School of Computer Science

COMP47470

Project 2
Spark and Hadoop

Teaching Assistant:	Cormac English
Coordinator:	Dr Anthony Ventresque
Date:	Monday 8th March, 2021
Total Number of Pages:	5

General Instructions

- This project is largely based on the material covered in the lab sessions. However, you may desire additional functionality (such as Java string manipulation functions for Hadoop). We would encourage external research in this regard - the documentation for Scala/Java etc. will be helpful for you. As an example, Scala functionality may be explored **here** (in this instance - the flatmap function!)
- We ask you to hand in an archive (zip or tar.gz) of your solution: code/scripts, README.txt file describing how to run your programs, a short pdf report of your work (no need to include your code in it).
- The report should not be longer than 10 pages (this is not a hard constraint though).
- The report must be submitted as a PDF, but stylistically you may use whichever software you like to prepare it (e.g. Word, Google Docs, Latek) as long as any code included can be copy-pasted!
- For Sections 1 and 2, each answer in the report should include the following:
  - The answer to the question (usually a number)
  - The code you used - either include a snippet (if the code is short) or reference an attached script/txt file. You may use scala scripting for this assignment, but a text file including **all lines of code you used (NB - running them sequentially must replicate your work)** will be accepted.
  - A brief explanation of what your code is doing/how it operates. This can be very short (one sentence) as long as it demonstrates an understanding of each portion of your code. For extensive scala commands (e.g. with multiple functions strung together) please make sure you explain what each portion is doing.
- For Section 3 - please complete **both** of the two available options
- The breakdown of marks for the project will be as follows:
  - Exercise 1 (Hadoop): 30%
  - Exercise 2 (Spark): 30%
  - Exercise 3 (Reflection): 40%
- **Due date: 11/04/2020**

1 Hadoop

In this section you will run Hadoop MapReduce jobs on a dataset of shuttle trips. Download the data using the following code (remember to make it all one line!):

```
wget --no-check-certificate
↪ 'https://docs.google.com/uc?export=download&id=1Jvw6rGC8X8H7bDXzSiP4mf2Q_wyOFWs2'
↪ -O shuttle.csv
```

1. How many rides occurred per weekday?
2. Using a mapreduce operation, count the number of rows in the dataset
3. What was the most common number of passengers?
4. How many rides (grouped per day) had a total amount greater than 20?
5. How many trips occurred on weekdays?
6. How many trips occurred before 2pm? (you may include/exclude 2pm - just be consistent!)
7. What is the average "fare amount" per day . You may use a combination of mapreduce output and bash scripting to accomplish this (e.g. two separate mapreduce tasks with a small bash script "combining" the two outputs). For the purposes of this assignment, you can assume that the fare will be an integer between 0 and 9.

1.1 Notes:

- This section can be completed by modifying the "WordCount" example provided to you in Lab 4. Please look at the solutions document for that lab to orient yourself. You will be editing the WordCount file and compiling it inside your docker container - this will limit your debugging potential so it may be preferable to test some changes in a separate Java environment before inputting them (see the last point below).
- The majority of modifications will be made in the TokenizerMapper class (hint: look at the context.write() and word.set() functions) and the IntSumReducer class (take a look at the code after "public void reduce").
- Please remember to look at the Hadoop documentation - many of the functions will not accept strings/integers, and will instead require the creation of new variables of the appropriate type - e.g. in the context.write() function in the TokenizerMapper class, the number "one" is not an integer or a string - it is an "IntWritable" - if we wanted to modify this write to include a different integer (using 7 as an example), we would have to make a new IntWritable as so: "IntWritable t1 = new IntWritable(7);" and include this in the write: "context.write(word, t1);. Similarly the "key" variable in IntSumReducer is not a string, but a "Text" item that can be constructed similarly. This information is available in the code, but might require a close reading!
- The following resource may be useful to you: MapReduce Tutorial

- You may find it helpful to create a smaller dataset to probe functionality on! Alternatively, you might require a Java IDE to test some specific functionality - if you do not have Java on your system, there are some basic implementations available online - see here

2 Spark

For this section, you will be assessing a dataset of Netflix shows. Download the data using the following code (remember to make it all one line!):

```
wget --no-check-certificate
↪ 'https://docs.google.com/uc?export=download&id=1qzTZflrsZ89dYPct3KpFO0O9LmrN2yck'
↪ -O netflix.csv
```

Launch the Spark shell and then create an RDD or a DataFrame from the input file. For each of the following tasks, write Scala code to solve it. You can use operations on DataFrames (see lab 5 and here, in the Scala API, DataFrame is simply a type alias of Dataset[Row]), spark SQL on Dataframes (see lab 5), or operations on RDDs (see lab 5, here, and here).

1. How many rows are in the dataset?
2. Are there more TV Shows or Movies in the dataset?
3. Compute the total duration for each type (Movie/TV Show) in the dataset. You may disregard the season/minute distinction for this question.
4. What is the oldest (by release year) movie in the dataset?
5. Determine the number of movies per country. Some country fields will contain multiple countries delimited by a semicolon.
6. Determine the most frequently occurring term in the the "description" field (i.e what word appears the most often in that column!)
7. To the nearest decade (e.g. 2000-2009 == 2000s), what are the top 5 decades in terms of number of films released.
8. In what month have the most films been **added** to Netflix (i.e. sum of all the films **added** in a given month)

2.1 Notes:

- Remember to check your "pipeline" if you are receiving unexpected errors - Scala is very precise, so make sure you know what is being passed from each function to the next - it may not be what you think! A good example is during the mapping process - repeated maps (e.g. when splitting on successive dividers) can often lead to nested arrays (rather than an expected single level array!). As a hint, think about what flatMap(_.toList). might do in a situation like this!
- Don't be afraid to do outside research for this section - and especially make sure to look at the documentation linked above for functions etc. - but you shouldn't require anything too distant from what you have already completed in the labs!

- Simplicity is key - try to avoid creating complex structures (e.g. an array of arrays of STRING,INT ...) where possible. It will be less frustrating for you, and will speed up the process of creating a solution.

3 Reflection

Please complete **both** of the following two titles:

3.1 Topic 1 - Spark and Hadoop

Write a short report (max. 2 pages) exploring the following topic: "Spark & Hadoop: Advantages and drawbacks of each technology, and why you might choose one over the other". Your discussion should address the following topics:

- A brief overview of the two technologies
- What the major technical differences are between Spark and Hadoop
- In what circumstances would you choose Spark over Hadoop
- In what circumstances would you choose Hadoop over Spark

You are expected to cite at least one paper in your answer, in addition to any web/alternative resources that you use. You may use a citation style of your choice, as long as you are consistent! Two papers (concerning Spark/RDDs and Hadoop/HDFS) are available on Brightspace to help you orient yourself - you may use these to fulfil the paper requirement (but you are not limited to using only them!).

3.2 Topic 2 - Paper Review

Write a short report (max. 2 pages) on one of the research papers that are available on Brightspace. The following list of sections is an indication of how to write your paper. Some of the items might not be relevant for all papers, and you might want to add some sections in your report (e.g. evaluate the posterity of a solution etc.). We will have an open mind when reading your report and we just want to see how you analyse a research paper and are able to discuss it - in short there is no one single perfect report, everything that shows you made an effort to understand and focus on the important parts (research methodology, hypotheses, etc.) will be welcome.

- identify the question/challenge the paper addresses. Explain in your own words what the motivation for the research is.
- describe briefly the related work, i.e., the other (related) solutions that the authors compare themselves to. Show the limitations of these related solutions
- Give an outline of the solution proposed by the authors (no need to go into details) showing the main components
- describe their scientific method: what are the research questions they evaluate, how do they evaluate

- describe briefly their results
- give your impression on the idea, what you liked about the paper and whether you see any limitations etc.