Lab: Writing a 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.

The Problem

In this lab you will counted the number of hits for each different IP address in 12 web log files, one each 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*.

You 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.

Instructions

1. Input Data

   Unpack the access_log.gz file and upload it to HDFS. You can unpack the file with the following command:

   ```
   gunzip access_log.gz
   ```

2. Source code

   You will find the source files in the `src/stubs` directory. You will also find a `src/hints` directory that contains a copy of the source files modified to make the assignment easier to complete, and a `src/solution` directory that contains a copy of the source files modified to complete the assignment.

3. Complete the `ProcessLogs`, `LogMonthMapper`, `CountReducer`, and `MonthPartitioner` classes in the `src/stubs` directory to count the number of occurrences of unique IP address in each month.

4. Assemble and test your solution.

Write the Mapper

1. Starting with the `LogMonthMapper.java` stub file, write a Mapper that maps a log file output line to an IP/month pair.

2. The Mapper should emit a Text key (the IP address) and Text value (the month). 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`

Hint: In the Mapper, you may use a regular expression to parse to log file data if you are familiar with regex processing.

Remember that the log file may contain unexpected data – that is, lines that do not conform to the expected format. Be sure that your code copes with such lines.

Write the Reducer

3. Starting with the `CountReducer.java` stub file, write a Reducer that sums up the number of hits for each IP address.

Write the Partitioner

4. Modify the `MonthPartitioner.java` stub 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.

Modify the Driver

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

Test your Solution

7. Build and test your code. Your output directory should contain 12 files named `part-r-000`*xx*. Each file should contain IP address and number of hits for month *xx*.

Hints

- Write unit tests for your Partitioner!
- You may wish test your code against the smaller version of the access log before testing against the full log file. However, note that the test data may not include all months, so some result files may be empty.