CSE 511 - Data Processing at Scale - Portfolio Report

Arizona State University
Christopher Richard Bilger
ASU ID: **REDACTED**

Abstract - This paper will review the problems, solutions, design considerations, personal contributions, and lessons learned from the project given during the CSE 511 course.

1. Introduction

The projects for CSE 511 Data Processing at Scale started with small, introductory programs that were used to introduce the team members to one another, and also to introduce each team member to the development environment setup, the Scala programming language[3], the Spark big-data analysis tool[4], IntelliJ, and the Python programming language. Through this portfolio report, I will cover the projects completed in this course, my role in completing those projects, and what I have since learned and continue to use from these projects.

2. Solutions

When my team set out to work on this project, we decided to solve phase 1 and phase 2 individually before meeting and discussing our solutions. This allows me to easily show what I did to solve the problems that I encountered in each phase of the project. I will split up my solutions into two sections, one for each phase of the project, and explain in detail my solutions to the problems given.

Phase 1:

For the first phase of the project, I had to create two functions. One of these functions had to return a boolean value of whether or not an **x-y** point resides inside of the boundaries of a set of **2 x-y** points, creating a rectangle. The first step that I did was to convert the single-point string into an **x** variable and a **y** variable. I then converted the longer multi-point string into two separate **x-y** variables, consisting of the corner points of the rectangle. Once the conversions were finished, I checked if both the **x** and the **y** variables resided inside of the **x-y** bounding box, and returned a boolean representation of this value. Below is an image of the steps, in order, outlined above.

```
def ST_Contains(queryRectangle: String, pointString: String): Boolean = {
    val point = pointString.split(",")
    var x = point(0).toDouble
    var y = point(1).toDouble

    val boundaries = queryRectangle.split(",")
    var x1 = boundaries(0).toDouble
    var y1 = boundaries(1).toDouble
    var x2 = boundaries(2).toDouble
    var y2 = boundaries(3).toDouble

    if (x >= x1 && x <= x2 && y >= y1 && y <= y2)
        return true

    if (x >= x2 && x <= x1 && y >= y2 && y <= y1)
        return true

    return false
}
```

Fig. 1. Phase 1, function 1 code implementation

The second function had to return a boolean value of whether or not the Euclidean distance[2] between two points was less than or equal to a given distance **d**. The first step that I completed to solve this problem was to convert the two point-strings into **x** and **y** variables, respectively. I then calculated the Euclidean distance between these two points, before checking if this distance is less than or equal to the given distance **d**. I returned the boolean value of this logical comparison. Below is an image of the above steps, in order.

```
def ST_Within(pointString1: String, pointString2: String, distance: Double): Boolean = {
    val point1 = pointString1.split(",")
    var x1 = point1(0).toDouble
    var y1 = point1(1).toDouble

    val point2 = pointString2.split(",")
    var x2 = point2(0).toDouble
    var y2 = point2(1).toDouble

    var d = sqrt(pow(x1 - x2, 2) + pow(y1 - y2, 2))
    if (d <= distance)
        return true

    return false
}
```

Fig. 2. Phase 1, function 2 code implementation

Phase 2:

The second phase of the project consisted of joining the contents of the first phase along with SQL database

querying as well as some mathematical analysis to find the solution to the problem given. Below I will outline the steps that I took to solve this problem.

Entrance.scala:
I added our group number to the "appName" function-call parameters.

HotzoneUtils.scala:
I copied over the "ST_Contains" function that I wrote in phase 1 of this project, into the "HotzoneUtils" object.

HotzoneAnalysis.scala:
I realized that the "runHotZoneAnalysis" function definition was only missing the correct return value of the hot zone data frame. I took the "join result" data frame, grouped by the "rectangle" keyword string, and then iterated over this data frame to find and sort only by the "rectangle" keyword string.

HotcellUtils.scala:
I added a calculation function to find the number of adjacent hot cells for a given **x-y-z** coordinate position. To solve this, I had a simple counter mechanism that started at an initial value of 0 and would then increment by 1 if and only if the given **x-y-z** coordinates lie on either the minimum **x-y-z** value or maximum **x-y-z** value. I then iterated through the possible counter values and returned the corresponding integer value.

I also added a function that calculates the Getis-Ord score, or the amount of point clustering that is occurring, in a given area. This function takes in numerous parameters; such as the total number of cells, total number of hot cells, x-y-z coordinates, etc. and outputs a double value corresponding to the value given from the following formula[1]:

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

Fig. 3. Getis-Ord formula

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

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

Fig. 4. Two sub-components of the formula are in Fig. 3.

HotcellAnalysis.scala:
After writing one too many solutions to this problem, which I see as the biggest problem to be solved for this project, and still having errors, I re-designed my solution from the ground up. The function "runHotcellAnalysis" was missing the database querying functionality so that is was I decided to add first. If anything, I would be able to solve that problem and, at the very least, obtained data from the database which can then be converted into the information that I am looking for. That is, the top 50 pickup coordinates are aggregated and sorted by their Getis-Ord score. Once I solved the problem of being able to connect to and query the database, I calculated the Getis-Ord score from the previously obtained database query results. I solved my initial problem by re-thinking the function as a database query which I could then convert and calculate the necessary information from. I, like many others, had another problem that needed to be tackled. When I ordered by descending Getis-Ord score I was still getting incorrect results on the Coursera AutoGrader. I noticed that this was correctable by explicitly ordering all of the results by

not only descending the Getis-Ord score, but also by their x, y, and z coordinates.

3. Results

Throughout the time that I spent working on the projects that are covered in this portfolio report, I came across numerous findings that I would classify as intriguing and useful for would-be students that might also take this course. I will split them up into two sections; the first section is my findings from the first half of the project, which is up to and includes the assignment labeled "Project Milestone 4", and the second section will cover the portions of the project after "Project Milestone 4".

Section 1:
I found the project discussions, especially the discussion on SQL versus NoSQL database, to be incredibly insightful concerning big-data processing and operations on big-data as the scale of the data drastically increases. For the "Project Milestone 4" assignment, which was completed by each team member individually before our meeting and discussion of results, I found that this project had large-encompassing use-cases in the real world. I tend to look at how a project can directly relate to both my studies as well as my future career development. If others go into this project with that same sense of open-mindedness and overall interest in the subject matter being taught (Data Processing at Scale), then I think that this project will help to show to them how many uses spatial queries have in both the real world and the theoretical world.

Section 2:
This section covers one portion of the project and that is "Project Milestone 5". At face value, this part of the project appears quite daunting, but I think that for newcomers to the Scala and Spark world, this is an excellent test of one's ability to pivot and learn something new. I found the Hot Zone Analysis very interesting. I learned that it is possible to calculate the relative scale, or "hotness" in this case, of individual 2-dimensional rectangles using Spark and a small amount of common SQL. Again, as I stated above in section 1, I found that when I went into this part of the project with an open mind, I was able to learn more about the underlying structure of the data and how it can directly influence everyday applications.

4. Contributions

I think that my impact on the overall group project was fairly substantial. As explained above, each team member worked on each assignment individually until we each had a solution to the problem at hand. We then met up and created Zoom conference calls to share our findings and then to submit the group portions of the project. My contributions span each portion of the group project. Below is a detailed listing of my contributions towards the successful and timely completion of each project assignment.

Project Milestone 1:
This is a shorter section of the group project; however, I aided in summarizing my thoughts, as well as the group discussion, into the submitted file. I joined the group discussion open-minded. I shared my insights and learned from my team members through their anecdotes.

Project Milestone 4:
I positively influenced the source code that my team ended up submitting for the assignment and taught my team members what I had learned about the Scala programming language while completing this assignment on my own.

Project Milestone 5 - Source Code:
I wrote approximately half of the "README.pdf" file that was submitted along with the source code for this assignment. I also wrote approximately half of the source code, tested the final group submission source code, and compiled the final "zip" file that was submitted for this portion of the group project.

Project Milestone 5 - Systems Documentation Report:
For this portion of the group project, each team member took one section of the report, wrote that section, and then we compiled the individual sections before submitting the final draft. My portion of this assignment to write was the "Design Solution & Methodology - Phase 1" section. Along with writing this section of the Systems Documentation Report, I also drafted the overall structure of the report as well as finalized the contents of the report.

5. Lessons Learned

My work on this project greatly contributed to my computer science and software engineering knowledge and skills. I learned an entirely new programming language, Scala, which I had heard of in the past but did not expect nor anticipate ever learning. I find this very rewarding because I want to continually learn and evolve my skill sets in the field of computer science. My day-to-day work is in the web development space, so using the full-fledged IDE IntelliJ was a newly acquired skill that I will immediately transfer into my career. Before taking this course, I had very limited knowledge of large-scale data and the processing of large-scale data. Now that this course is wrapping up, however, I can proudly say that my subject matter knowledge of data processing, large-scale operations, SQL, NoSQL, etc. is greatly improved and I am very confident that I will be able to use this knowledge going forward in both my academic as well as my professional careers.

Team #7:
Christopher Bilger
Benjamin Parrish
Balaji Radhakrishnan
Ashraf Sayyad
Chirag Sindhwani
Jebaraj Vasudevan

References

[1] “ACM SIGSPATIAL Cup 2016.” ACM SIGSPATIAL GIS Cup 2016, ACM SIGSPACIAL, sigspatial2016.sigspatial.org/giscup2016/problem.

[2] “Euclidean Distance.” ROSALIND, rosalind.info/glossary/euclidean-distance/.

[3] “The Scala Programming Language.” Scala-Lang, www.scala-lang.org/.

[4] Apache Spark™ - Unified Analytics Engine for Big Data, spark.apache.org/.