Nicolas Escobar (escobarn)  
Harsh Reddy (hagandav)

Cloud Computing - Project 5 Report  
HBase FreqIndexBuilder

**Deliverables**

Zip your source code, results and report in a file named username project5.zip. Submit this file to the Canvas

submission page.

* Complete source code (**located in src/iu/pti/hbaseapp/clueweb09/FreqIndexBuilderClueWeb09.java)**
* A written report describing the main steps (**current file**)

**Evaluation**

The point total for this project is 3, where the distribution is as follows:

* Completeness of your code and output (2 points) (**output 🡪 output/project2**)
* Correctness of written report (1 points)

**HBase FreqIndexBuilder Application Report**

The main component of the application is located in the ***FreqIndexBuilderClueWeb09.*class**. The flow is indicated below. The portion highlighted in yellow refers to the code that was added as part of the assignment.

**main(String[] args)**

1. Main runs and creates a new instance of the HBase configuration
2. The HBase configuration instance along with the arguments received in main are passed to the configureJob method

**configureJob(Configuration conf, String[] args)**

1. This method creates a new instance of Scan type to specify the columns that will be used to apply the scan in the Mapper.
2. A new Job instance is also created using the configuration instance received. The setJarByClass method defines the class that will be used for the Mapper
3. The initTableMapperJob and initTableReducerJob methods are executed and passed along the details of the tables and other parameters required for the Mapper and Reducer respectively.
4. In this case, only the Mapper will be executed with data coming from the ClueWeb09DataTable. The argument for the Reducer class will be NULL but the output of the Mapper will be written to ClueWeb09IndexTable.
5. Finally the number of reducers is specified (as mentioned 0 in this case) for the job and the instance is returned

**FibMapper  
map(ImmutableBytesWritable rowKey, Result result, Context context)**

1. When the map method is executed, the data from the *clueWeb09DataTable* in HBase will be pulled from the result instance based on the defined column family and qualifier.
2. The content will be converted from bytes to String and passed to the *getWordFreq* method to get a HashMap of words and frequencies for the content that is being processed.
3. For each <word, frequency> pair contained in the HashMap, we create a new HBase Put object which will have the word as the rowkey.
4. Then, we add to the rowkey: the column family (frequencies), the qualifier (docId) and the value (frequency).
5. Finally, we write the put object to the context so it will be inserted to the clueWeb09IndexTable as specified in the initTableReducerJob
6. If we take one of the sample values from the output file, such as the one below, we noticed the following:
   1. 0,5px is the rowkey or word for which we are calculating the inverted index
   2. The four rows (00xxx200871 to 00xxx200874) are the different docIDs where the instance of this particular word appears with their corresponding frequencies (4 in this case)

------------0,5px------------

00000200871 : 4

00000200872 : 4

00000200873 : 4

00000200874 : 4

The sample output from clueWeb09IndexTable is shown below (complete sample file can be found in **output/project2.txt**)

scanning table clueWeb09IndexTable on frequencies...

------------0'1------------

00000230265 : 1

------------0'23.08------------

00000235243 : 1

------------0,0.00,1,0.00------------

00000118373 : 1

------------0,0.00,1,0.00,2,0.00------------

00000118369 : 1

00000118370 : 1

00000118371 : 1

00000118372 : 1

------------0,0.00,1,0.00,2,0.00,3,0.00,4,0.00,5,0.00,6,0.00,7,0.00,8,0.00,9,0.00------------

00000118368 : 1

------------0,01euros------------

00000226930 : 1

------------0,1.7,5.0------------

00000231836 : 1

------------0,28804,1690753\_1690758\_1693514,00------------

00000121800 : 1

------------0,4458,360183\_395924,00------------

00000121979 : 1