# CS 4740 Programming Assignment 3

Note: you can pause any time during this assignment.

Goals:

Gain hands-on experience with the MapReduce framework. MapReduce, initially developed by Google (see paper below), is a programming paradigm used primarily in Hadoop systems. MapReduce provides a more efficient way to process a large amount of data. This assignment will have three parts. The first is to introduce MapReduce and the basics of 'mr_job' with the Iris dataset. We will then explore a larger data file (a book). Lastly, we will look at more complex MapReduce programs as well as an introduction in Amazon's Elastic MapReduce .

This assignment will use Python and the built-in 'mrjob' library.

Link to original Google paper:
https://www.usenix.org/legacy/events/osdi04/tech/full_papers/dean/dean.pdf

Step 1: (In each of the following steps, you need to refer to the instructions in this step)

Follow the instructions in the official 'mrjob' tutorial: https://mrjob.readthedocs.io/en/latest/

While there is a lot of useful information in this tutorial, focus specifically on the installation, "Why mrjob", "Fundamentals", "Concepts", and "Writing Jobs" sections.

If you have never used Python before, you may want to refer to https://www.youtube.com/watch?v=YYXdXT2l-Gg (Windows and Mac) or https://www.youtube.com/watch?v=UXjjEZroOu0 (Linux). You can find a lot of beginning tutorials online.

Step 2:

Download the Iris dataset from the following link: https://archive.ics.uci.edu/ml/datasets/iris

Read the description file to familiarize yourself with the dataset.

Using Python and mrjob, write a program to find each of the following metrics:

1) the minimum sepal length
2) the maximum petal width
3) the average sepal width for the class "Iris Setosa"
4) the difference in average sepal and petal length for all non-"Iris Setosa"

Please refer to the end of this document about what needs to submit.

Step 3:

Now that we have a basic familiarity with MapReduce, we will move to a different example.

Please download the attached “harry.txt” from your Collab assignment PA3 link.

Calculate the word frequency for the text file. The word frequency is defined by the number of times a word appears in the book. Specifically, please find the number of occurrences for the following words: ‘magical’, ‘soaring’, and ‘lopsided’ and include the results in your final submission.

You may have noticed that the output list of word frequencies includes the same word with additional punctuation (e.g., "late", "late,", "late.\"). Please refer to the section “Writing your second job” in the ‘mrjob’ tutorial above or search “python remove punctuation” in Google to solve this problem and only show the word without punctuation in the output.

Step 4:

Using the structure from step 3, you can find out the frequencies of the words in *Harry Potter and the Prisoner of Azkaban*. Please list the top ten most frequently used words and their associated word count in the final results.

Hint: You may have noticed that to find the top ten most frequently used words, the previous issue in step 3 with punctuation must be fixed. You may also want to use more than one mapper/reducer (the additional mapper/reducer is for sorting) or more than one program (the second program prints out the final result). Please refer to https://mrjob.readthedocs.io/en/latest/guides/quickstart.html#writing-your-second-job for the multi mapper/reducer.

Step 5:

The last step of this homework is to get basic experience using Amazon Elastic MapReduce (EMR). To do this part of the assignment, please follow the following steps:

**1. Configuring AWS credentials**

Configuring your AWS credentials allows mrjob to run your jobs on Elastic MapReduce and use S3.

- Create an Amazon Web Services account
- Go to Security Credentials in the login menu (Click on your user name, go to your security credentials), say yes, you want to proceed, click on Access Keys, and then Create New Access Key. Make sure to copy the secret access key, as there is no way to recover it after creation.

Now you set aws_access_key_id and aws_secret_access_key in your mrjob.conf file like this:

```
runners:
  emr:
    aws_access_key_id: <your key ID>
    aws_secret_access_key: <your secret>
```

**2. Run your job with -r emr**

You can store this file at /etc/mrjob.conf, ~/.mrjob.conf, or ./mrjob.conf. Then run your job with -r emr:
python your_mr_job_sub_class.py -r emr < input >
E.g., python your_mr_job_sub_class.py -r emr harry.txt

If you store this file at a different directory, to run your job with -r emr, you need to pass it via --conf-path using the command below.
python your_mr_job_sub_class.py -r emr --conf-path <YourDirectory> < input >
E.g, python step4.py -r emr --conf-path /af1/hs6ms/mrjob.conf harry.txt

(Optional) For more details, you can refer to https://mrjob.readthedocs.io/en/latest/guides/runners.html#running-on-emr and https://mrjob.readthedocs.io/en/latest/guides/runners.html#configuration.

So far, you have run your script from step 4 in EMR. Please verify that the output matches your results from step 4 and include screenshots of your console log (this is the output in command prompt or terminal).

A few things to note:

- When running the script, you need to replace <input> with the input data file such as harry.txt.
- Feel free to use an existing security key that you already have for your 'mrjob.conf' file.
- You may get the following message in terminal: "Waiting 10 minutes for logs to transfer to S3... (ctrl-c to skip)". If you get this, please 'ctrl-c'
- This portion will take time as Amazon has to create and then terminate a cluster. If you find yourself waiting for ~15 minutes, do not worry!

Submission:

Please include ONE PDF in the following format:

- The first page should contain a list of answers to each of the scripts written above in order (in steps 2-5) and screenshots for console log in step 5:
  - Example:
    - 1) 4.57
    - 2) 3.59, etc.
- The second page (and onwards) should contain the corresponding list of python scripts you used to calculate the answers
  - Example:
    - 1)

```
def print_first():
    print(4.57)
```

Important notes:

- All these programs (especially in the steps 2 and 3) can be done iteratively without using MapReduce. However, to receive credits on these parts, you must perform your calculations using MapReduce.
- Please start this assignment several days in advance of the deadline. This is largely due to the number of programs being written (though most follow a similar general structure)
- When running these programs, include the name of the data file (e.g. ‘harry_potter.txt’) in the command line argument