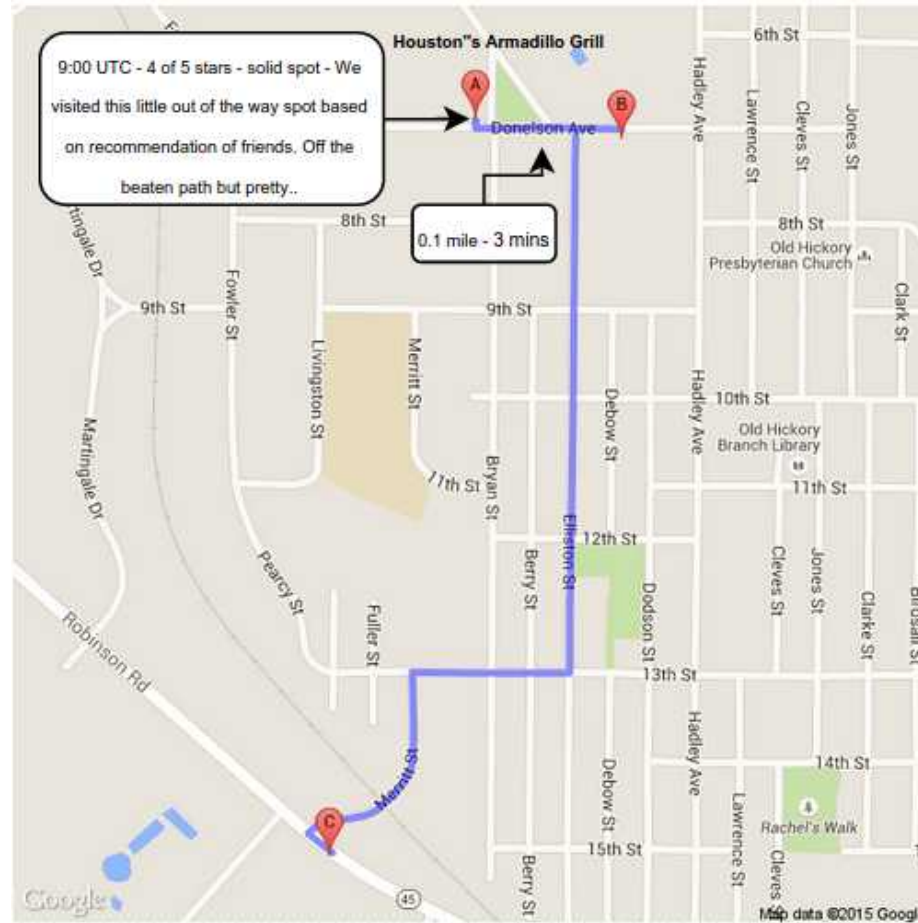
# Static Map



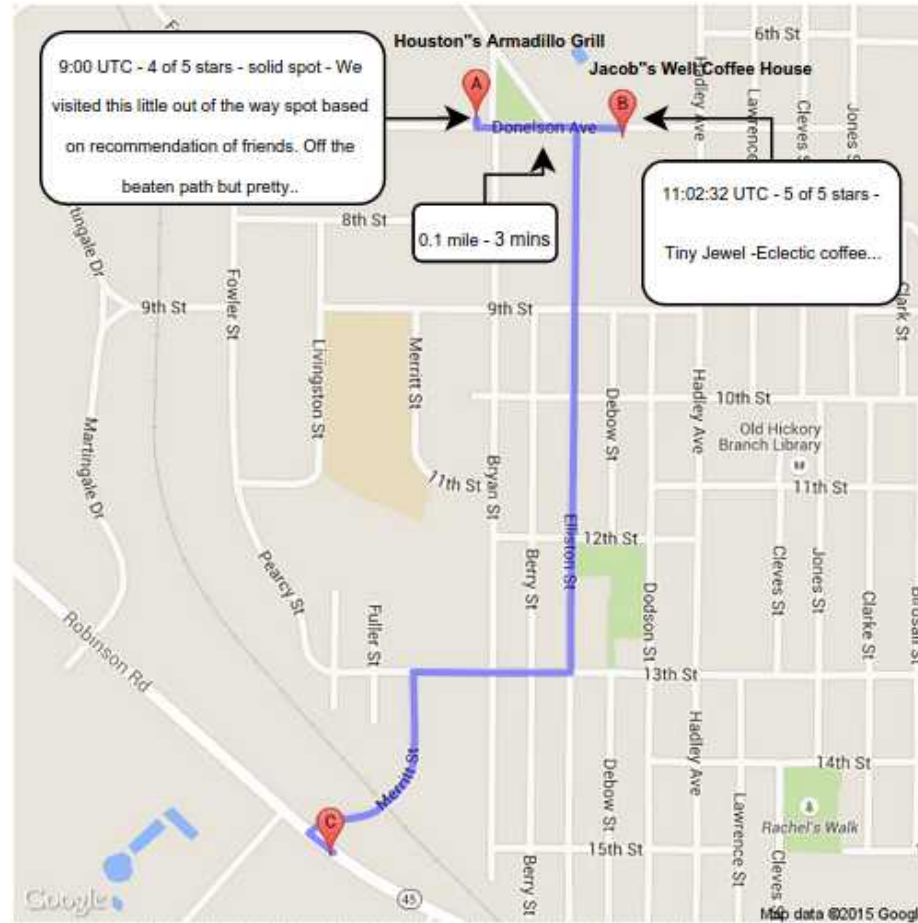
# Static Map



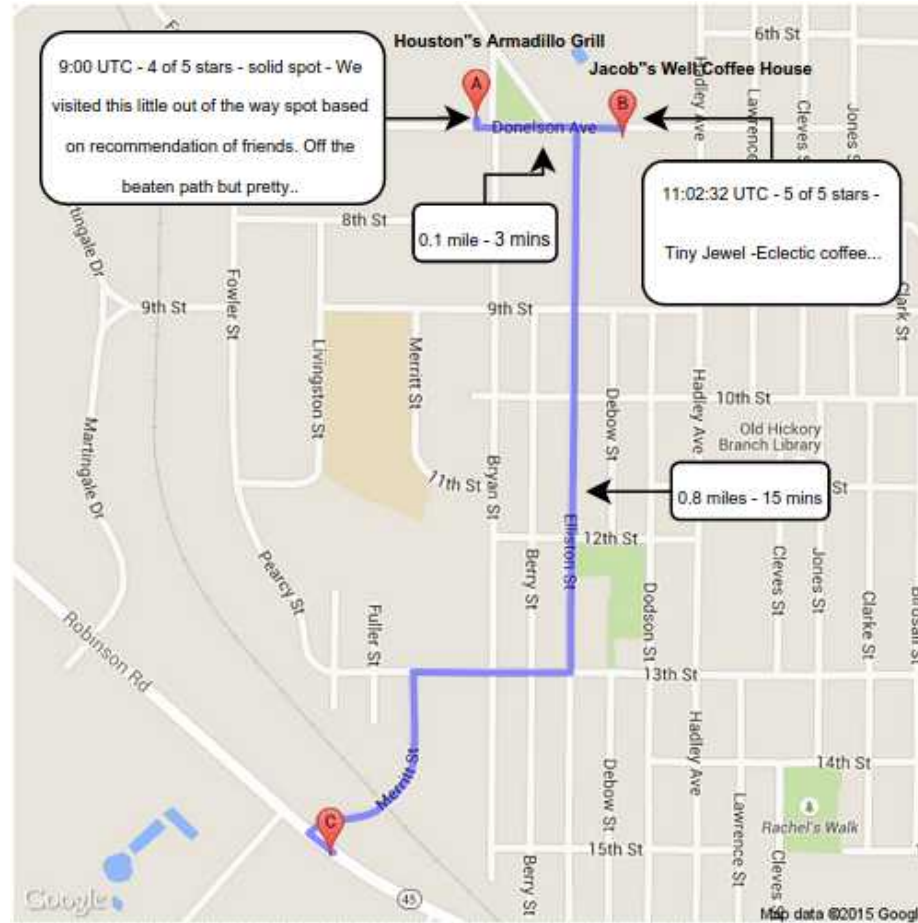
# Static Map



# Static Map

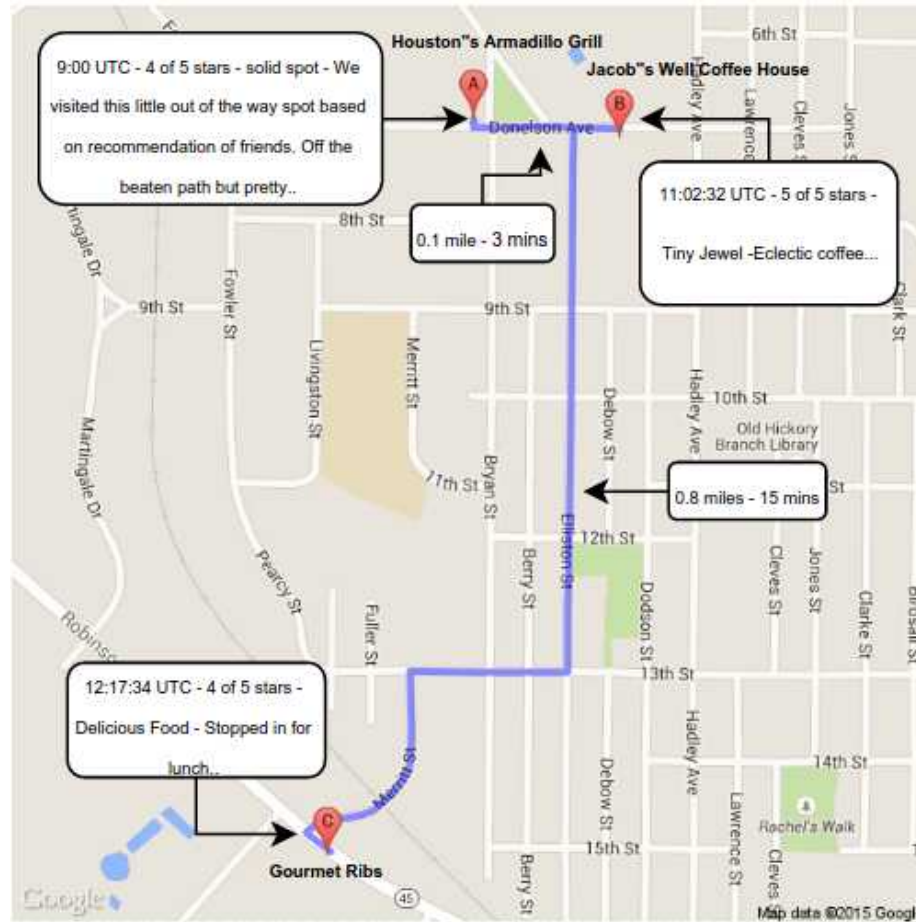


# Static Map

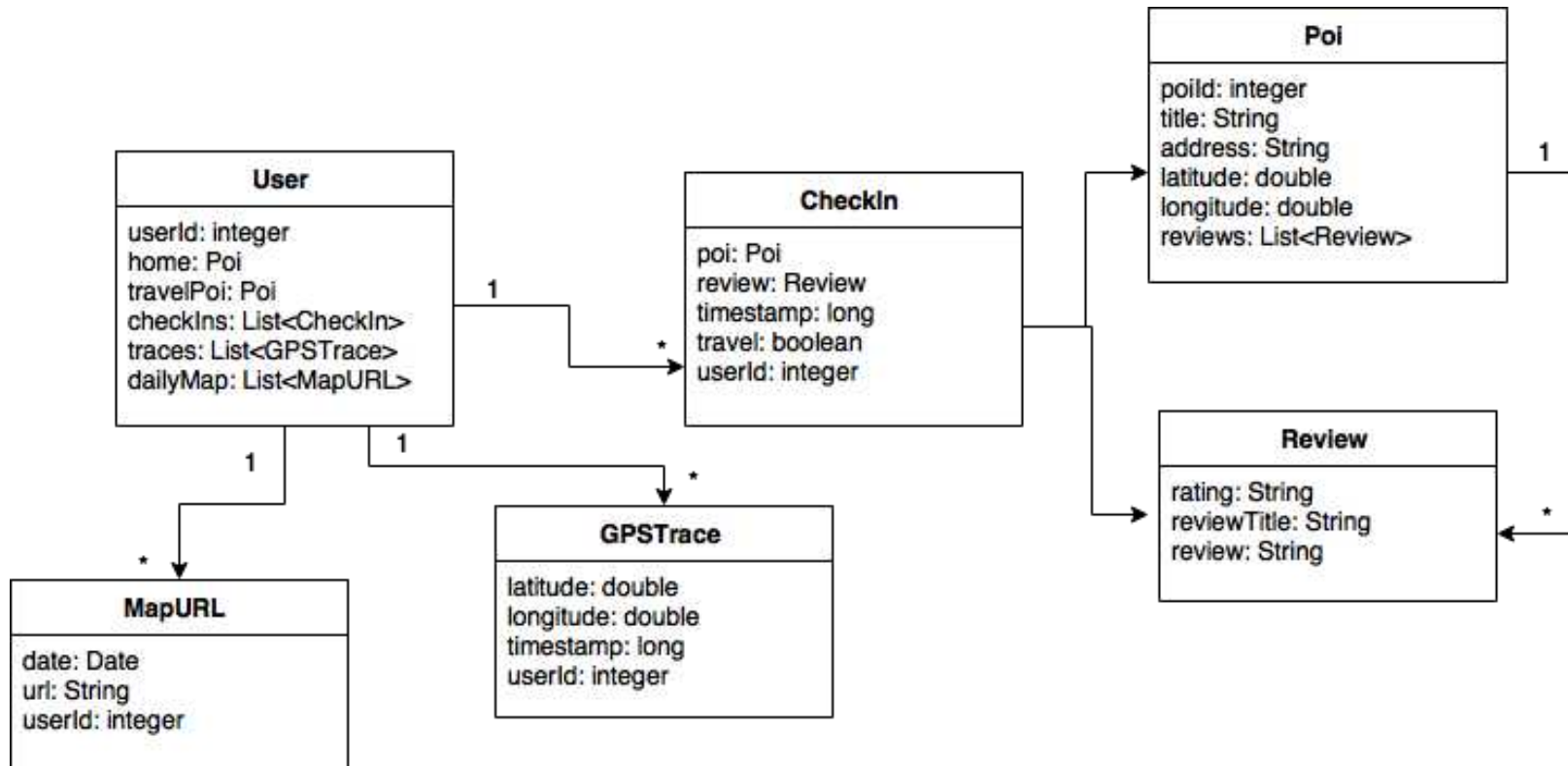




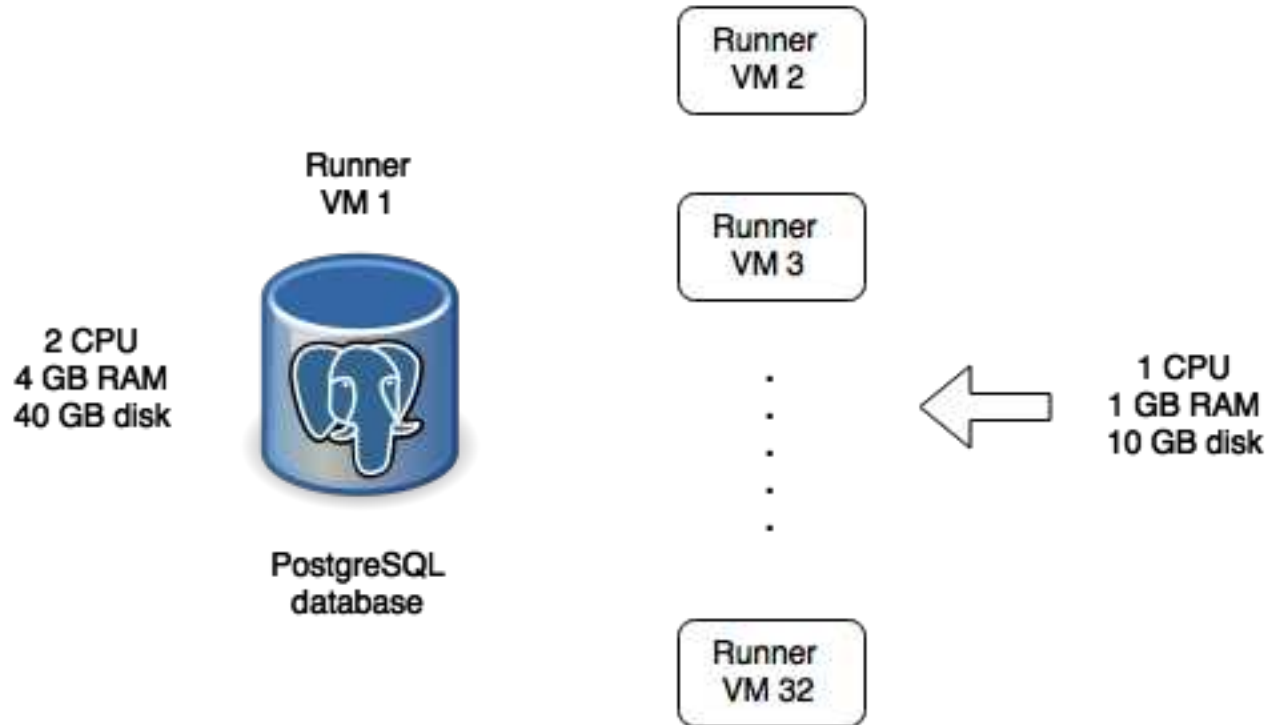
# Static Map



# Generator Attributes



# Generator Deployment Setup



# Execution Input Parameters

---

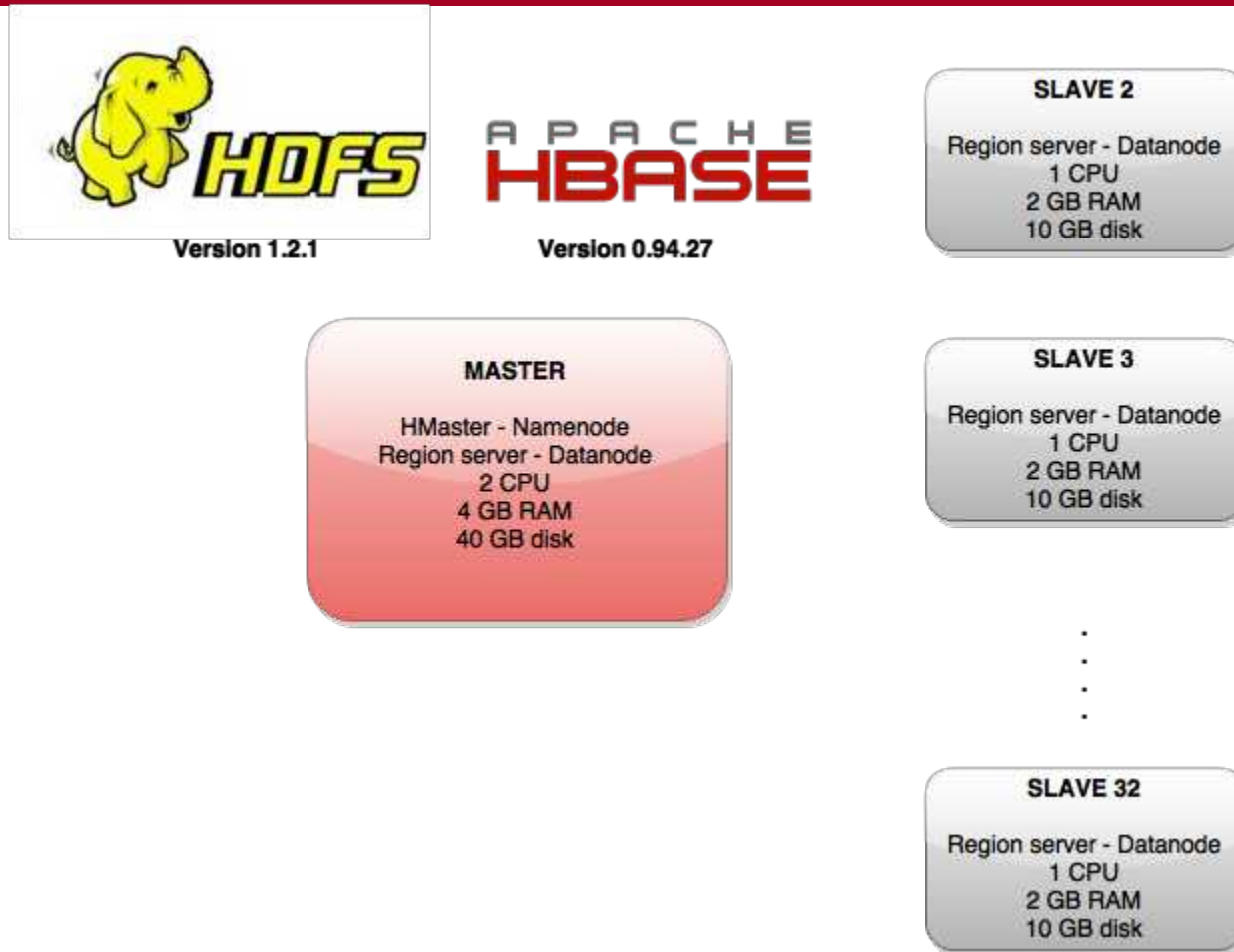
- $\text{chkNumMean} = 5$   $\text{chkNumStDev} = 2$
- $\text{chkDurMean} = 2$   $\text{chkDurStDev} = 0.1$
- $\text{maxDist} = 50000.0$   $\text{dist} = 500.0$
- $\text{startTime} = 9$   $\text{endTime} = 23$
- $\text{startDate} = 01-01-2015$   $\text{endDate} = 03-01-2015$

# Generated Dataset

---

- 9464 users with 2 months daily routes
- 1,586,537 check-ins → 641 MB
- 38,800,019 GPS traces → 2.4 GB
  
- Added a 14 GB twitter friend graph

# HBase cluster



# HBase data model

---

- Friends table
  - Row: user id
  - Column Qualifier: friend user id
  - Cell Value: friend user id
- Check-ins table
  - Row: user id
  - Column Qualifier: timestamp
  - Cell Value: check-in data
- GPS traces table
  - Row: user id
  - Column Qualifier: “lat long timestamp”
  - Cell Value: GPS trace data

# Queries

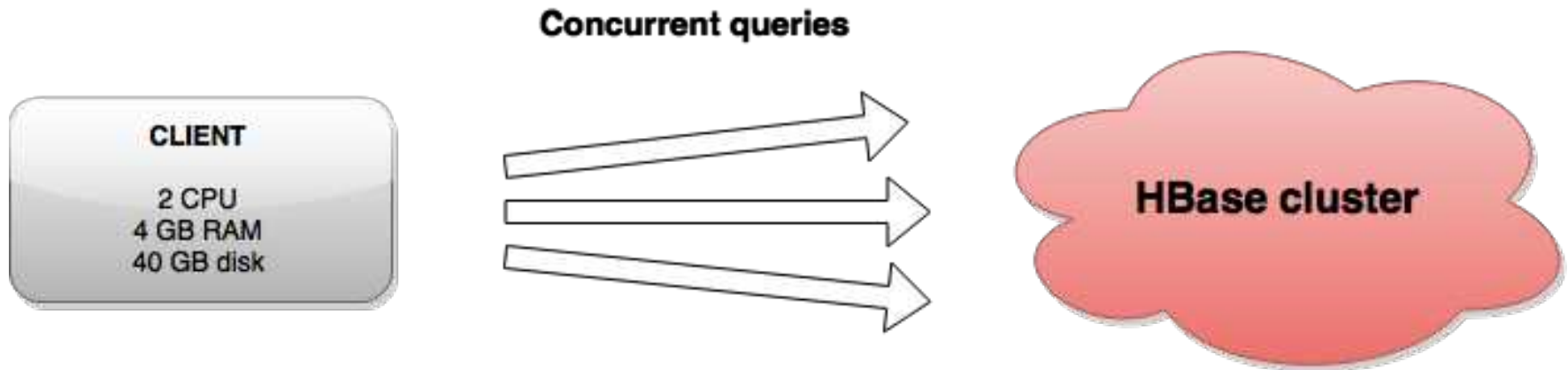
---

1. Get the most visited points of interest of a certain user's friends
2. Get the check-ins of all the friends of a specific user for a certain day into chronological order (News Feed)
3. Get the number of times that a user's friends have visited the user's most visited POI

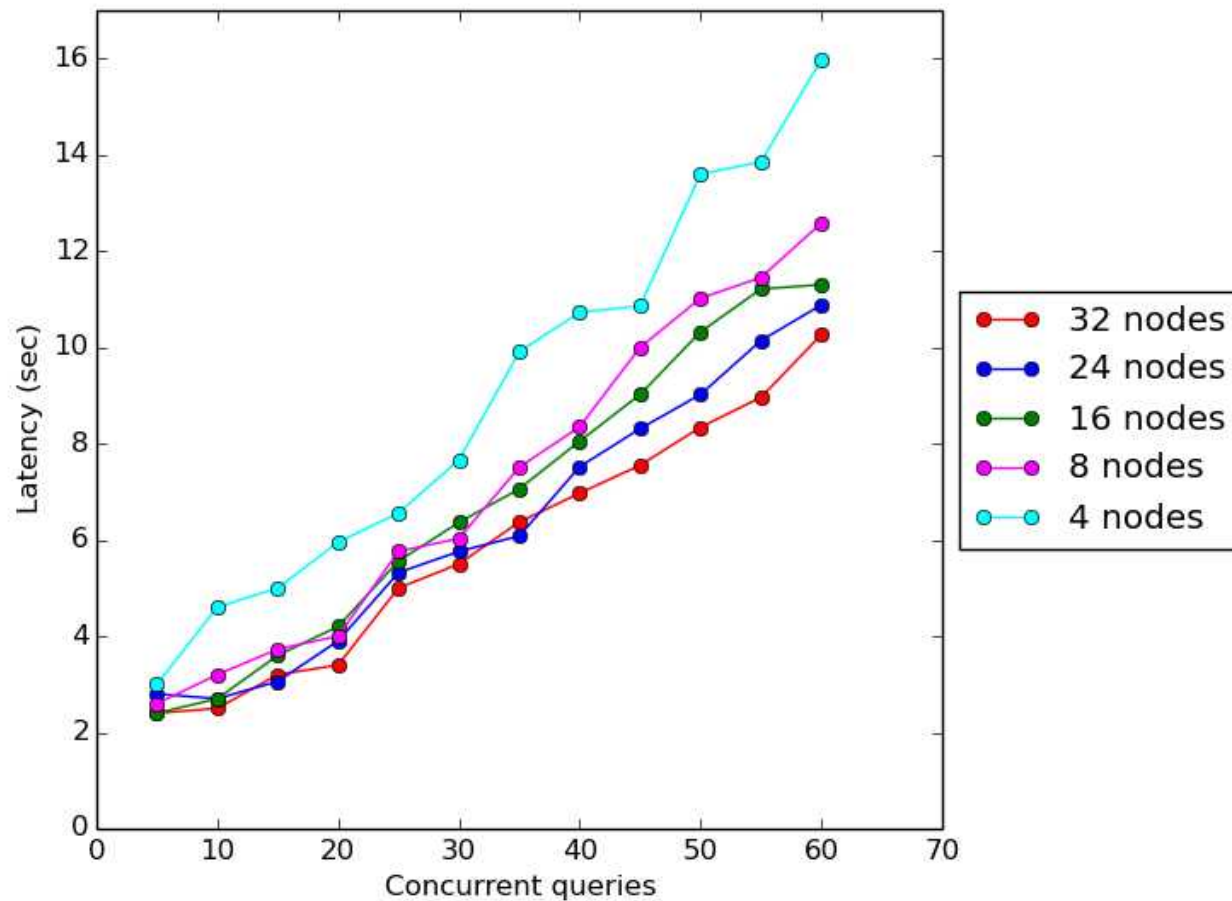
Implemented using HBase coprocessors on data balanced region servers



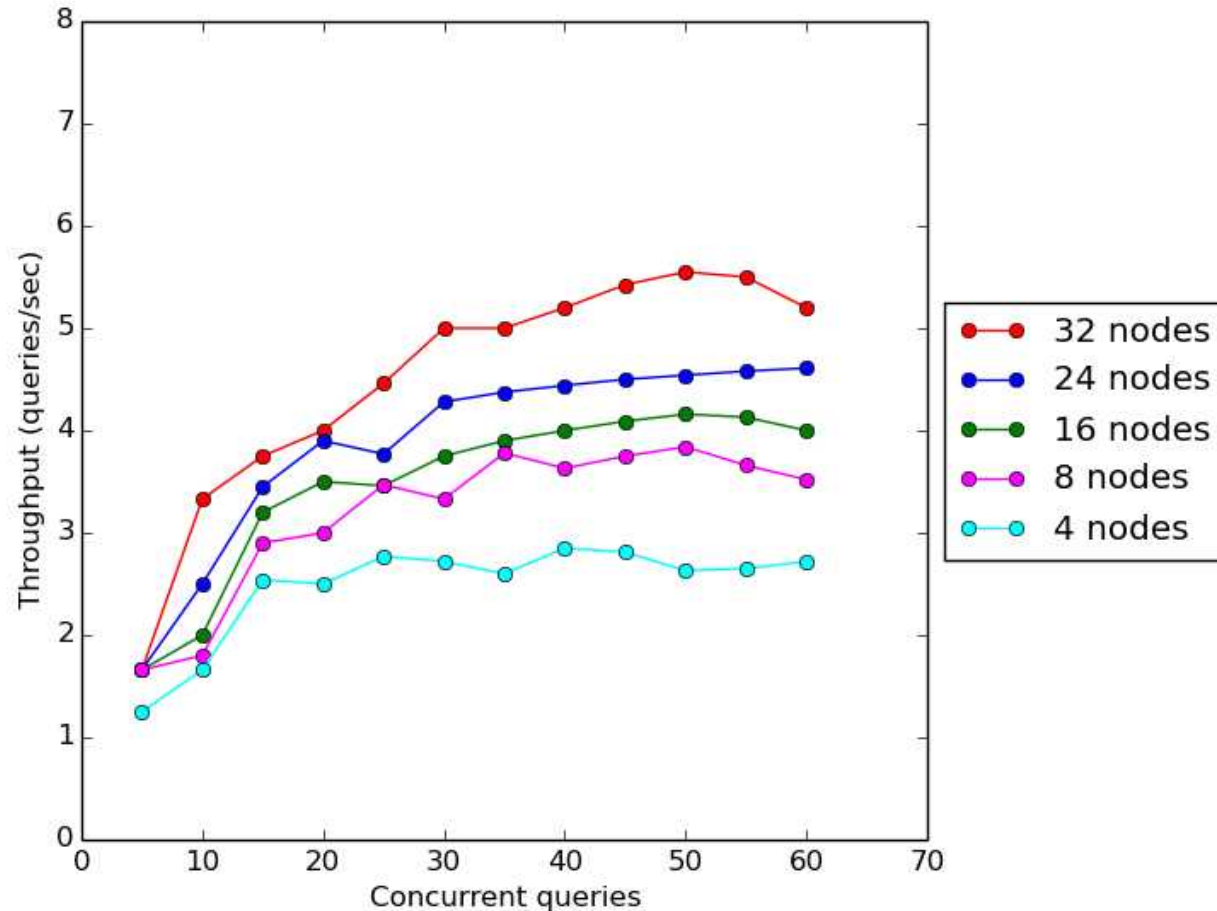
# Workload generation setup



# Scalability Testing



# Scalability Testing



# Conclusion

---

- HBase cluster is scalable for the specific data storage model of the dataset produced by the generator
- HBase provides indeed good performance and data management tools for Big Data social networking services

# Questions

---

