CPSC 4330 Big Data Analytics

In-Class Exercise 5 – Customize your own partitioner

In this exercise, you will write a MapReduce job with multiple Reducers, and create a Partitioner to determine which Reducer each piece of Mapper output is sent to.

Your task is to count the number of hits made from each different IP address in the sample web server log file that you uploaded to the /exercise/weblog directory in HDFS when you completed in-class exercise 1. The final output should consist of 12 files, one for each month of the year: January, February, and so on. Each file will contain a list of IP addresses, and the number of hits from that address in that month.

We will accomplish this by having 12 Reducers, each of which is responsible for processing the data for a particular month. Reducer 0 processes January hits, Reducer 1 processes February hits, and so on.

Note: We are actually breaking the standard MapReduce paradigm here, which says that all the values from a particular key will go to the same Reducer. In this example, which is a very common pattern when analyzing log files, values from the same key (the IP address) will go to multiple Reducers, based on the month portion of the line.

Step 1: Start Eclipse

1. Download the zip file "ex5.zip" from Canvas and unzip it. There are four files after unzipping:

    - MonthPartitioner.java (Partitioner)
    - ProcessLogs.java (driver)
    - CountReducer.java (Reducer)
    - LogMonthMapper.java (Mapper)

2. Review the instructions in step 2 of exercise 3 to create a new java project "partitioner" and a package "stubs" for this exercise.

Then copy all four java files unzipped in the previous step to the eclipse workspace on your computer for the project "partitioner" under the "stubs" folder

(e.g. /Users/linli/eclipse-workspace/partitioner/src/stubs).

Configure the Build Path of the project as how we did in step 2 of exercise 3.

Step 2: Write the Program in Java

Complete the code in the java files.

1. Starting with the LogMonthMapper.java file, write a Mapper that maps a log file output line to an IP/month pair. The Mapper should emit a Text key (the IP address) and Text value (the month). Note that the input file is space-delimited, and the delimiter in the date is "/".

e.g.

Input: 96.7.4.14 - - [24/Apr/2011:04:20:11 - 0400] "GET /cat.jpg HTTP/1.1" 200 12433

Output key: 96.7.4.14

Output value: Apr

2. Write the Reducer

Complete the CountReducer.java file. In a single invocation the *reduce()* method receives a string containing one IP address (the key) along with an iterable collection of Text (the values - e.g. a list of "Jan"), and should emit a single key-value pair: the IP address and the total number of hits from that IP address in that month (e.g. Jan).

3. Write the Partitioner

Modify the MonthPartitioner.java file to create a Partitioner that sends the (key, value) pair to the correct Reducer based on the month.

Remember that the Partitioner receives both the key and value, so you can inspect the value to determine which Reducer to choose.

4. Write the Driver

Modify your driver code to specify that you want 12 Reducers. Configure your job to use your custom Partitioner.

Step 3: Compile and Test your Solution

Build and test your code (review exercise 3 for how to build and test your code). Your output directory should contain 12 files named part-r-000xx. Each file should contain IP address and number of hits for month xx.

Hints:

You may wish to test your code against the smaller version of the access log in the /exercise/testlog directory on HDFS before you run your code against the full log in the /exercise/weblog directory. However, note that the test data may not include all months, so some result files will be empty.