CPSC 5330 -- Spring 2024
Week 1 Lab: HDFS and Hadoop

Part 1 (5 points)

Write a shell script with the name

`word-count-shakespeare`

that will print the ten most frequent terms in a subset of the corpus: the Shakespeare plays. The Shakespeare plays are all files in the corpus whose filename begins with '`shakespeare-`'.
Your script should write 10 records to STDOUT of the form <term> <space> <count>.
Your solution must use the WordCount MapReduce application from this week's in-class demo.

Assume the script will be run on an instance of the class EC2 image in a directory containing three files from this week's in-class lab

- The program `WordCount.java`
- The script `compile-map-reduce`
- The script `run-map-reduce`

Your script must "clean after itself" – by deleting all files it created.

Part 2 (5 points)

You will notice that the "words" produced by MapReduce wordcount are pretty messy, for example:

Abyss 4
Abyss, 5
abyss 7
abyss, 4
abyss. 2
abyss: 1

There are problems of punctuation, and case sensitivity. Modify the word count program to a new Map Reduce module named `WordCountClean.java` so that it

(a) Removes all punctuation from each word
(b) Converts the word to lower case

There is a file `StringHints.java` in the repo showing how to do those two operations for Java strings. Hint: it is possible that an input word is all punctuation, in which case the "cleaned" word would have no characters. Your mapper should not pass on words with no characters.

Write a new script `word-count-shakespeare-clean`. This script will differ from the script you wrote in Part 1 in two ways:

- You will use your new `WordCountClean` module to do the map reduce word count.

- The script will emit N records rather than the hard-coded 10 records. The parameter N will be an argument to your script. (Search for "bash command line arguments" – it's very simple!) You may assume that the caller will supply a value for N.

Part 3 (5 points)

The number of unique words in a document or set of documents is often referred to as its *vocabulary*. The ratio of unique words to the total number of words in a set of document its "richness of vocabulary."

So, any way you want: answer the question of who has the richer vocabulary – Jane Austen, William Shakespeare, or whoever wrote the King James Bible?

Use the "cleaned up" definition of a word from Part 2.

You must write MapReduce script(s) to get the total number of words and the total number of unique words in a set of documents, but you can use a Python script to divide unique words by total words to get the ratio.

For this part you will not hand in a script. Instead, put together a document that is a mixture of code, output, and text, describing your approach and presenting your conclusion.

Part 4, EXTRA CREDIT, (2 points)

This part requires more changes to the Java code than the other parts, so feel free to skip it if your Java skills would require you to spend a lot of time on it.

What are the longest words in the corpus? Build new MapReduce code that finds the longest words. The output of the Reduce phase should be a triple

```
<length> <sample-word> <count>
```

where length is the word length, sample-word is *any* word from the corpus with that length, and count is the number of words with that length. Use the "clean" version of extracting words developed in Part 2.

You will write a script `longest-words` that writes to `STDOUT` the triples for the ten longest words in the full corpus, in descending order of length.

Part 0, Style Points (2 points)

(These are easy points to get, just by following instructions and handing in good clean work!)

Was the output clean and well presented? Was the text and explanations well written?
Were all the files there and named according to instructions?

To Hand In

A Zip file containing (only) these files

- For Part 1, your script `word-count-shakespeare`
- For Part 2, your program `WordCountClean.java` and your script `word-count-shakespeare-clean`
- For Part 3, your work in a file `vocabulary.pdf`
- For Part 4, your script `longest-words`
- A retrospective report in a file `retrospective.pdf` – a reflection on the assignment, with the following components
    - Your name
    - How much time you spent on the assignment
    - If parts of the assignments are not fully working, which parts and what the problem(s) are
    - Were there aspects of the assignment that were particularly challenging? Particularly confusing?
    - What were the main learning take-aways from this lab – that is, did it introduce particular concepts or techniques that might help you as an analyst or engineer in the future?