Writing a Partitioner

|  |  |
| --- | --- |
| **Exercise Dir** | ~/workspace/partitioner |
| **Eclipse Proj** | partitioner |
| **Java Files** | MonthPartitioner.java (Partitioner)  ProcessLogs.java (Driver)  CountReducer.java (Reducer)  LogMonthMapper.java (Mapper) |
| **JAR File** | partitioner.jar |

**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 the “More Practice with Writing MapReduce Java Programs” exercise you did previously, you built the code in log\_file\_analysis project. That program counted the number of hits for each different IP address in a web log file. The final output was a file containing a list of IP addresses, and the number of hits from that address.

This time, we want to perform a similar task, but we want the final output to consist of 12 files, one each for each month of the year: January, February, and so on. Each file will contain a list of IP address, 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.

**Write the Mapper**

Starting with the LogMonthMapper.java fixme file, write a Mapper that maps a log file output line to an IP/month pair. The map method will be similar to that in the LogFileMapper class in the log\_file\_analysis project, so you may

wish to start by copying that code.

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. Otherwise we suggest following the tips in the hints code, or just copy the code from the solution package.

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 Partitioner**

Modify the MonthPartitioner.java fixme 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**

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

**Test your Solution**

Build and test your code. Your output directory should contain 12 files named part-r-0000x. Each file should contain IP address and number of hits for month xx.

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

**END**