CSCI 415: Networking and Parallel Computation
Due Thursday November 9^{th} @ 11:59PM

Bonus Programming Assignment: MapReduce

The objectives of this assignment are:

1. Understanding the steps involved in designing a parallel program using the Hadoop MapReduce framework.
2. Writing a program in Java for counting words in a large file.

1 Description

In this assignment you will write a program in the Hadoop MapReduce framework that counts the number of times words appear in a large file. The mapper in the current WordCount example output a list of key-value pairs such that the value is always one. For example, if the text has
"the wind will wind the windmill"
the mapper in the original example will output:
(the, 1), (wind, 1), (will, 1), (wind, 1), (the, 1), (windmill, 1)

The new mapper will output:

(the, 2), (wind, 2), (will, 1), (windmill, 1)

For your reference, here is the code in the map function:

```
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();

public void map(LongWritable key, Text value, OutputCollector<Text,
                IntWritable> output, Reporter reporter) throws IOException {
  String line = value.toString();
  StringTokenizer tokenizer = new StringTokenizer(line);
  while (tokenizer.hasMoreTokens()) {
```

```
    word.set(tokenizer.nextToken());
    output.collect(word, one);
  }
}
```

You only have to change the code in the Mapper and the whole program should work fine and output the same counts as the original example.

Output: The output will be stored in a file in the output directory.

2 Hints:

1. You can build your code starting from the WordCount example that we have covered in class.
2. You only have to change the Map class. Use the two files (file01, and file02).

3 Submissions:

You only have to submit the mapper code. You can submit a text version of the code or the .java file.