Spatial Data Science & Engineering

Assignment 1

Maximum points Possible – 10

The required task is to understand and implement two tasks: i) performing geometrical operations such as calculating the area of minimum bounding rectangle of polygons ii) finding the maximum element among each band of GeoTIFF images.

Required Tasks

Part 1 (Finding Areas of MBR and Convex Hull)

Task: A code template has been given under folder Part-1 which loads a shape file into spatial RDD. The shape file has a geometry column which contains geometry objects such as polygons and multipolygons. You need to find the area of the MBR of each geometry object. In order to do this, you need to change only one file in the project under folder Part-1. Open src/main/scala/SpatialOP.scala and complete the parts where you can see TODO.

Setting Up Hadoop and Apache Spark Test Environment:

The following setup instructions are specific to Ubuntu operating system:

- Install Java version >= 1.8 and set up the JAVA_HOME environment variable. If you don't know how to set up the JAVA_HOME environment variable, follow the link: https://askubuntu.com/questions/175514/how-to-set-java-home-for-java. To check whether set up is done, run the command *echo $JAVA_HOME* on your command line. You should see the path.
- Download Hadoop-2.7.7 from the link: https://archive.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz and extract it to your desired location. Now, you need to set up the HADOOP_HOME environment variable. Run the command *sudo nano ~/.bashrc* on the command line. Copy the statement *export HADOOP_HOME=path to the folder hadoop-2.7.7* in the opened file. Save and close the file. Run the command *source ~/.bashrc* and check the correctness of the setup with *echo $HADOOP_HOME* command.
- Download Spark-3.0.3 from the link: https://archive.apache.org/dist/spark/spark-3.0.3/spark-3.0.3-bin-hadoop2.7.tgz and extract it to your desired location. Now, you need to set up the SPARK_HOME environment variable. Run the command *sudo nano ~/.bashrc* on the command line. Copy the statement *export SPARK_HOME=path to the folder spark-3.0.3-bin-hadoop2.7* in the opened file. Save and close the file. Run the command *source ~/.bashrc* and check the correctness of the setup with *echo $SPARK_HOME* command. If you browse the path to the SPARK_HOME, you should see *spark-submit* under the *bin* folder which is required for your testing.

How to submit your code to Spark:

- Go to project root folder
- Run *sbt clean assembly*. You may need to install sbt in order to run this command.
- Find the packaged jar in "./target/scala-2.12/Spatial-Operations-assembly-0.1.jar"
- Now, you can run the jar on Spark with spark-submit command. If you already have set up the Hadoop and Spark test environment, you should be able to run the spark-submit command from your command line. Submit the jar to Spark using Spark command "$SPARK_HOME/bin/spark-submit".
- Spark submit command for testing get-mbr: “$SPARK_HOME /bin/spark-submit PATH_TO_JAR_ Spatial-Operations-assembly-0.1.jar output_path get-mbr PATH_TO_FOLDER_OF_SHAPE_FILE”

How to debug your code in IDE:

- Use IntelliJ Idea with Scala plug-in or any other Scala IDE.
- In some cases, you may need to go to "build.sbt" file and change the value of dependencyScope from “provided” to “compile” in order to debug your code in IDE
- Run your code in IDE
- **You must revert** dependencyScope from “compile” to “provided” **and recompile your code before use spark-submit!!!**

Part 2 (Maximum of Bands in GeoTIFF Images)

Task: A code template has been given under folder Part-2 which loads GeoTIFF raster images into spatial dataframe. The dataframe has the following columns:

- origin – the path of an image in the disk
- Geom – the geographical location of the image
- height – the height of the image
- width – the width of the image
- bands - number of bands in the image
- data – the array containing the elements of bands

Your task is to complete the method get_bands_max() under the file task.py. The method takes a dataframe as parameter. The structure of the dataframe has been described above. You need to return a numpy array of shape m x n from this method. Here, m is the number of images/rows in the dataframe and n is the number of bands. Element A_{ij} in the numpy array indicates the maximum element of band j in image i.

Set up Instructions for Part A: You should install pyspark >= 3.0.3.

Submission Instructions:

- You should submit a .zip file containing two sub folders: Part-1 and Part-2. Folder Part-2 should only have the file *task.py*. Folder Part-1 contains 1) jar file *Spatial-Operations-assembly-0.1.jar* and 2) Apache Sedona project Spatial-Operation. Once you compile the Apache Sedona project, two target folders are generated. One inside the root folder of the project and another one inside the project subfolder. Delete both target folders since you are submitting the generated jar separately.
- You need to make sure your Part-1 code can be compiled by entering *sbt clean assembly* command. We will run the compiled jar on our cluster directly using "spark-submit" with parameters. If your code cannot compile and run, you will not receive any points.

Assignment Tips!

- Please make sure that there is no syntax or other errors in your submission. In case of any syntax error, 0 marks will be given.
- For any case of doubt in the assignment, please make use of the TA office hours and discussion board. Individual emails would not be entertained. TA will check the discussion board periodically.