Bishop's University

CS 450 – Elements of Big Data / CS 550 – Big Data Management and Analytics

Assignment 3: Sparkifying Income

1. Introduction

The objective of this assignment is to gain experience programming in Spark, to understand differences in implementations using Spark versus MapReduce or standard streaming, and to gain experience with Spark. Python version 3.7 or later and PySpark 3.0 or later must be used. You must also keep the data in Spark RDDs. Additional approved libraries that are not in the template will be listed here (if any):

```
import random
import numpy as np #for numeric algebra and arrays
import math
import hashlib
import csv
```

Here, you will repeat objectives from assignment 1 but using Spark rather than Streaming/MapReduce.

Data. You will use the same data as assignment 1: Two versions of the data are provided, (1) a small trial version with only 1000 integers to use while developing your method, and (2) a test that goes over 1 million integers to test your data on a larger dataset:

trial_incomes.csv

test_incomes.csv.zip

A demonstration code of different tools of Spark is provided in a notebook that can be run on Google Colab. In fact, it contains the installation of Spark, and demos for the following

functions defined in Spark: *reduceByKey, groupByKey, combineByKey, sortByKey, join, leftOuterJoin, rightOuterJoin, intersection, cogroup, groupWith, distinct, repartition, coalesce*. You are required to rely on functions in spark to resolve the task of this assignment. The data should be read in with *income_rdd = spark.sparkContext.textfile("input.csv",32)* and the rest is up to you as per fulfilling the instructions below. You should write your pyspark code in a notebook where the response for each task can be run successfully on Google Colab.

2. Tasks

The aim is to calculate distinct incomes, median of incomes, most frequent income, and count per 10power. For this subtask, you must calculate the true value of all four data summaries you produced in assignment 1:

1. **count of distinct incomes** – The number of distinct incomes in the dataset
2. **median** – The median of all incomes in the dataset: the income at which there is an equal number of values greater than the income as there are values less than the income.
3. **mode** – The mode of all incomes in the dataset: the most frequently seen income.
4. **count per 10power** – counting the incomes by powers of 10. That is, for each integer round it down to its nearest power of 10 (for example 3 map to $1 = 10^0$; 30 would map to $10 = 10^1$. 87 would map to $10 = 10^1$; 870 would map to $100 = 10^2$, 100 would map to $100 = 10^2$ etc….). Your goal is to count the number of integers between each power of 10.

Do not use an approximate algorithm but count every data point in the calculations. **However, you are restricted to only doing so with at most four shufflable transformation – a transformation which may cause a shuffle:** `reduceByKey, groupByKey, combineByKey, sortByKey join, leftOuterJoin, rightOuterJoin, intersection, cogroup, groupWith distinct, repartition, coalesce.`

Note that the solution may use less than 4 such transformations and you will also be graded on using spark transformation efficiently. For example, using a `groupByKey` when `reduceByKey` would suffice will cause loss of points. Your code will be tested on a dataset of the same size as test.csv but with a different distribution of positive integers as incomes. Do not assume any particular distribution.

3. Submission

You must submit the *pdf* file of your report. In fact, it must contain the responses in form of outputs for the four tasks. In addition, you must submit all your implementations in a `.ipynb` file that can be run on Google Colab. ***Please, do not submit your assignment in .zip or .rar files.***

4. Useful external references

- NumPy documentation
- SciPy documentation
- Spark Apache
- Python programming language