GeoSpatial Data and Spatio Temporal Analysis

Balaji Chandrasekaran
School of Computing, Informatics and Decision Systems
Arizona State University
Tempe, Arizona
bchandr6@asu.edu

***Abstract*—The project comprises of several phases. The initial two phases constituted of familiarizing GeoSpark, the implementation of GeoSpark for GeoSpatial Objects and running a statistical analysis of the distributed techniques such as Spatial Range Query, Spatial KNN Query, Spatial Join Query with and without the join of PointRDD using R-tree or Grid. The analysis was done using Ganglia and the system constituted of Hadoop Cluster that was running spark shell with Geo-Spark jar. The final phase of the project constituted of Spatio-Temporal analysis of a subset of NYC Yellow taxi cab dataset to compute the top 50 hotspots in the given envelope using Getis-Ord statistic.**

Keywords—Hadoop; Spark; GeoSpark; Getis-Ord; R-tree; Ganglia; Spatio-Temporal Analysis;

I. INTRODUCTION

Big Data comprises of extremely large data sets that could be used for analyzing, which results in identifying several meaningful patterns and trends. This process of studying huge amount of legacy data is of prime most significance for predicting the current and future trends. Most of the large data sets are constituted into clusters which helps them in maintaining them and processing the same. Even though there are plethora of tools available for monitoring and maintaining the clusters almost all organizations fail to hit the balance which would result in efficient and effective way of processing big data clusters and obtain desired result. This project deals with providing a hands-on experience of the Hadoop and Spark clusters, understanding the concepts of MapReduce, performance variations due to R-tree index and grid. The project also provides us with a real-world application of Apache Spark and Hadoop clusters for distributed computing with the help of GISCUP 2016 problem solving. [2]

II. PROJECT DESCRIPTION

A. Phase I: System Implementation Plan/System Prototype.

This phase involved the process of setting up of the Hadoop clusters and understand the System behavior by doing several functionalities.

1. Setup of Hadoop Clusters and load the GeoSpark jar into Apache Spark Scala shell.
2. Operations to be done:
    a. Creation of the PointRDD.
    b. Spatial Range Query: Query the PointRDD with and without building R-tree index using given input.
    c. Spatial KNN Query: Query the PointRDD with and without the R-tree index using given input.
    d. Spatial Join Query: Creation of rectangleRDD and use the same for joining pointRDD with the help of equal grid and with/without R-tree Index, R-tree grid and R-tree Index.

B. Phase II: System Demonstration/Experimental Evaluation and Analysis of Results.

There were two halves to this phase. First half was to implement a JAVA function using the Cartesian product algorithm for the purpose of Spatial Join Query. For the same use the GeoSpark Spatial Range Query that is already present in GeoSpark.

The other half was to compare the various functionality that we did in phase I and provide a valid reasoning for the differences that was observed between them using Ganglia which is a cluster monitoring tool and based upon the statistics of Execution time, Average memory, Average CPU utilization of the cluster.

C. Phase III:GISCUP 2016. [2]

In this phase, we took the problem that was provided in GISCUP 2016 competition which was a real-world application of distributed computing. The main objective of the problem was to find the top 50 hotspots of the given NYC taxi cab dataset using the Getis-Ord correlation.

This was the final phase and one of the prime most phase of the project. As we discussed earlier this phase gives us insight on how big data is used in real world and how it can be used to create a better living. The solution of this phase comprises of various application which are from business oriented to elevate the living of public by using it in Urban planning, business startup, transportation management etc. Hotspots are nothing but locations in the given envelope (boundary under consideration) which constitutes of greater traffic/activity. The data involved in

this phase is both temporal (periodic) and as well as spatial which means there would have to be a massive dataset for computation. We use the MapReduce capabilities of Hadoop and Spark Distributed-ness to resolve the problem.

III. Result Description

A. *Phase I: System Implementation Plan/System Prototype.*

The phase I was the initial phase whose main objective was to get to know the Hadoop environment. In this phase, we setup Hadoop and then Apache Spark on it and performed the operations mentioned in the problem statement in previous section. The setup constituted of 3 physical machines one being master and the other two being slaves running Ubuntu, Hadoop 2.6.4 and Spark 2.0.1. The result of the phase was a demo video which explains and goes through the process of the initial phase and the outcome was knowledge on how to run Hadoop clusters and then Apache spark on them and performing operations on spatial data in the environment that has been setup. [4]

TABLE I. Cluster Setup

Cluster Setup			
System	**Memory**	**CPU**	**Cores**
Master	2. 9 GB	Intel i5	2
Worker1	6.7 GB	Intel i7	4
Worker2	6.7 GB	Intel i5	2

B. *Phase II: System Demonstration/Experimental Evaluation and Analysis of Results.*

The phase II was an extension of the previous phase. The objective of this phase was divided into two, first objective being to understand in depth the several variations of spatial operation that were carried out on the environmental setup in previous phase. This also comprise of a way to justify the deviations in variations of same operation done in previous phase, for this objective we choose Ganglia as the monitoring tool for Hadoop cluster monitoring. We studied the variations using several metrics like memory and CPU usage, execution time and presented our conclusion in a detailed manner. One of the prime observation is that — Queries those use indexing is faster compared to the ones that does not use indexing, but they consume more memory and nested loop queries with grid partitioning are faster than Cartesian join queries. The other Objective of the phase was to implement the spatial join query for which we used simple Cartesian product algorithm present and GeoSpark API.

C. *Phase III:GISCUP 2016. [2]*

The phase III problem has been already discussed in an elaborate fashion. For us to implement a solution to the given problem we considered several ways of solving the problem. The data that was given with the problem set was too huge to be processed due to the limited infrastructure available, so we filtered it down to January 2015 – 31 days. Getis-Ord was used since it is the most standard and trusted one which is used for denoting patterns in spatial type data clusters with the help of z-score that it generates. The entire envelope that was given for consideration can be divided into smaller regions of unit length which is a region of pickup or drop operation. These smaller regions can be represented as cells. We then use these cells and identify the top 50 hotspots with the help of total number of pickup operations carried out inside each cell.

We use the Getis-Ord statistic to calculate the score for each of the cell. Which is then used for identifying the top 50 hotspots.

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

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

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

G_i^* – Getis-Ord, $\bar{x}$ – mean of cell scores, S – standard deviation, n – no of cells, x_i – value of cell, $W_{i,j}$ – neighbor parameter of cell.

Fig. 1. Getis-Ord Equations.

After doing in depth research we choose to follow the approach that has been mentioned in the following paper "Spatio temporal hotspot computation on Apache Spark" by Paras Mehta, Christian Windolf, Anges Voisard. [3] The algorithm that we used consisted of four main steps –

1) Data Pre-Processing: In this we load the given data into HDFS by processing the CSV file and removing unwanted columns i.e. fields and removing the data which are outside the given envelope (40.5, –40.9, -74.25, -73.7) i.e. outliers.

2) Cell Generation: In this step, we created the cells and created a mapping that could be used for the Getis-Ord calculation for every cell based on a naïve approach using the neighborhood cells. Each of the cell created was typically 0.01 latitude high and 0.01 longitude wide.

3) Score Calculation: For score calculation, we do a MapReduce. Each cell that is under consideration for which Getis-Ord need to be calculated can be surrounded by either 26/17/11 neighboring cells depending upon whether it is inside/boundary/corner respectively. The Map phase consists of assigning a value of 1 and key which consists of coordinates of Spatio-temporal type to the cell that

localizes each point of activity in spatial and temporal planes. The Reduce phase comprises of mapping key with value in which value is consolidated cell score for all the cells. The various partitions are unaware of each other's data as this is done inside each of the RDD.

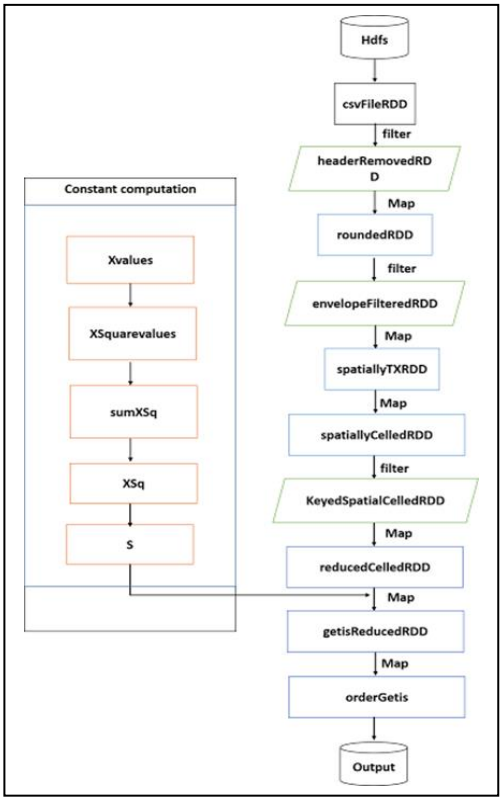

Fig. 2. Flow Chart.

4) We then used a HashMap and stored <KEY, VALUE>:<Cell-Value, Getis-Ord Score>. The HashMap was then sorted into a list with respect to the Getis-Ord Score, this is the shuffle phase of the solution. Then we retrieve and provide the top 50 hotspots from the list which are the required ones.

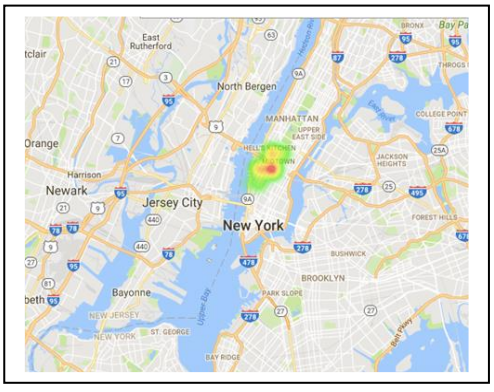

Fig. 3. Heat-Map visualization of top 50 Hotspots.

IV. Contribution

A. Cluster Setup

One of the machine in the cluster was setup and maintained by me. My Laptop enacted the role of Worker1. The process of setting up Hadoop and extending it to run Apache Spark was done by me. Apart from that I was also responsible for the process of integrating it with the master.

B. Video

Video management was done by me. I was responsible for recording and editing the video which was the outcome of phase I.

C. Development

I was responsible for the development of the spatial range query with/without indexing and the spatial KNN query with indexing. This provided me with great insight into spatial operations in Big Data. When it comes to phase III, the process of data preprocessing and cell generation were my responsibility.

D. Algorithm

The solution was constructed after careful consideration of several algorithms/techniques available. I made sure that I have full knowledge of the algorithm specified in "Spatio temporal hotspot computation on Apache Spark" [3] and about Getis-Ord statistic. [7]

E. Ganglia

When it comes to ganglia monitoring I was responsible for collecting data and necessary parameters about my machine (Worker1). This data was then used for constructing the report for phase II, which constituted of comparing the different types of operations that were carried out in previous phase.

F. Report

I was responsible for spatial mapping of hotspots, flowchart, tables, bar graphs, references section, abstract section and introduction section.

V. Knowledge Acquired

A. Skills

Hadoop, Apache Spark, MapReduce, Spatial Operations (Queries – join, range, KNN), Ganglia, HDFS, Java, Scala, GeoSpatial API, R-tree, RDD – Resilient Distributed Datasets, Getis-Ord statistic.

The project also provided us with a practical and real world application of Hadoop – Spark Clusters and MapReduce. The application provided us insight on how important and un-separable Big Data is in real life scenario. It was really fascinating to see such an example which provided us a base on how Hadoop could be used in several real-life scenarios for the betterment of life.

VI. TEAM MEMBERS

Aravind Rajendran, Purushotham Kaushik Swaminathan, Suresh Gururajan, Thamizh Arasan Rajendran.

REFERENCES

[1] Ord, J.K. and A. Getis. 1995. "Local Spatial Autocorrelation Statistics: Distributional Issues and an Application" in Geographical Analysis 27(4).

[2] ACM SIGSPATIAL Cup 2016, Problem Definition (Online): http://sigspatial2016.sigspatial.org/giscup2016/problem Accessed: 03/24/2017

[3] Mehta, Paras, Christian Windolf, and Agnès Voisard. "Spatio-Temporal Hotspot Computation on Apache Spark (GIS Cup)."

[4] Group 25 - DDS Project Phase 1, submission link: https://www.youtube.com/watch?v=ShfpMlbMFYo

[5] Jia Yu, Jinxuan Wu, Mohamed Sarwat. "GeoSpark: A Cluster Computing Framework for Processing Large-Scale Spatial Data". (short paper) In Proceeding of the ACM International Conference on Advances in Geographic Information Systems ACM SIGSPATIAL GIS 2015, Seattle, WA, USA November 2015

[6] The Scala Programming Language, https://www.scala-lang.org/

[7] Getis Ord Statistic for Geospatial operations: https://en.wikipedia.org/wiki/Geospatial_analysis

[8] Data Serialization (Online), accessed 03/24/2017, https://spark.apache.org/docs/latest/tuning.html#data-serialization

[9] Zaharia, Matei, et al. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.