Group 50
Leonardo Remondini 19970128-T512
Haris Poljo 19970731-6213

Lab 1 - Part 1

How to run the code

Go through the following steps to compile and run the code

1. Start the HDFS NameNode and DataNode (if they are not running). Then create a folder input in HDFS, and upload the files in it.

```
$HADOOP_HOME/bin/hdfs --daemon start namenode
$HADOOP_HOME/bin/hdfs --daemon start datanode


$HADOOP_HOME/bin/hdfs dfs -put users.xml
$HADOOP_HOME/bin/hdfs dfs -ls
```

2. Start the HBase and the HBase shell.

```
$HBASE_HOME/bin/start-hbase.sh
$HBASE_HOME/bin/hbase shell
```

3. Create the HBase table topten with one column family info to store the id and reputation of users.

```
create 'topten', 'info'
```

4. Set the environment variables.

```
export HADOOP_CLASSPATH=$($HADOOP_HOME/bin/hadoop classpath)
export HBASE_CLASSPATH=$($HBASE_HOME/bin/hbase classpath)
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$HBASE_CLASSPATH
```

5. compile the code and make and make a jar file.

```
javac -cp $HADOOP_CLASSPATH -d topten_classes topten/TopTen.java

jar -cvf topten.jar -C topten_classes/ .
```

6. Run the application

```
$HADOOP_HOME/bin/hadoop jar topten.jar id2221.topten.TopTen users.xml
topten_output
```

7. Check the result in the HBase shell

```
scan 'topten'
```

Mapping function

1. A line (user) of the input file is readed and converted into String.
2. The String value is taken as input by the `transformXmlToMap()` function, which will map the xml string to the user's entries.
3. Reputation and ID of the user are extracted and added to a TreeMap as a (key, value) pair. TreeMap keeps its entries sorted according to the natural ordering of users' reputation.

Cleanup function

1. The cleanup method gets called once after all key-value pairs have been through the map function. (Each mapper has its own cleanup function)
2. The top 10 records are extracted from the TreeMap and each of them is output as a `new Text (ID+" "+Reputation)` to the reducer.

Reducing function

1. The `Text` received as input is parsed. ID and Reputation are extracted.
2. each Reputation and ID extracted are added to a TreeMap as (key, value). TreeMap keeps its entries sorted according to the natural ordering of users' reputation.
3. When all `(Reputation,ID)` pairs have been added to the TreeMap, the top 10 records are extracted and sent to the `'topten'` table in HBase as output.

Main function

1. HBase configuration is created.
2. The Mapper Class is set.
3. We configure our job to have one reducer only and therefore there will be only one input group for this reducer that will contain all the potential top ten records.
4. The Reduce Class is set and connected to the `'topten'` table of HBase.
5. The output class of the (Key,Value) pair of the Mapper is defined.

HBASE setting

1. HBase configuration is created in the main function
2. The reducer is connected to the `'topten'` table in the main function
3. For each top 10 value extracted from the TreeMap in the reduce function, a new row with two columns (`"rep"` and `"id"`) is added to `'topten'` through a `Put` function which is output to HBase

Results (topten table displayed by HBase)

Rank	USER ID	REPUTATION
1	2452	4503
2	381	3638
3	11097	2824
4	21	2584

5	584	2289
6	84	2179
7	434	2131
8	108	2127
9	9420	1878
10	836	1846