Hadoop Assignment
University of Amsterdam BSc Informatica
Concurrency & Parallel Programming

Contents

1 Overview

The goal of this assignment is to enable you to gain experience programming with:

- The Hadoop open source framework
- Breaking down a task into a parallel distributed MapReduce model

2 Assignment

In this assignment you are called to analyze a twitter dataset:

1. Find the top ten tags used in this dataset
2. Perform sentiment analysis on the twitts written in English & find the average sentiment and the standard deviation for each of the top ten tags

In natural language processing, language identification is the problem of determining which natural language is used in a text, while sentiment analysis aims to determine the attitude of a person according the her/his writings.

3 Implementation

After you have studied and understood the tutorial use the code framework to implement the twitter dataset analysis. To clone the code from github use:

```
$ git clone https://github.com/skoulouzis/MapReduceAssignment14
    -15.git
```

Or download from: `https://github.com/skoulouzis/MapReduceAssignment14-15/archive/master.zip`. The dataset you will be analyzing can be found on blackboard.

3.1 Tag Counter

To find the top ten tags in the dataset you will need to modify the `Map` class in order to detect hash tags. To complete this part of the assignment you'll need to go through the following steps:

1. Use the mappers to detect if a line contains any hash tag
2. If the line contains one or more hash tags emit them to the reducers using the appropriate keys and values
3. Use the reducers to count the number of hash tags
4. Look at final output and determine which hash tags are the most popular

Attention: You don't have to implement any sorting algorithm to get the top ten tags. Simply inspect the output file and determine the tags with the most occurrences.

3.2 Sentiment Analysis

For the second part of the assignment you will need to modify both the `Map` and `Reduce` classes. To perform language detection you will use the JLangDetect API (`https://github.com/melix/jlangdetect`). Import in you class the UberLanguageDetector package:

```
import me.champeau.ld.UberLanguageDetector;
```

To detect the language of a String variable named `text` you can use:

```
String lang = UberLanguageDetector.getInstance().detectLang(
    text);
```

If the language in the variable `text` is in English the `lang` variable will be set to `'en'`.

To perform sentiment analysis on English text you will use the Stanford Natural Language Processing API (`http://nlp.stanford.edu/`). To use this API you'll need to import the necessary packages in you class:

```
import edu.stanford.nlp.ling.CoreAnnotations;
import edu.stanford.nlp.pipeline.Annotation;
import edu.stanford.nlp.pipeline.StanfordCoreNLP;
import edu.stanford.nlp.rnn.RNNCoreAnnotations;
import edu.stanford.nlp.sentiment.SentimentCoreAnnotations;
import edu.stanford.nlp.trees.Tree;
import edu.stanford.nlp.util.CoreMap;
```

To detect the sentiment of a String variable named `text` you can use the following method:

```
private int findSentiment(String text) {
  Properties props = new Properties();
  props.setProperty("annotators", "tokenize, ssplit, parse,
      sentiment");
  props.put("parse.model", parseModelPath);
  props.put("sentiment.model", sentimentModelPath);
  StanfordCoreNLP pipeline = new StanfordCoreNLP(props);

  int mainSentiment = 0;
  if (text != null && text.length() > 0) {
int longest = 0;
Annotation annotation = pipeline.process(text);
for (CoreMap sentence : annotation
.get(CoreAnnotations.SentencesAnnotation.class)) {
  Tree tree = sentence
  .get(SentimentCoreAnnotations.AnnotatedTree.class);
  int sentiment = RNNCoreAnnotations.getPredictedClass(tree);
  String partText = sentence.toString();
  if (partText.length() > longest) {
    mainSentiment = sentiment;
    longest = partText.length();
  }

}
  }

  //This method is very demanding so try so save some memory
  pipeline = null;
  System.gc();
  return mainSentiment;
}
```

This method returns an integer that reflects the sentiment of a text. The higher the number the more positive the sentiment of the text. Additionally, in this method it is important to specify the `parseModelPath` and `sentimentModelPath` variables. These two variables are used for setting the path where the `englishPCFG.ser.gz` and `sentiment.ser.gz` files are located (available in blackboard). The `englishPCFG.ser.gz` file is used to perform the parsing of the text and the `sentiment.ser.gz` to detect the sentiment. To use these two files you will first have to make them available for all the mappers. This is done with the use of the `DistributedCache`. `DistributedCache` is a facility provided by the MapReduce framework to cache files (text, archives, jars etc.) needed by applications. To use it you will have add the files you want to the `DistributedCache` in the configuration step (in in the `configureJob()` method) :

```
    DistributedCache.addCacheFile(new Path(filePath).toUri(), conf)
        ;
```

To retrieve the path of the files form the `Map` class you have to override the `configure()` method. The `configure()` initializes a new mapper instance from a `JobConf`. This method is called before the `map` method:

```
    @Override
    public void configure(JobConf conf) {
      try {
Job job = Job.getInstance(conf);
//Get the cached files here
    } catch (IOException ex) {
Logger.getLogger(Map.class.getName()).log(Level.SEVERE, null, ex)
    ;
    }
  }
```

Therefore, the steps you'll need to take to complete this assignment are:

1. Add the `englishPCFG.ser.gz` and `sentiment.ser.gz` files in the distributed cache
2. Override the `configure()` method in the `map` class and get the local path for the `englishPCFG.ser.gz` and `sentiment.ser.gz` files
3. Use the mappers to detect if a line contains any hash tag
4. Detect the language of that line
5. If the language is English perform the sentiment analysis
6. Emit the proper keys and values to the reducers
7. Use the reducers to calculate the average and standard deviation

The standard deviation of a mean can be calculated using:

```
    sd = sd + Math.pow(val - mean, 2);
```

3.3 Reporting

Items you are expected to hand in:

1. Small report (maximum 2 pages)
2. Your source code including comprehensive comments. When you are done with your assignment, you don't need to include the libraries. Simply handing your `src` folder

Your report should contain the following sections:

1. Introduction: Very briefly (no more that 2 paragraphs) describe the major components of Hadoop
2. Implementation: Describe your approach for implementing the twitter dataset analysis
3. Results:
   (a) A table with the top ten hash tags you found, their occurrences, their average sentiment and their standard deviation
   (b) A graph showing speedup measurements when using 1, 2, 4, 8 mappers
4. Conclusion:
   (a) Discuss what is the benefit of adding more mappers
   (b) If you didn't observe any speed up by adding more mappers explain why

3.4 Experiments

Keep in mind that you will be shearing the cluster, therefore your measurements will be influenced by each-others experiments. With this in mind, when you measure speedup you should perform multiple experiments analyze them statistically and present your findings. You should graph the speedup and include error bars. Speedup is a metric that shows how much faster an algorithm runs if we add more instances to execute it. It gives the relative performance improvement when executing a task and is calculated by:

$$S_p = \frac{T_1}{T_p}$$

Where T_1 is the execution time when using one node, in this case one mapper and T_p is the execution time when using p nodes, in this case 1,2,4 or 8 mappers.

3.5 Notes

If you want to run your experiments at any time you want without having to be logged in you can use the `at` command:

```
$ at 06:45
at> ls
at> <EOT>
job 1459 at 2014-11-12 06:45
```

After typing `at 06:45` you type the command you wish to run and to exit you press Ctrl+D. The above command will execute the `ls` command in 06:45. More to get more details you see man page.

3.6 Deadline and possible points

Deadline: **November 28**
Points: Tag Counter: 4/10, Sentiment Analysis: 6/10.
To score full points your code has to work and produce correct results. Also in you report you will have to include all the requested graphs and results and also answer to all the questions of Section 3.3