Project 4: Working with Hadoop

DS 730

Overview

In this project, you will be working with input, output, Python and the MapReduce framework. You will be writing multiple mappers and reducers to solve a few different problems.

If you recall, the map function is *stateless* and this is especially important when dealing with Hadoop and distributed work. We can't guarantee that any 1 mapper will read in all of the data. Rather, 1 mapper will likely read in a small portion of the data. The output that your mapper produces must only depend on the current input value. There is no need to sort anything in your mapping phase as Hadoop will do it for you.

Project Tasks

You must write a mapper file and a reducer file to solve the two problems posed in this project. You should test both of your solutions on your local Hadoop setup before running a larger input instance on Amazon Web Services.

Problem 1: Words with same number of vowels

For this problem, Hadoop will define what your input files are so there is no need to read anything in. The problem will be similar to the problem we worked on in lecture with a small twist. Instead of printing out how many times a word appears in the file, you want to print out how many words have the same number of vowels. For this problem, only the number of vowels matter. The actual vowel is not important. The output will be the number of vowels (let's call it **x**), followed by a colon, followed by the number of words that had **x** vowels. The output is sorted by the number of vowels.

Example:

Let's say the file contained the following words:

```
hello how are you doing today hello
```

The output would be:

```
1 : 1
2 : 4
3 : 2
```

Words are separated by spaces, tabs and new lines only. As an example, "half-time" would be considered 1 word.

Using the example above, since 'how' contains 1 vowel, the first output is **1 : 1** (i.e. exactly 1 vowel occurred in exactly 1 word).

'hello', 'are', 'doing' and 'hello' contained 2 vowels each. This is why we ended up with **2 : 4**. A word appearing more than once will be counted more than once.

Finally, 'today' and 'you' contained 3 vowels. For this problem, we will assume 'y' is a vowel. The final output line was **3 : 2**.

Problem 2: Pet census analysis

Assume you work for a pet store and you want to know where to spend your marketing money. A "pet census" was sent to all cities in your area. Each city compiled the data and sent you the results. Each city compiled the data in different ways but they all followed a basic rule. For each citizen, a string appears in the file with the following information:

If the citizen owns...	Then the string will include...
3 cats	3 **C**'s
2 dogs	2 **D**'s
4 fish	4 **F**'s
1 monkey	1 **M**

Example:

A citizen with 3 cats, 2 dogs, 4 fish and 1 monkey might appear as **CCCDDFFM**. However, another city may have compiled their information differently and that same citizen living in another city might have the string of **CCDDCFMF**.

Your goal is to print out how many citizens have the exact same number of pets.

Example:

Assume this is the input file:

```
CCDFM CDCDM FFDM FMDCC CDMFC MDFF
```

Your output would be:

```
CCDFM : 3

CCDDM : 1

DFFM : 2
```

Since you are defining where your input and output are at runtime, you can have them be wherever you want.

When you are working with Hadoop and other software at your job, you will likely be given some large dataset and have to work with it. You will not be given a set of sample input and output files to test your code. Therefore, part of this project is coming up with your own sample input and output files. You can expand some of the examples given in the questions and manually check if your answer is correct. If it is correct for several small cases, then you should feel good about your answer being correct for a large case that you can't manually check.

Submitting Your Work

When you are finished, submit *a .zip file* to the **Project 4 dropbox** that includes:

- Your mappers and reducers for both problems.