CSE-511: Data Processing at Scale Portfolio Report (July 2024)

Yousef Majadbi, Computer Science and Engineering Program, Ira A. Fulton Schools of Engineering,
Arizona State University, ymajadbi@asu.edu

***Abstract*—This report presents the work completed for CSE 511 "Data Processing at Scale" at Arizona State University, focusing on two main projects: NoSQL project and Hot Spot Analysis project. The report details solutions, results, contributions, and skills and knowledge acquired.**

***Index Terms*— Apache Spark, Getis-ord, haversine, Hot Spot Analysis, Latitude, Longitude, NoSQL, Python, Scala, Spatial Analysis, UnQLite.**

I. Introduction

This report covers two significant projects completed as part of the course CSE 511 "Data processing at scale" taken in Summer 2024. These projects demonstrate the application of advanced data processing techniques and spatial analysis using NoSQL databases and Apache Spark, respectively. The projects highlight the practical skills and knowledge gained during the course.

The NoSQL project provided hands-on experience with NoSQL databases, focusing on retrieving data using Python. The project involved implementing two key functions to filter businesses based on city and location, demonstrating the flexibility and efficiency of NoSQL databases compared to traditional relational databases.

And the Hot Spot Analysis project involved performing spatial hot spot analysis on a large dataset of NYC taxi trips using Apache Spark and Scala, and creating a solution that can extract crucial data from the provided dataset which can then be used to make operational and strategic choices.

II. Explanation of the Solution

A. Project 1 – Function 1

The first function – *FindBusinessBasedOnCity* used to fetch all businesses from NoSQL collection, find the businesses present in the city provided, and save the results to the provided location. Then for each founded business I stored the name, full address, city, and state of the business in the following format. Each line of the saved file will contain: Name$FullAddress$City$State.

B. Project 1 – Function 2

Second function – *FindBusinessBasedOnLocation* searches businesses from a given NoSQL collection to find the name of all the businesses present within a maximum distance from a given location and save them to a given location. Then for each founded business I stored the name of the business only. The formula for calculating the distance between two geographic coordinates was provided to us as the haversine formula. I used this formula in a helper function to calculate the distance between two coordinates. The helper function would take in the latitudes and longitudes of both coordinates and return the distance between the coordinates according to the haversine formula:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \operatorname{atan} 2\left(\sqrt{a}, \sqrt{(1-a)}\right)$$

$$d = R \cdot c$$

where φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km)

C. Project 2 – Hot Zone Analysis

This task involved performing a range join operation on a rectangle dataset and a point dataset. For each rectangle, the number of points located within the rectangle will be obtained. The hotter rectangle means that it includes more points. The ST_Contains function was implemented to check if a point lies within a given rectangle.

D. Project 2 – Hot Cell Analysis

This task focused on applying spatial statistics to spatio-temporal big data to identify statistically significant spatial hot spots using Apache Spark. The steps required before calculating the z-score included:

1. Loading the data and determining the cell coordinates (x, y, z).
2. Calculating the number of points in each cell.
3. Computing the mean and standard deviation of the point counts.
4. Calculating the Getis_Ord stats for each cell, based on below formula:

$$G_i^* = \frac{\sum_{j=1}^{n} w_{i,j} x_j - \bar{X} \sum_{j=1}^{n} w_{i,j}}{S \sqrt{\frac{\left[n \sum_{j=1}^{n} w_{i,j}^2 - \left(\sum_{j=1}^{n} w_{i,j}\right)^2\right]}{(n-1)}}}$$

where x_i is the attribute value for cell $j, w_{i,j}$ is the spatial weight between cell *i* and *j*, *n* is equal to the total number of cells, and:

$$\bar{X} = \frac{\sum_{j=1}^{n} x_j}{n}$$

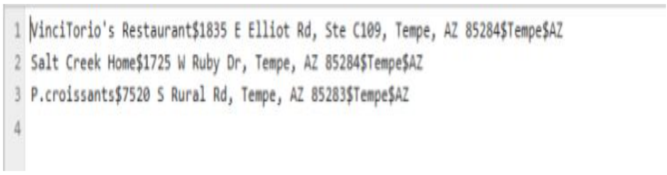
$$S = \sqrt{\frac{\sum_{j=1}^{n} x_j^2}{n} - (\bar{X})^2}$$

III. Results

A. *Project 1 – Function 1*

The FindBusinessBasedOnCity function generated a list of businesses located within the specified city. The output file contained the full address, city and state of the business of each found business in the format Name$FullAddress$City$State. This demonstrated the efficiency of NoSQL databases in handling large datasets with flexible querying capabilities.

The function was evaluated against a sample NoSQL database provided for this project. The response time for querying the database and generating the results was notably efficient, showcasing the high performance of NoSQL systems for read-heavy operations. Below is a screenshot of the output file:



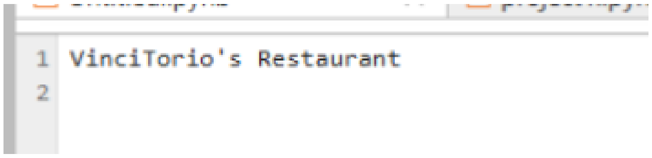
```
VinciTorio's Restaurant$1835 E Elliot Rd, Ste C109, Tempe, AZ 85284$Tempe$AZ
Salt Creek Home$1725 W Ruby Dr, Tempe, AZ 85284$Tempe$AZ
P.croissants$7520 S Rural Rd, Tempe, AZ 85283$Tempe$AZ

```

Findings:

- The spatial distribution of pickups showed a different pattern based on the location coordinates.
- We can do extra filters on this function, to retrieve more specified business category in a particular city, for example we can search for restaurants specializing in Pizza.

B. *Project 1 – Function 2*

The FindBusinessBasedOnLocation function produced a list of businesses within a maximum distance from a given location. The output file contained the name of the business only. This function leveraged the Haversine formula to calculate distances between geographic coordinates, ensuring accurate results. The function was evaluated with various locations to validate its accuracy and performance. It effectively identified businesses within the specified range. Below is a screenshot of the output file:



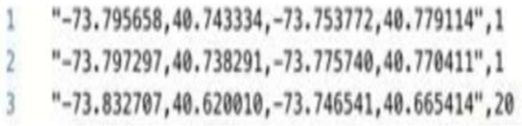
```
VinciTorio's Restaurant

```

Findings:

- The NoSQL database demonstrated flexibility in handling several types of queries, like city-specific and spatial proximity searches.
- We can use this function in various applications to suggest nearby hotels, restaurants, or attractions based on the user's current location and preferred distance range.

C. *Project 2 – Hot Zone Analysis*

After performing the join operation on rectangle datasets and point datasets, I was able to identify the count of points number within each rectangle and generate output file which contains all rectangles (zones) along with their respective point counts, sorted by rectangle string in ascending order. The below results were written to output file after running the ”.jar” program file using testing data that was provided to us:



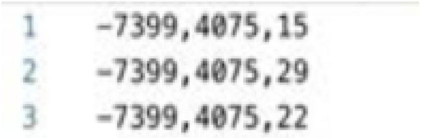
```
"-73.795658,40.743334,-73.753772,40.779114",1
"-73.797297,40.738291,-73.775740,40.770411",1
"-73.832707,40.620010,-73.746541,40.665414",20
```

Findings:

- This analysis revealed several high-activity zones where taxi pickups were significantly concentrated.
- The spatial distribution of pickups showed a different pattern based on the location coordinates.
- This crucial analysis can beused to make operational and strategic choices to advance and improve customer service.

D. *Project 2 – Hot Cell Analysis:*

The Hot Cell Analysis task focused on identifying statistically significant hot spots in the NYC taxi trip dataset using the Getis-Ord G_i^* statistic. Then coordinates of top 50 hottest cells sorted by their G score in a descending order, and saved in text file that was generated after running the .jar program file as shown below:



```
-7399,4075,15
-7399,4075,29
-7399,4075,22
```

Findings:

- Several cells showed a high G-score, indicating significant taxi activity.
- The analysis showed distinct spatial patterns of taxi pickups, which can be observed in various locations.
- This analysis as well can be used to improve the customer service experience when it be given to the client.

IV. Contributions

A. *NoSQL Project*

In the NoSQL project, my primary contribution involved implementing two main functions: *FindBusinessBasedOnCity* and *FindBusinessBasedOnLocation*. These functions were developed to query a NoSQL database using Python and UnQLite.

Function 1 – FindBusinessBasedOnCity: I developed this function to filter businesses based on a given city name. Where function retrieves business name, full address, city and state, and writes them to a text file.

Function 2 – FindBusinessBasedOnLocation: I developed this function to find businesses within a specified distance from a given location. I implemented a helper function to use the Haversine formula to calculate the distance between geographic coordinates, then write the results to a text file.

Throughout the project, I ensured that the functions met the project requirements by evaluating them using the provided sample database.

Achieving this required understanding NoSQL database's structure and the technique to query the collection of NoSQL databases.

B. *Hot Spot Analysis Project*

In the Hot spot analysis project, I contributed by performing spatial hot spot analysis on a large dataset of NYC taxi trip dataset using Apache Spark and Scala. My contributions included setting up the environment, modifying the provided code templates and implementing the required functions for both Hot zone analysis and Hot cell analysis tasks.

Task 1 – Hot Zone Analysis: For this task, I modified the HotzoneUtils.scala and HotzoneAnalysis.scala files to perform join operation on a rectangle dataset and point dataset. I implemented the *ST_Contains* function to check if a point lies within a given rectangle.

Task 2 – Hot Cell Analysis: For this task I modified the HotcellUtils.scala and HotcellAnalysis.scala files to apply spatial statistics to spatio-temporal big data, by calculating the Getis-Ord statistics for each cell to identify the top 50 hottest cells.

My contributions to this project involved understanding of spatial data processing and statistical analysis using Apache Spark.

V. Lessons Learned

A. *NoSQL project*

Through this project I learned the following:

1. Working with NoSQL databases, especially the Document databases.
2. I learned how the data is stored in NoSQL databases in the form of collections which contain documents in JSON form.
3. The difference between NoSQL and SQL in the matter of storing and querying the data.
4. How to use UnQLite in Python to conduct several fundamental operations on NoSQL databases, including connecting to databases, accessing collections, and filtering documents.

B. *Hot Spot Analysis Project:*

Through this project I learned the following:

1. How to use Apache Spark for big data processing, particularly for spatial data processing.
2. I learned how to setup the big data processing environment by installing Hadoop, Apache Spark and Scala on my local Windows machine.
3. I gained experience in performing range join operations, spatial queries, and statistical analysis using Scala.

VI. References

[1] A variety of calculations for latitude/longitude points including the Haversine formula, with the formulas and code fragments for implementing them. Available at: Calculate distance and bearing between two Latitude/Longitude points using haversine formula in JavaScript (movable-type.co.uk)

[2] ACM SIGSPATIAL Cup 2016 for getis-ord statistic. Available at: http://sigspatial2016.sigspatial.org/giscup2016/problem

[3] Fast Python bindings for UnQLite, a lightweight, embedded NoSQL database and JSON document store. Available at: unqlite-python — unqlite-python 0.9.3 documentation

[4] Apache Spark - A Unified engine for large-scale data analytics. Spark 3.4.1. Available at Overview - Spark 3.4.1 Documentation (apache.org)

[5] Learn Scala | Scala Documentation. Available at: https://docs.scala-lang.org/

[6] Hadoop | Apache Hadoop 3.3.6. https://hadoop.apache.org/docs/stable/

[7] ALI DAVOUDIAN, LIU CHEN and MENGCHI LIU, A Survey on NoSQL Stores. Available at: A Survey on NoSQL Stores

[8] Abdul Jabbar, Pervaiz Akhtar, Samir Dani, Real-time big data processing for instantaneous marketing decisions: A problematization

approach. Available at: Real-time big data processing for instantaneous marketing decisions_ A problematization approach (sciencedirectassets.com)