CS235 Assignment Phase 2

Krupa Hegde - 862006278

Problem Definition:

- Compute and output the number of conferences per city and compute the top 10 locations.
- Output the list of conferences per city.
- For each conference regardless of the year (e.g., KDD), output the list of cities.
- For each city compute and output a time series of number of conferences per year
- Visualize the output of Q4 using heatmap.

Hadoop :

Hadoop is an open source framework by Apache to process large data set or Big Data set. Storage part in Hadoop is the Hadoop Distributed File System(HDFS). Important modules of Hadoop are Hadoop Commons, HDFS, Hadoop Yarn and Hadoop Map Reduce. Hadoop Commons has libraries and jars required by other Hadoop modules. Yarn is used to manage computing resources and to schedule user applications.

MapReduce:

Hadoop MapReduce is a programming model for processing large data in parallel on cluster. MapReduce has jobtracker and task tracker for each cluster. Map Reduce splits the data to smaller chunks which are processed by map tasks. Then the map output is sorted and it is given to the reduce tasks. MapReduce uses <key ,value> pairs for its operation.

Implementation :

I have setup the Hadoop in given lab machine using the commands provided by the link [2]

- Compute and output the number of conferences per city and compute the top 10 locations.

  To solve this, I have modified the word count code given in [1]. Fig 1 shows the modification of mapper class. The input data is in tab separated format where first column contains acronym second contains conference name and last column is the location of the conference. To compute number of conferences per city, I have split the data to get different colums using the split function of java. Mapper splits each data as key value pair eg, a city name may occur for different conferences so the mapper output would be <'abc' , 1> for conference 'xyz' held in 'abc' and <'abc', '1'> for 'kdd' held in 'abc' . these output are given to the reducer.

```
16 public class AssignQ1 {
17
18       //mapper implementation
19     public static class TokenizerMapper
20         extends Mapper<Object, Text, Text, IntWritable>{
21
22       private final static IntWritable one = new IntWritable(1);
23       //variable to hold key
24       private Text word = new Text();
25
26       public void map(Object key, Text value, Context context
27                       ) throws IOException, InterruptedException {
28
29           //split the data to access different columns
30           String[] splitStr=value.toString().split("\t");
31           //set location for the key
32           word.set(splitStr[2]);
33           context.write(word, one);
34
35       }
36     }
```

Fig. 1

```
37  //reducer implementation
38    public static class IntSumReducer
39        extends Reducer<Text,IntWritable,Text,IntWritable> {
40      private IntWritable result = new IntWritable();
41
42      public void reduce(Text key, Iterable<IntWritable> values,
43                         Context context
44                         ) throws IOException, InterruptedException {
45          //variable to count the conferences
46
47        int sum = 0;
48        for (IntWritable val : values) {
49          sum += val.get();
50        }
51        result.set(sum);
52        //output key, value
53        context.write(key, result);
54      }
55    }
56
```

Fig 2

Output of mapper is taken by reducer which counts the number of conferences per city. That is it counts the values per key. Fig 2 shows the modifications for reduce class.

I have sorted the values of this output using the linux command to see the top 10 cities.

Command - Sort -t$'\t' -k2 -nr output1.txt

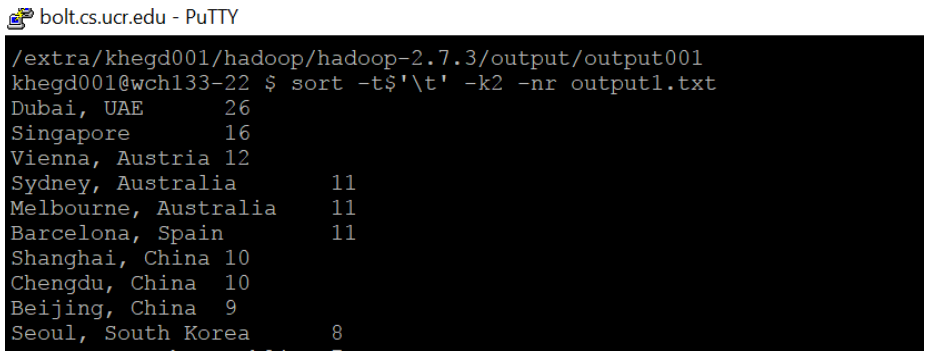

Fig 3

Fig 3 shows the top 10 cities and number of conferences.

- Output the list of conferences per city.

  To get the conferences per city, key should be the location and values are the conferences.

```
15
16  public class AssignQ2 {
17  //mapper implementation
18    public static class TokenizerMapper
19         extends Mapper<Object, Text, Text, Text>{
20
21      private final static IntWritable one = new IntWritable(1);
22      private Text word = new Text();
23          private Text word1 =new Text();
24          private Text word2 = new Text();
25      public void map(Object key, Text value, Context context
26                      ) throws IOException, InterruptedException {
27
28          //split the input to different columns
29          String[] splitStr=value.toString().split("\t");
30          word1.set(splitStr[0]);
31          //get keys -cities
32          word.set(splitStr[2]);
33
34          context.write(word, word1);
35
36      }
37    }
```

Fig 4

The procedure is similar as the first problem. But here value is not the integer so define a Text variable to store values. Fig 4 shows the code snippet for the mapper class. And Fig 5 shows the code snippet for reducer class. Value is not the integer so IntWritable has to be changed to Text.

```
37    }
38  //reducer implementation
39    public static class IntSumReducer
40          extends Reducer<Text,Text,Text,Text> {
41      private Text result= new Text();
42
43      public void reduce(Text key, Iterable<Text> values,
44                         Context context
45                         ) throws IOException, InterruptedException {
46          // group the conferences for each city
47          String sum="";
48        for (Text val : values) {
49            sum +=val.toString();
50            if(val.toString()!=" " || val.toString()!="\n" || val.toString()!="  ")
51                    sum+=",";
52          }
53        //result values
54          result.set(sum);
55
56          //output key values - reducer output
57          context.write(key, result);
58        }
59      }
60  //configurations
```

Fig 5

- For each conference regardless of the year (e.g., KDD), output the list of cities.
  In this, we need conference acronym as the key and location of the conference is the value. Here as per the question we have to remove year from the acronym. Fig 6 shows the code snippet for mapper and reducer classes. In mapper we need first part of the acronym because the last part contains year. So in order to remove the year form 1st column I have used substring method of java.

```
17    public static class TokenizerMapper
18          extends Mapper<Object, Text, Text, Text>{
19
20    //  private final static IntWritable one = new IntWritable(1);
21      //variables to store key value data
22        private Text word = new Text();
23          private Text word1 =new Text();
24  private Text word2 = new Text();
25      public void map(Object key, Text value, Context context
26                      ) throws IOException, InterruptedException {
27
28                  // split the input by tab delimiter
29                  String[] splitStr=value.toString().split("\t");
30                  //get 1st column data and exclude year
31                  word2.set(splitStr[0].substring(0,splitStr[0].length()-4));
32                  //get 3rd column
33                  word.set(splitStr[2]);
34                  //set and pass key values to reduces
35                  context.write(word2, word);
36
37          }
38    }
39
40    public static class IntSumReducer
41          extends Reducer<Text,Text,Text,Text> {
42      private Text result = new Text();
43
44      public void reduce(Text key, Iterable<Text> values,
45                         Context context
46                         ) throws IOException, InterruptedException {
47          //concatinate values per city
48          String sum="";
49            for (Text val : values) {
50              sum +=val.toString()+",";
51            }
52            result.set(sum);
53            //output the key value result
54            context.write(key, result);
55          }
56        }
```

Fig 6

- For each city compute and output a time series of number of conferences per year
  In this we have a composite key as we have to select number of conferences on each city and year together. The key would be the combination of city and year. So I have added year and cities in same Text variable and given it as the key and value is the integer. Fig 7 shows the code snippet for mapper and reducer class.

```
public class AssignQ4 {

    //mapper class implementation
  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

      //for value
      private final static IntWritable one = new IntWritable(1);
      private Text word = new Text();

      public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
          // split the input by tab delimiter
           String[] splitStr=value.toString().split("\t");
           //extract city and year from the data
           word.set(splitStr[2]+" "+splitStr[0].substring(splitStr[0].length()-4,splitStr[0].length()));
           //set the composite key contained in word
           context.write(word, one);
        }
    }

  //reducer implementation
  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    //integer varible to store result
      private IntWritable result = new IntWritable();

      public void reduce(Text key, Iterable<IntWritable> values,
                         Context context
                         ) throws IOException, InterruptedException {
          //count the values for each key
          int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      //output the count per key
      context.write(key, result);
    }
  }
  //configurations
```

Fig 7

To get the year from 1st column again I have used substring function and along with this added cities to get the key. Reducer can count the values that is number of conferences based on this key.

- Visualization :

  To visualize the the output of 4th problem in a heatmap I have used D3.js

  Heatmap: A type of visualization which represents data values contained in matrix using different colors.

  D3. Js is javascript library for visualization . I have used D3.js Because it is easy to implement and is user friendly. D3 can take json data, csv data or tsv data using the function d3.json() or d3.csv . As it is in Java script I have embedded it within HTML code. The basic Idea is in HTML if we want to draw something on the webpage we use svg element . Every element in svg is available in Dom. So we need to draw the heatmap on svg. We have to define X and Y axes. Important things to consider to the size of svg. Then define range of scales and map the values. Tickformat can also be given. I have created an svg element to hold all the elements that axis function produces. These can be grouped so that translating and transforming will be easier. Using .call() operator, axis can be called. For the heatmap color range is set so from lower value to a higher value the color differs. Blue color is for the lower values representing less conferences Red color is for higher values representing more conferences in that city on a given year. The data is less and for most of the data, number of conference is less that is 1. Usually the domain is selected

using smaller value and the higher values. Since most of the data are one to show the differences I have given the domain[1,7] so number of conferences being 1 will take blue value between the 7 would take darker blue and near 7 and values greater than 7 take red color.

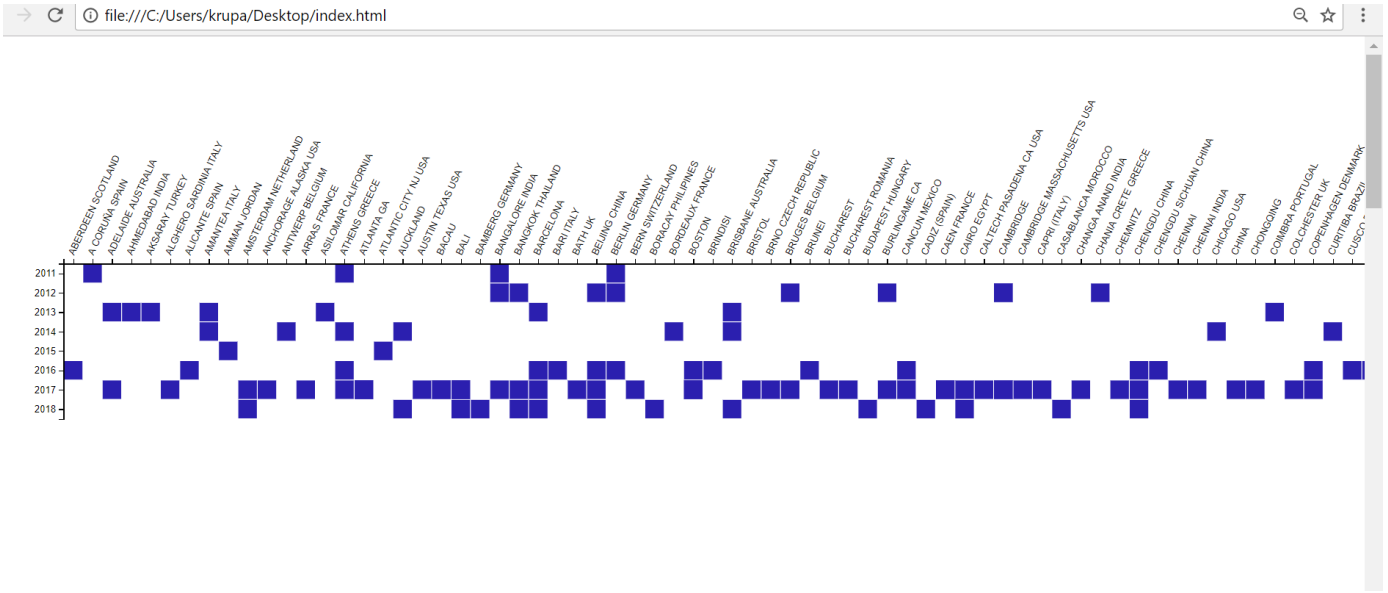

Fig 8

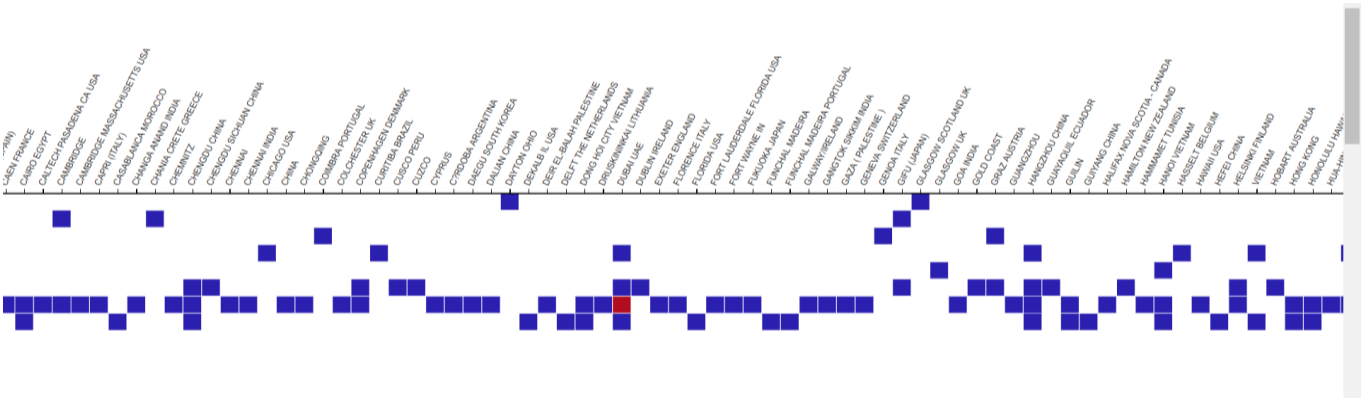

Fig 9

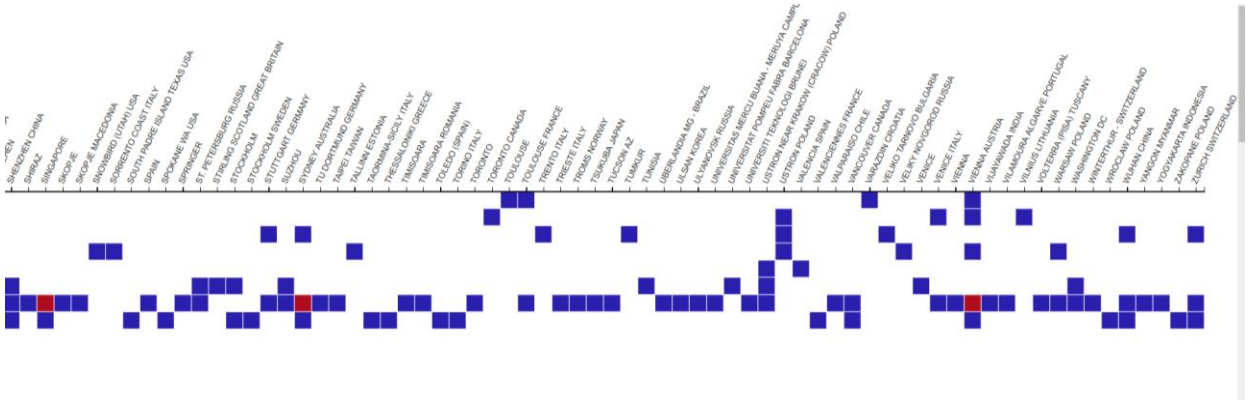

Fig 10

From the figures it is clear that there are very less number of cities with more number of conferences in a year. For the year 2017 the number of conferences are more and from years 2011 -15 the number of conferences were less.

Files attached:

I have attached the output files.

Java program files

D3.js File

Data.csv used for visualisation

Reference:

[1] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

[2] https://hadoop.apache.org/docs/r2.7.0/hadoop-project-dist/hadoop-common/SingleCluster.html

[3] https://bl.ocks.org/Bl3f/cdb5ad854b376765fa99