

# Lab 4: Geospatial Query in MongoDB

## 1. Outline

In this lab, you will perform geospatial querying in MongoDB with prepared geo-located Twitter data in Chicago. You will insert data into the MongoDB instance via modifying and running a python script, and perform spatial indexing and geospatial query with a GUI client available in the lab repository in Bitbucket.

## 2. Data

The geo-located Twitter data of Chicago reside in `/gpfs_scratch/geog479/lab4/`

- Copy the folder lab4 to your home directory
- Use the head or tail command to view the data structure
- Each row in the file contains 5 fields (tab separated): user\_id, latitude, longitude, timestamp, message\_content respectively
- Tell me how many lines in file 2014\_07\_04\_chi.txt
- Tell me what is the size of file 2014\_07\_04\_chi.txt

Pop the data into MongoDB with a python script

- The python script is named dbimport.py in the same folder
- Use nano to view the details
- To run the script, Python needs the pymongo library, which is available in Anaconda (a python module in ROGER). You need to load that module before running the script, otherwise it would fail.
- Before you running the script, please modify the script to create a database in MongoDB with the same name as your NetID
- You can also import other dataset from other files in the folder (pay attention to the code).

## 3. Geospatial Database Query

Download the GUI client from the lab repository in Bitbucket

Unzip the file and connect to the database

- Host name: 141.142.168.54 and port (**default: 27017**)

Create spatial index to speed up database querying

- `db.chicago.createIndex( { loc : "2dsphere" } )`

Find the geo-located tweets within the search distances from a specified location

- `db.chicago.find({loc: {$nearSphere: {$geometry: {type : "Point",coordinates : [-87.639160, 41.878628]}},$minDistance: 1000,$maxDistance: 5000}}})`

Find the geo-located tweets within a specified region (polygon)

- `db.chicago.find({loc:{$geoWithin: { $geometry: {type : "Polygon", coordinates: [ [ [-87.639546, 41.878053], [87.639643, 41.879506], [-87.638602,41.879363], [-87.638270,41.878045], [-87.639546, 41.878053]]]]}}})`
- Go to Google Maps and pick up a pair of coordinates of your own place-of-interest and perform the query illustrated above.