DS503 Big Data Management

Project 1

Tianyu Li(tli) & Xiaosong Wen (xwen2)

There are Five Folders in our zip file: DataGeneration, Problem1, Problem2, Problem3, and Problem4

1. Data Generation

The DataGeneration folder contains the Point and Rectangle class, which are the data structure we need, and the Generator class, which is used to generate `11000000 points, 70000000 rectangles and 100 k centroids used in kMeans.`

Use the command:

```
hadoop fs -put ~/project2/*.csv input
```

To uploading the csv files into Hadoop file system

2. Problem 1

The Problem1 folder contains the mr job for SpatialJoin.

To run this job, it need *input/*Points.csv, input/Rectangles.csv, output/Problem1, x1,y1,x2,y2

If no window input, the job will work on the whole data set.

We have individual mappers for point and rectangle, separating them into 10*10 blocks. For one point, it will only show in at most one block. For each rectangle, it will show in all blocks it covers.

3. Problem 2

The Problem2 folder contains the mr job for kMeans.

To run this job, it need input/Points.csv input/kCentroids.csv, output/kMean, k

K can be any int between 10 to 100, inclusive. If K is not given it will be 100 by default.

The kCentroids.csv contains 100 random points on the map. With given k, the job will read first k points for the first round centroids.

The mapper will assign the point (value) to the closest centroid (key), the combiner will calculate the local sum, the reducer will calculate the sum and then the new centroid.

4. Problem 3

To run this job, first upload the json file to input folder in hdfs

Then export the package “problem3” as a jar file and use hadoop jar to execute. No parameters required.

5. Problem 4

This job use input/Points.csv generated by data generation part.

To run this job, export the package “problem4” as a jar file and use hadoop jar to execute. Two parameters are radius and number of neighbors.