CLOUD COMPUTING

Project 5: HBase FreqIndexBuilder

Abhishek Naik and Shree Govind Mishra

17th March, 2017

HBase FreqIndexBuilder is a modified and advanced version of the WordCount program. It not only stores the number of occurrences of each word, but also the related document name (identification number) as HBase Inverted Index record. These are built for supporting efficient searches in huge files. The flow and the main steps carried out as a part of the project have been highlighted in the text below:

In this project, we have edited the FreqIndexBuilderClueWeb09.java file. This file has three main functions:

1. The *main()* function:

The main() function has been provided as a standard implementation. We have not changed this function and has been used as, in the code.

The main() function creates an object of the Configuration and Job class.

The object of the Configuration class is passed as a parameter to the object of the Job class.

1. The *configureJob()* function:

As the name suggests, this function is used for configuration of the Job.

It creates an object of the Scan class to specify the columns that would be used.

A new Job object is also created. The *setJarByClass()* method specifies the class that would be used as the mapper.

The *initTableMapperJob* and *initTableReducerJob* are passed the details of the columns of the tables, along with some other parameters. These are required for the Mapper and the Reducer, respectively.

Also, in our case, the number of Reduce tasks has been set to 0, using the function *setNumReduceTasks().*

1. The *map()* function:

In this function, the data from the HBase table *ClueWeb09DataTable* is retrieved using the object of the Result class.

This data is then converted into String using the *toString()* method.

Later, we call the *getTermFreqs()* to create a HashMap *hm*. This *getTermFreqs()* function counts the frequencies of terms in the content.

Then, for each <word, frequency> pair contained in the hashmap, we create a new *Put* object.

This object has the word as the row’s key.

Then we use the *add()* function to add the column family, the qualifier and the values.

In the end, we write the *Put* class’ object to context so that it would be inserted into the *ClueWeb09DataTable*.