CSE 512:

DISTRIBUTED AND PARALLEL DATABASE SYSTEMS

Phase 3- Report

Group TODO:

Anjani Sai Kumar Chatla	1211029375	achatla@asu.edu
Jose Eusebio	1204772396	jeusebio@asu.edu
Venkata Krishna Bandla	1211173740	vbandla@asu.edu

Summary

The goal of this algorithm is to calculate hotspots from a dataset of New York City taxi trip records from January 2015. This algorithm contains two sets of map-reduce functions. The first instance of map-reduce will aggregate the taxi trip data into a spatial grid. The second map-reduce instance uses the trip data to calculate the Getis-Ord statistic for each block.

Algorithm

1. Map-Reduce 1: Aggregating the trip events

   Input: CSV file containing NYC Taxi data from January 2015

   Map: Parse the input, creates a key-value pair for each event instance. Keys represent spatial-temporal blocks of size: 1 day x .01° latitude x .01° longitude. Each key will be created using a combination of the time, latitude, and longitude of the block that the event occurred in. The value in each key-value pair will be 1.

   Reduce: Get the total amount of events that occurred in each block by obtaining the sum of all the values for each key.

   Output: list of spatial-temporal blocks and the number of events that occurred in the block

2. Map-Reduce 2: Computing Mean X

   Input: Map-Reduce 1 output

   Map: Get the values from each input.

   Reduce: Add up all the values

   Output: Return the sum of the reduce divided by N

3. Map-Reduce 3: Computing S

   Input: Map-Reduce 1 output

   Map: The the squared values from each input

   Reduce: Sum up the values from map

   Output: Calculate S using the sum of squared values, N, and the Mean X

4. Map-Reduce 4: Computing Getis-Ord statistic for each block

   Input: Map-Reduce 1 output

   Map: Create a global 3-d list of blocks and associated events. Creates a key using the time, latitude, and longitude of each block.

   Reduce: For each block, get the neighbors and associated number of events. Using this data, compute the Getis-Ord.

   Output: list of Getis-Ord statistic for each block.

5. Sorting step: Find the top 50 Getis-Ord values by swapping the key and value found in Map-Reduce 4 and then using the function: SortByKey().

   Helper Functions:

   ParseInp(): reads in the CSV values and converts it to a String that is used as a Key

   boundarycheck(): checks to see if given coordinates are within the boundaries of the problem

   getNeighbors(): returns a list of the 26 neighbors of a given spatial-temporal coordinate

6. Class Record to hold the latitude, longitude and time_stamp of cells and methods of the method objects