KTH Royal Institute of Technology **ID2221 Data–Intensive Computing**

Lab 1 Documentation

Andrea Nardelli `andnar@kth.se`

Tianze Wang `tianzew@kth.se`

September 23, 2018

Code commentary

The implemented code can be found in the file `TopTen.java`. Each mapper outputs the top ten records of its split by reputation, and the single reducer outputs the global top ten.

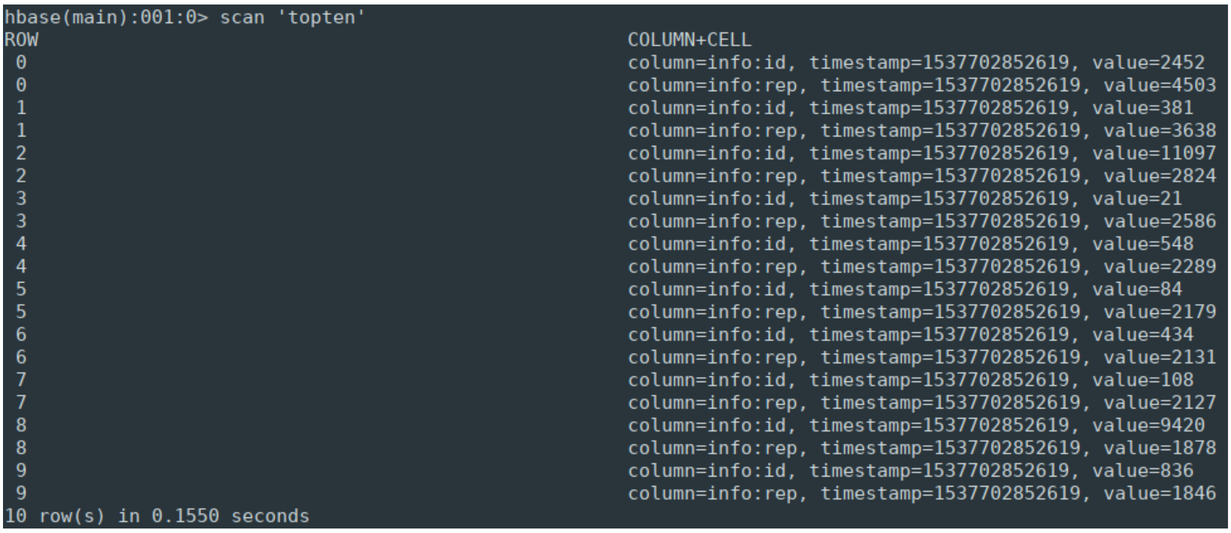
- **Mapper**: the mapper has two functions. First, in the `map` function it receives each line of input, parses it with the provided helper function, discards the parsed XML when it is invalid (if the property map is empty or `Id == null`), and then writes the key–value pairs in its internal data structure with `Reputation` as key and the user record (i.e. its XML line) as value.
  Second, in `cleanup` we emit the top ten records by reputation to the reducer. In order to achieve this, we use the internal properties provided by the `TreeMap` data structure. This map provides ordering on its entries by the natural ordering of the keys of those entries. For integers, the natural ordering is increasing, which means that the last 10 entries stored by the `TreeMap` are the ones with highest reputation. These are emitted to the reducers with 10 calls of the `pollLastEntry()` function.
- **Reducer**: the code of the reducer is very similar to the mapper. It parses the input records from the mappers again (note that the validity of the XML is guaranteed as those records made it through the mapping), extracts the top 10, and writes their `Id` and `Reputation` in the HBase table. We insert the values to HBase as strings in order to have them be human–readable as HBase normally stores bytes.
- **Configuration**: In the `main` function we define the configuration of the MapReduce job. In order we first define the classes of the job, mapper and reducer. Then we define the output type of the intermediate key–value pairs, set the number of reducer tasks with `job.setNumReduceTasks(1)` in order to obtain the global top ten, specify the directory for the input files and specify the output table on HBase.

How to run

In the zip file we have provided a helper bash script called `test.sh` to compile and run the application. It combines step 5&6 of the instructions of the assignment, creating the directory `topten_classes` while assuming the source file to be `topten/TopTen.java`. It will then create a `topten.jar` file which is executed. Note that the namenode, datanode, and HBase must already be running and the table `topten` with column family `info` must already exist.

Results

Figure 1, transcribed in Table 1, shows the output of `scan 'topten'`.



```
hbase(main):001:0> scan 'topten'
ROW                                                       COLUMN+CELL
 0                                                        column=info:id, timestamp=1537702852619, value=2452
 0                                                        column=info:rep, timestamp=1537702852619, value=4503
 1                                                        column=info:id, timestamp=1537702852619, value=381
 1                                                        column=info:rep, timestamp=1537702852619, value=3638
 2                                                        column=info:id, timestamp=1537702852619, value=11097
 2                                                        column=info:rep, timestamp=1537702852619, value=2824
 3                                                        column=info:id, timestamp=1537702852619, value=21
 3                                                        column=info:rep, timestamp=1537702852619, value=2586
 4                                                        column=info:id, timestamp=1537702852619, value=548
 4                                                        column=info:rep, timestamp=1537702852619, value=2289
 5                                                        column=info:id, timestamp=1537702852619, value=84
 5                                                        column=info:rep, timestamp=1537702852619, value=2179
 6                                                        column=info:id, timestamp=1537702852619, value=434
 6                                                        column=info:rep, timestamp=1537702852619, value=2131
 7                                                        column=info:id, timestamp=1537702852619, value=108
 7                                                        column=info:rep, timestamp=1537702852619, value=2127
 8                                                        column=info:id, timestamp=1537702852619, value=9420
 8                                                        column=info:rep, timestamp=1537702852619, value=1878
 9                                                        column=info:id, timestamp=1537702852619, value=836
 9                                                        column=info:rep, timestamp=1537702852619, value=1846
10 row(s) in 0.1550 seconds
```

Figure 1: The contents of the `topten` table from the HBase shell.

Table 1: The output in table format.

Reputation	Id
4503	2452
3638	381
2824	11097
2586	21
2289	548
2179	84
2131	434
2127	108
1878	9420
1846	836