Homework 1

Hadoop 30088 - Fall 2016

Part 1. Running MR2 and Spark jobs (8 points)

It is important that you finish Practice 2. After you have finished the practice, you should find the following very easy.

Running in Eclipse

In Eclipse, use the practice 2 workspace and:

- run MR2WordCount using data/shakespeare/histories/* as input.
- run SparkWordCount using data/shakespeare/tragedies/* as input.

Upload to Canvas the output file part-r-00004 generated by MR2WordCount, reducer 4.
Upload to Canvas the output file part-000004 generated by SparkWordCount, partition 4.

Running on a Cluster

Now generate the jar files using **maven install** in Eclipse. Use these jar files to

- run MR2WordCount on the VM Hadoop cluster. Use the “hadoop jar” command.
- run SparkWordCount using the Spark Master. Use the “spark-submit” command.

Find the log files using your browser:

- use Hue
    - Navigate to / var/ log/ hadoop-yarn/ apps/ cloudera/ logs
    - Find the log for the last job - this will be your last MR2 job.
    - Open the log, view it as a text file, and take a screenshot[1].
    - Save the screenshot as HW1ss1
    - Upload HW1ss1 to Canvas
- use the Spark Master (http://quickstart.cloudera:8080/)
    - Drill down on the worker
    - Find your last job
    - Open the stderr log and take a screenshot.
    - Save the screenshot as HW2ss2
    - Upload HW1ss2 to Canvas

[1] Screenshots have to be taken from the Host computer. You can’t get a screenshot within the VM.

Part 2. Writing MR2 and Spark jobs (17 points)

This part is a follow-up to Practice 3 and Lecture 3 (on October 27). In this section, you will be writing your own MR2 and Spark jobs. Feel free to create a copy of one of the practice projects and use it to create your homework project.

The Stock Screener

You may or may not have used Google's stock screener. In the next few weeks, we are going to be writing our own using the stock and company data that you have on your VM.

Let's begin by using the NASDAQCompanyInfo file you have in directory:

`~/Desktop/datasets/companies`

This is a comma-delimited file. If you open the file, you will see the header:

"Symbol","Name","LastSale","MarketCap","IPOyear","Sector","industry","Summary Quote",

Write an MR2 job to do the following

- In main(), set the input to the `NASDAQCompanyInfo` file
- In your Mapper:
    - Parse out the Symbol, MarketCap and Sector for each line.
    - Convert MarketCap to a usable number - for example: convert $1B to a 1000000000.00 (double).
    - Filter out Symbols that have a MarketCap greater than $1B.
- In your Reducer:
    - Write out the Symbols in each Sector.
    - Write out the number of symbols for each Sector.
    - Write out the total capitalization for each Sector.

Upload to Canvas your java code. Title the code "MR2Screener1.java".

Challenge: write a Spark job to do the same (Extra credit: 10 points)

Write a Spark job that uses `NASDAQCompanyInfo.` The job should write out the Symbols in each sector, the number of Symbols in the sector and the total capitalization for the sector.

Upload to Canvas your java code. Title the code "ChallengeSparkScreener.java".

I award partial credit, so if you attempt the challenge, but cannot get all the pieces, feel free to submit what you have done.

What to turn in, and when

In Canvas, you will find that you have an area where you can upload files for the Homework 1 assignment. For full credit, you need to upload:

Outputs from Part 1:
part-r-00004 (2 points)
part-00004 (2 points)

Screenshots from Part 1:
HW1ss1 (2 points)
HW1ss2 (2 points)

Code from Part 2:
MR2Screener1.java (17 points)

Bonus challenge (not required, this is extra credit):
ChallengeSparkScreener.java (10 points)

FULL CREDIT FOR THE HOMEWORK is 25 Points. An additional 10 points is awarded as extra credit. You *never* have to do extra credit.

Homework 1 is due on November 6 (Sunday) at midnight.

You have two full weeks to finish this homework. After that, heavy penalties apply.

PLEASE DO NOT COPY ANOTHER PERSON'S CODE. Students who turn in identical code will be asked to explain, in detail, the way that their code works.