B649 Cloud Computing Project 4 Report

HBase Word Count

Team Members: Yash Ketkar (ykketkar@indiana.edu) | Neelam Tikone (ntikone@indiana.edu)

HBase Word Count Logic:

Map:

Input is: <Row, Content >

Output is: <Word, Frequency>

Code Snippet:

```
HashMap<String, Long> wordFreqs = getWordFreq(content);

for (Map.Entry<String, Long> w: wordFreqs.entrySet()) {
      context.write(new Text(w.getKey()), new LongWritable(w.getValue()));
}
```

Logic: Here the key is the rowkey of an HBase record related to a specified URI, and the content is the stored text of that URI. We then use the getWordFreq method which makes use of Apache Lucene to return a HashMap of words with their frequencies. We then write these values of word along with their frequency as <Text, LongWritable> to the context.

Reduce:

Input is: <Word, Frequency>

Output is: < ImmutableBytesWritable, Put>

Code Snippet:

```
long totalFreq = 0;

for (LongWritable f: freqs) {
      totalFreq += f.get();
}

Put p = new Put(Bytes.toBytes(word.toString()));
p.add(Constants.CF_FREQUENCIES_BYTES, Constants.QUAL_COUNT_BYTES,
Bytes.toBytes(totalFreq));
context.write(null, p);
```

Logic: Here we receive each word along with its frequency. We then add all the intermediate frequencies from multiple map tasks. We will then write the sum to an HBase table with the put operation which contain the information of each word.

Output:

```
scanning table WordCountTable on frequencies...

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

count : 1

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

count : 1

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

count : 1

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

count : 4

------------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------------

count : 1

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

count : 1

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

count : 1

------------0,28804,1690753_1690758_1693514,00------------

count : 1

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

count : 1

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

count : 16
```