Lab 6: Getting Hadoop To Work

Due date: February 17, 11:59pm.

Note: Lab 7 will be assigned on February 17.

Lab Assignment

Assignment Preparation

This is an individual lab.

Your goal is to follow the set of instructions to complete a simple Hadoop program provided to you by the instructor. The general instructions are:

1. You are provided with a Java file `StudentFilter.java`. This file contains pretty much a complete setup for a simple MapReduce job on Hadoop.
2. Portions of the Java code have been witheld from the file provided to you. These portions primarily come from the `map()` and `reduce()` methods. You are given instructions (below) on how to complete these two methods (Each person in the class will have to write a somewhat different program from others).
3. You are asked to compile and run the program that you finalized using `hadoop`. Your deliverables for this lab are the output generated by the run (stored in the output file) as well as the confirmation of a successful `hadoop` run in the form of the diagnostic messages reported by `hadoop` while running your program.

Assignment

This section discusses your assignment in detail.

Input File. The program you write will operate on a file `students.csv` located in the `/data` directory on HDFS. This file should be readable to all of you. This file contains a list of students and their majors. A few sample rows from this file are:

```
1,Deidra Manson, Software Engineering
2,Imelda Janise, English
3,Audie Porraz, Biology
4,Marlo Olbrish, Computer Engineering
5,Filiberto Catino, Computer Engineering
9,Johnathon Branin, Biology
10,Armando Gallina, Biology
11,Athena Vandeyacht, Statistics
```

First column is the row number (it will serve as the key of the key-value pairs provided as input to your MapReduce program), second column is the name of the student, third column is the major of the student. The second and the third columns combined form the value part of the key-value pair provided to your program.

Task. Your goal is to write a MapReduce program that *finds all records where the first name of the student starts with the same character as **your** first name*[1]. Your program shall ignore all other records. For each selected record, your program shall report (as a key) the full name of the student, and as a value your name (first name and last name).

It is possible that no student in the `students.csv` file has first name that starts with the same letter as yours. In this case (after you have established it for sure), change your program to report the names of all Biology majors as keys, followed by (as values) by your full name.

Example. My version of the program, after going through the input shown above, produces the following output:

```
Audie Porraz    Alex Dekhtyar
Armando Gallina Alex Dekhtyar
Athena Vandeyacht       Alex Dekhtyar
```

A person whose first name is "Caitlin Donovan" would have to write a program that produces the following output:

```
Audie Porraz Caitlin Donovan
Johnathon Branin Caitlin Donovan
Armando Gallina Caitlin Donovan
```

[1]For example, because my first name is "Alexander", my version of the program has to look for first names that start with letter "A": "Audrie", "Angela", "Aaron", etc.

StudentFilter.java. I am sharing with all of you the file `StudentFilter.java`, which contains most of the code for your program. It is already organized as a proper MapReduce job for Hadoop. The code in the `map()` and `reduce()` methods is stubbed - *you will have to complete it as specified above.* All other code is present.

Additional tasks. The `StudentFilter.java` program provided to you is set to read the input file from a specific `HDFS` location (`/data/students.csv`) and is set to place output into the `/user/<you>/test/filter` directory (please, ensure, before running the program that you create the `test` directory inside your home directory on `HDFS`).

After you successfully compile and run your program, please modify the code to take the locations of the input file and the output directory from the command line. Submit the results of running your program after it has been appropriately modified.

Deliverables

Submit three files:

- The `part-r-00000` output file produced as the result of your program's run.
- A file named `studentFilterRun.txt` which contains the diagnostic output Hadoop prints to the terminal when it is run. The file must start with the `hadoop` command that runs your program, and contain all the diagnostic output until the next linux prompt.

  You can prepare this file by running your hadoop job and the pasting the necessary text into a file. Please note that every hadoop run is recorded by the hadoop process monitor. I will use your submitted file to confirm that you indeed successfully completed the job.

  To make it clear what is needed I will share the file from one of my successfull runs with you.
- The source code of your program.

Submission

Submit all your files using handin:

```
$ handin dekhtyar lab06 <FILES>
```

Good Luck!