Nicolas Escobar (escobarn)  
Harsh Reddy (hagandav)

Cloud Computing - Project 4 Report  
HBase Word Count

**Deliverables**

Zip your source code and report in a file named username project4.zip

**Evaluation**

The point total for this project is 1.5, where the distribution is as follows:

* Correctness of your code and output (1 points)
* Completeness of written report (0.5 points)
* The report should explain the logic behind your code

**The main component of the application is located in the *WordCountClueWeb09* 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 both the Mapper and the Reducer.
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. Finally the number of reducers is specified for the job and instance is returned

**WcMapper  
map(ImmutableBytesWritable row, 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.
3. For each <word, frequency> pair contained in the HashMap, we write the pair to the context so it will be passed to the Reducer to calculate the final word count and load the result back to HBase

**WcReducer**

**reduce(Text word, Iterable<LongWritable> freqs, Context context)**

1. The reducer will receive a list of frequencies for each word
2. For each word, we will sum up all the frequencies contained in the *freqs* list of frequencies and store the result in a *totalFreq* variable
3. We create an instance of the Put object and pass the word as the rowkey
4. Then, we add the column family (frequencies), qualifier (count) and value (*totalFreq* in bytes) to the recently created Put object
5. Finally, we write the Put object as a value to the context so it will apply the PUT command in HBase to the target table defined previously in *initTableReducerJob* method and with the parameters defined when creating and setting the Put object