CS351 - IT Workshop II (Cloud Computing)

Assignment 8

Due on 31st October, 2024

(Please take screenshots, reduce the size of the images and submit in Google Classroom)

1. In this assignment we shall make use of the Hadoop streaming api to write our map reduce code. Hadoop Streaming uses Unix standard streams as the interface between Hadoop and your program, so you can use any language that can read standard input and write to standard output to write your MapReduce program. However, we shall use Python for this assignment. All we need to do is write a script for the mapper and reducer, and hadoop will take care of the rest. First, we shall simulate the behavior of a map reduce program using simple unix commands to understand how the mapper and reducer works in Section 2.

2. **Map Reduce Simulation:**
   For this simulation, we shall do a simple word count. The problem statement is: Given a text file (https://norvig.com/big.txt), get the count of every word in it. Now, the first order of business is to write the mapper.

   Word Count Mapper
   The job of the mapper is simple, read lines from stdin and spit out the key value pairs to stdout. Write a python script for the same. An example of expected output from the input is given in Table 1. Make sure to include the python shebang in your scripts and chmod +x yourscript.py . Test your script by running cat inputfile | ./mapper.py

Input File	Mapper Output
hello world testing testing hello	hello,1 world,1 testing,1 testing,1 hello,1

Table 1: Mapper Example

Word Count Reducer

The next task is to write the script for the reducer. The reducer job is to take the output of the mapper and spit out the final count of every word to stdout. We will simulate the sorting phase of the hadoop framework with sort command, so that all the same words are ordered together. Test your script by cat inputfile | ./mapper.py | sort -t ','-k1 | ./reducer.py

Mapper Output		Reducer Output
hello,1 world,1 testing,1 testing,1 hello,1	hello,1 **sort** hello,1 testing,1 testing,1 world,1	hello,2 world,1 testing,2

Table 2: Reducer Example

Run it in Hadoop

Now that we have written our mapper and reducer, we are ready to execute our program in Hadoop.

hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-*.jar -input path/to/inputfile -output path/to/outputdir -mapper path/to/mapper.py -reducer reducer.py

3. **Hadoop Assignment:**

- Implement a map reduce program to find all distinct words in the file. Perform data cleaning in the mapper such that all punctuations are removed and all words become lowercase.

inputfile	Map Reduce output
Hello World! Apache hadoop. apache spark.	apache hadoop hello spark world

Table 3: Distinct Words MR example

- Extend the word count example to include a combiner. Simply use -combiner combiner.py option.
- You are given a dataset of N points and you are given the C candidate points. Implement a map reduce program to assign each of the N points to the nearest (Euclidean distance) candidate point and update the candidate points by taking the average of all the points that were assigned to it. You may hard code the candidate points in your mapper if you want. Make use of the iris dataset (http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data), which is in the format
  sepallength, sepalwidth, petallength, petalwidth, class

  Use the following three points given in Table 4 as candidate points. For this exercise you may use consider the sepal length and sepal width, and remove the rest.

  5.8,4.0,1.2,0.2,Iris-setosa
  6.1,2.8,4.0,1.3,Iris-versicolor
  6.3,2.7,4.9,1.8,Iris-virginica

  Table 4: Candidate Points