# Apache Spark for Spatial & Hotspot Analyses

Glenn Skawski

Team Members: Dustin Anderson, Shravani Gorenta, Peter Zhang

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
Tempe, Arizona, Maricopa County
{gskawski, dhander1, sqorenta, pzhang81}@asu.edu

## 1 Introduction

The group project was a simulation in the group being contracted by a taxi service (the "client") to develop and run spatial queries on their database containing geographic and real-time location data collected from services rendered. The client's database is large and unstructured, so using a popular Big Data software is required. The database includes data entries defined by latitude, longitude, and temporal attributes. Goal is to extract database data for client to use for operational (day-to-day) and strategic level (long term) decisions.

The first client deliverable is the Spatial Analysis composed of several Spark SQL queries to process point and rectangle data for the client to identify relationships between locations and defined 2D spaces. The second client deliverable is the Hotzone Analysis for the client to categorize distinct areas based on observed taxi volume. The third client deliverable is the Hotcell Analysis for the client to categorize spatio-temporal unit areas based on a statistical analysis of each unit with respect to observed taxi volume, location, and time.

## 2 System Design - Apache Spark

The project was built using Apache Spark which is a framework for processing big data built on top of Hadoop's Distributed File System that can store data across multiples machines for scalability [5]. Apache Spark improves upon Hadoop in terms of performance and flexibility of use. Apache Spark works on Resilient Distributed Datasets (RDDs) which are objects that can be distributed across a cluster of machines [3]. RDDs are hidden from the user in that a program can be designed as if occurring on a local machine and Apache Spark distributing the work using a driver node and worker nodes [6].

## 3 Spatial Analysis

### 3.1 Spatial Range Query

Goal is to extract points located in each rectangle from a set of points and rectangles.

*3.1.1 Input.* A file with each line being a point defined by latitude and longitude attributes. A second file with each line being a rectangle defined by two latitude and two longitude attributes.

*3.1.2 Approach.* [2]

1. Iteratively compare each rectangle and point;
2. Initiate point rectangle comparison to false;
3. Parse string inputs and transform to array of type double;
4. Compares point location to rectangle boundaries. Points located on a boundary are included;
5. Return true if point is within boundaries otherwise return initial false value.

### 3.2 Spatial Distance Query

Goal is to extract point pairs corresponding to points located within a given radial distance from each other.

*3.2.1 Input.* A file with each line being a point having latitude and longitude attributes.

*3.2.2 Approach.* [2]

1. Iteratively compare each point pair
2. Initiate point1 and point2 comparison to false;
3. Parse string inputs and transform to array of type double;
4. Compute the two points Euclidean distance and compare to given distance;
5. Return true if points are less than or equal to given distance otherwise return initial false value.

## 4 Hot Spot Analysis

### 4.1 Hotzone Analysis

Goal is to extract the total number of pickups in each rectangle for a given rectangle set.

*4.1.1 Input.* A file with each line containing several attributes collected from an individual taxi service. A second file with each line being a rectangle defined by two latitude and two longitude attributes.

*4.1.2 Approach.* Same approach as the spatial range query [2].

### 4.2 Hotcell Analysis

Goal is to identify cells with highest pickup volume based on their computed Getis-Ord statistic [1].

*4.2.1 Input.* A file with each line containing several attributes collected from an individual taxi service.

*4.2.2 Assumptions.*

1. Space is a grid of 3 x 3 x 3 cells (27 total) where x, y, and z are represented by latitude, longitude, and time, respectively;
2. Size of each cell is 0.01 degrees x 0.01 degrees x 1 day;
3. Entire test area is that defined by latitude 40.5N – 40.9N and longitude 73.7W – 74.25W; and

4. A neighborhood is a cell's 27 neighbors cells including itself.

*4.2.3 Approach.* [2]

1. Iterate each line of input file to create new view that contains only x (latitude), y (longitude), and z (time) for each row;
2. Multiply each cell's x and y by 100;
3. Create variables to represent test area boundaries;
4. Parse array define rectangles upper latitude, left longitude, lower latitude, and right longitude designations;
5. Create view filtering out cells not within test area boundary;
6. Create view of only distinct cells based on latitude, longitude, and time attributes. Add attribute for each cell that is total number of pickup instances ($x_j$) observed for cell;
7. Calculate mean ($\bar{X}$) of distinct view by the following equation where $n$ is the total number of cells in the test area:

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

8. Calculate standard deviation ($S$) of distinct view by the equation:

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

9. Create method determining each cell's total neighbors (aka weight, $\sum_{j=1}^{n} w_{i,j}$) from cell's location to boundaries;
10. Create view from distinct of all possible combinations of two cells. Includes condition limiting cell pairs to those with x, y, z distances of 1, 0, or -1 relative to each other;
11. Calculate cells' weight and sum pickup instances ($\sum_{j=1}^{n} x_i$) of cells' neighborhood;
12. Calculate cells' Getis-Ord statistic described by the equation [4]:

$$G_i^* = \frac{\sum_{j=1}^{n} w_{i,j} x_j - \bar{X} \sum_{j=1}^{n} w_{i,j}}{S \sqrt{\frac{\left[n \sum_{j=1}^{n} w_{i,j}^2 - \left(\sum_{j=1}^{n} w_{i,j}\right)^2\right]}{n-1}}} \quad (3)$$

13. Sort view in descending order and output first 50 cells.

# 5 Results

## 5.1 Range Query

Figure 1 is the query's output table of points located in the rectangle space given a 2D rectangle and point dataset. Figure 2 is the query's output of rectangle-point tuples for points located in each rectangle space given a point and rectangle.

| _c0|
|---|
|-93.579565,33.205413|
|-93.417285,33.171084|

**Figure 1. Range Query**

| _c0| _c0|
|---|---|
|-93.63173,33.0183...|-93.579565,33.205413|
|-93.63173,33.0183...|-93.417285,33.171084|

**Figure 2. Range Join Query**

## 5.2 Distance Query

Figure 3 is the query's output table of points located within a radial distance from a location given a point dataset, point, and distance. Figure 4 is the query's output table of point pairs for those located within a radial distance from each other given two datasets of points and a distance.

| _c0|
|---|
|-88.331492,32.324142|
|-88.175933,32.360763|

**Figure 3. Distance Query**

| _c0| _c0|
|---|---|
|-88.331492,32.324142|-88.331492,32.324142|
|-88.331492,32.324142|-88.388954,32.357073|

**Figure 4. Distance Join Query**

## 5.3 Hotzone & Hotcell Query

Figure 5 is the query's output table of rectangles and corresponding total number of observed pickups given taxi and rectangle datasets. Figure 6 is the query's output table of cells with the highest Getis-Ord statistic given a taxi dataset.

| rectangle|numOfPoints|
|---|---|
|-73.789411,40.666...| 1|
|-73.793638,40.710...| 1|
|-73.795658,40.743...| 1|

**Figure 5. Hotzone Query**

| x| y| z|
|---|---|---|
|-7399|4076| 23|
|-7401|4071| 26|
|-7397|4079| 12|

**Figure 6. Hotcell Query**

# 6 Contributions

I contributed to the project through two general roles that are team moderator and sharing equal responsibility programming for project milestones. My first contribution was drafting a project charter, communication plan, and schedule for the team which detailed team goals, objectives, roles, responsibilities, milestones deliverables, means of communication, and team members anticipated challenges. To gather information for these initial documents I organized a group meeting to gather members working preferences and scheduling conflicts and setup a shared 8-week calendar that consolidated conflicts, future group meetings, project deadlines, and agreed upon deliverables members were responsible for. The calendar also was a medium to setup meetings for the group to access. The initial schedule was generally followed and used for the duration of the project. Simultaneously I setup a shared group folder containing all the project files, directions, and the groups internal files for visibility of each members work and to avoid duplicating efforts.

After finalizing the group's work methods, I initiated drafting Milestone 1 which included providing a summary of our group's discussion on database applications on the cloud and whether current trends would lead to a change in popularity of relational databases. Milestone 2 was the group's first introduction to Apache Spark and how it will be used for the project. My responsibility at this point was to setup the software and learn how to run the test examples. This was a contribution by having the capability to compile and test other group members code blocks into a single project folder.

The group next started working on Milestone 4 and 5 by breaking them down by what we expected their components to be. The components were then delegated to group members. The result being that two members would collaborate on developing the spatial query and two members would collaborate on the hotspot analysis where I was responsible for the latter portion of the project.

I primarily focused on the hotcell analysis portion. I started by organizing the problem statement into its components for easier conceptualization. Doing so included myself completing the following tasks:

1. Robust definitions of inputs, key words, and micro- (cells) and macro-geometries (3D environment);
2. Meaning, breakdown, and application of statistical functions provided in the problem statement
3. Project assumptions; and
4. Relationships between geometries and attributes and steps for application to code.

The goal of this step was to build the entity-relationship (ER) diagram which is a method to consolidate client requirements, inputs, and problem components into an organized and decipherable format of entities, attributes, and relationships. I communicated the results of this step to the group for feedback on my conclusions and approach. Differing opinions were discussed and applied if the group found it appropriate.

With an ER diagram to work from, I started the process of getting more familiar with the software, languages, and class-provided template in order to build the query logic. The hotcell logic was then worked on as a collaborative effort between the group members and me. The first step in doing so was detailing the expected steps needed to complete the logic where each step could be translated into distinct sections of code. The format of this step can be exemplified in the hotcell approach sections. With the approach defined, creating the hotcell logic was a trial and error effort with more detail than is able to be described given the scope of this report. The result of this effort was the hotcell logic that outputted the desired results and earned a score of 7 out of 8 points from the auto-grader. The draft logic was presented and explained to the group whereby each member worked to identify what part of the logic contributed to the incorrect output. Another member of the team was able to identify and fix the logic to achieve full credit.

Milestone 3 was due around the same time of completing the initial hotcell attempt. I was responsible for drafting and submitting this report upon the groups review. By this time our group had completed Milestone 4 and nearly completed Milestone 5.

My final contribution was to draft the System Documentation Report. The aim being to describe the work completed in a level of detail whereby another programmer could apply our solution with relative ease. It also included describing the group's process that lead to a solution, task distribution, and challenges. I approached this task with the mindset of applying class-provided resources and concepts that would center the report content in terms of communicating the solution to the client which is the context for why the project is to be done. I think this was important to incorporate because of how it ties the project into course content and illustrates a complete lifecycle of a practical database application even though the client portion of the problem was a relative non-factor in the final solution. I then shared the report with the group for feedback and finalized after including group member's amendments.

# 7 Lessons Learned

## 7.1 Project Conceptualization & Reporting

The project provided the opportunity to apply database design methodology which includes the following steps: 1) requirement analysis; 2) conceptual database design (ER diagram); and 3) relational database schema. I learned how to decipher and organize the problem statement from the client's data and requirements. Then develop an ER diagram organizing the raw data into entities, attributes, and relationships. Finally building the logic from the ER diagram in Apache Spark using the Spark SQL language. Related is learning best practices for communicating the project solution to a client. This was through writing the System Documentation Report. The value is that developed product is incomplete without effective communication.

## 7.2 Data Processing Technology

The project included the opportunity to learn and apply knowledge to several software packages, languages, and libraries that I now have experience in and can utilize in the future. This included Apache Spark, Hadoop, Scala, Spark SQL, Java, and SparkSession. Notable to future application is the Apache Spark framework which has widespread use and application in any Big Data context. The additional experience with using Apache Spark and Hadoop's DFS is a vital resource for working with big data where optimizing processing speeds will be expected.

## 7.4 Database Applications

The class emphasized database applications to our everyday experiences. Included is database understandings from the perspective of technical development, databases as a business, and projected future trends. I plan on entering the professional workforce after the MCS program, so this industry knowledge is helpful knowing that I will need to show employers that I am capable of integrating database theory to professional applications. The project added to this by providing experience solving a problem with value that can be understood in an industrial setting.

## 7.5 Team Collaboration

The project was my first experience working on a CS project in a group setting. While I have previous teamwork experience, there are improvement opportunities and facets that I will apply in the future. The experience taught me to be more detailed, persistent, and explicit when discussing the project approach with a group. I found myself wasting efforts that may have been prevented with the groups input. The goal would be to stimulate and evaluate ideas in order to come to a consensus on a robust action plan. Another take away is the need to be adaptable. An individual has little to no control over others or situations, so always be ready and able to adjust one's preferences in order to help the group. This is hardly an insight unique to CS, but with more experience one gains more resources at their disposal to better adapt. I will continue to learn and try new skills to build this capability.

# 8 References


[1] ACM SIGSPATIAL. “Problem Definition.” *ACM SIGSPATIAL GIS Cup 2016*, ACM SIGSPATIAL, 2016, sigspatial2016.sigspatial.org/giscup2016/home.

[2] Dustin Anderson, Shravani Gorenta, Glenn Skawski, and Peter Zhang. 2020. *System Documentation Report*. CSE 511 Data Processing at Scale.

[3] Apache Spark 3.0.0. “Spark Overview.” *Spark Overview*, Apache Spark 3.0.0, 2020, spark.apache.org/docs/latest/index.html.

[4] Ord, J. K., and Arthur Getis. “Local Spatial Autocorrelation Statistics: Distributional Issues and an Application.” *Geographical Analysis*, vol. 27, no. 4, 2010, pp. 286–306., doi:10.1111/j.1538-4632.1995.tb00912.x.

[5] Sarwat, Mohamed, director. *Week 6 - Big Data Tools*. *Coursera*, Arizona State University, 2020, www.coursera.org/learn/cse511/home/week/6.

[6] tutorialspoint. “Spark SQL - Quick Guide.” *Spark SQL Tutorial*, tutorialspoint, 2020, www.tutorialspoint.com/spark_sql/index.htm.