Juan David Botero
Hadoop
15 de junio de 2019

Hadoop MapReduce

For this assignment first its crucial to understand the **problem** we are trying to solve, the **input** data available and the desired **output**:

Problem We want to calculate the 5 most relevant conversations (hashtags) in Twitter for each country.
Input: value pairs (country, tweet_text) where country is the country code of the twitter user and tweet_text contains the text and the hashtags (#) that we want to count.
Example: (US, "The components of #hadoop are #yarn #hdfs and #mapreduce")
Output: value pairs (country, hashtags) where country is the country code and hashtags is a list of strings that contains the 5 most tweeted hashtags.
Example: (US, #hadoop #spark #yarn #aws #azure)

After understanding what is our input data, how it's formatted and how does the output of the job should be, it is easier to reverse engineering and find the different processes that MapReduce needs to do.

My approach to this problem is as follows:

Job1
Map Stage

The number of map tasks depends on the number of blocks of input data for the 193 countries member of the United Nations. So, in this stage we have many blocks of data with the key as country and value as the tweet with the hashtags.

(US, "I love #madrid and #barcelona with #passion…but maybe more #BARCELONA")
🔑 , 🧮

This Map Stage will drop the strings that don't follow the rule of **being after a "#" character and ending with a space, comma, point, colon, semicolon " ,.:;"**, so all the hashtags are extracted from the tweet.

The new Key Value Split will output the country code and a 0 or a 1 as key, if the first letter of the tweet is inside a-m it will be a 0, if its inside n-z it would give it a 1, and the hashtags in lowercase as a value plus a counter. The lowercase is important so hashtags can be grouped by meaning and avoid having duplicated values because of upper and lower cases and possibly a wrong answer.

The first letter of the tweet is useful to better distribute the data load of the reducer in two groups because 2 groups * 193 countries = 386 possible keys, while if you use country as key countries with much more people, like India or China, will have a huge amount of data for the reducer to handle and if you use country and hashtag as key, the combination of words and countries are much higher.

(US0,#madrid,1)
🔑 , 🧮
(US0,#barcelona,1)
🔑 , 🧮
(US1,#passion,1)
🔑 , 🧮
(US0,#barcelona,1)
🔑 , 🧮

*A **combiner** will be applied to every map task in order to "pre-reduce" the number of records and have less data load for the next job.

(US0,#madrid,1 #barcelona,2)
🔑 , 🧮
(US1,#passion,1)
🔑 , 🧮

Sort and Shuffle will sort alphabetically the keys from all of the mappers and shuffle them to the same node so they can be processed. One possible issue here is that if a group has many values for countries with big populations, some problems may apply if only 1 node reduces, so if this is the case for that key (country_code 1or 0), more nodes should be assigned and then reduced again to get 1 final output per country and group.

(US0,#madrid,1 #barcelona,2)
🔑 , 🧮

(US1,#passion,1)
🔑 , 🧮

***Reduce Stage**

Here the reducer will output a list of all the hashtags with the counter for each country and each group, so we will have 386 files as outputs. Dropping the single values where the counter =1.

(US0,#madrid 98 #barcelona 47 #hadoop 46 #facebook 40 #amazon 30…)

🔑 , 🧮

(US1,#passion 97 #messi 110 #uefa 43…)

🔑 , 🧮

Job2

Map Stage

The map stage in Job2 will get the input from the Reducer of Job1 and merge the two groups with country as key and the value a list of the hashtags with their counter.

(US,#passion 97 #messi 110 #uefa 43 #madrid 98 #barcelona 47 #hadoop 46 #facebook…)

🔑 , 🧮

Reduce Stage

Here the reducer will get an input for each of the 193 countries and reduce the list to show the top 5 hashtags in descending order for each country.

(US, #messi 110 #madrid 98 #passion 97 #barcelona 47 #hadoop 46)

🔑 , 🧮

*Disclaimer: I'm not sure if the combiner and the reducer in **Job1** can add the counter for the hashtags if they are both inside the value: key,(#xyz counter)…what I found was that the only way to add the counter if it is alone on the value. If this is true, this solution would not be possible.