

## Big Data Programming – Assignment 3

Varaprasad Kurra

Panther ID - #002430487

### Task – I

Code changes in the final *HadoopPageRank* Java Project.

We added a final Mapper java class named *HadoopPageRankResultMapper* to the existing Project. The sole purpose of this class is to sort the results of the final output from the last iteration. A few changes are also made to the existing *HadoopPageRank.java* that contains the configurations of the other Initial, Main and Final Mapper and Reducer classes.

#### *Changes in the HadoopPageRank.java*

```
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class HadoopPageRank extends Configured implements Tool {

    public int run(String[] args) throws Exception
    {
        int iteration = 0;
        Path inputPath = new Path(args[0]);
        Path basePath = new Path(args[1]);
        FileSystem fs = FileSystem.get(getConf());
        fs.delete(basePath, true);
        fs.mkdirs(basePath);
        Job initJob = Job.getInstance(getConf(), "pageRank");
        initJob.setJarByClass(HadoopPageRank.class);
        initJob.setMapperClass(HadoopPageRankInitMapper.class);
        initJob.setReducerClass(HadoopPageRankInitReducer.class);
        initJob.setOutputKeyClass(Text.class);
        initJob.setOutputValueClass(Text.class);
        initJob.setInputFormatClass(TextInputFormat.class);

        Path outputPath = new Path(basePath, "iteration_" + iteration);
        FileInputFormat.addInputPath(initJob, inputPath);
        FileOutputFormat.setOutputPath(initJob, outputPath);

        if ( !initJob.waitForCompletion(true) ) {

            return -1;
        }

        int totalIterations = Integer.parseInt(args[2]);
```

```

while ( iteration < totalIterations ) {
    iteration = iteration +1 ;
    inputPath = outputPath;
    outputPath = new Path(basePath, "iteration_" + iteration);
    Job mainJob = Job.getInstance(getConf(),"Iteration " + iteration);
    mainJob.setJarByClass(HadoopPageRank.class);
    mainJob.setMapperClass(HadoopPageRankMainJobMapper.class);
    mainJob.setReducerClass(HadoopPageRankMainJobReducer.class);
    mainJob.setOutputKeyClass(Text.class);
    mainJob.setOutputValueClass(Text.class);
    mainJob.setInputFormatClass(TextInputFormat.class);
    FileInputFormat.setInputPaths(mainJob, inputPath);
    FileOutputFormat.setOutputPath(mainJob, outputPath);

    if ( !mainJob.waitForCompletion(true) )
    {
        return -1;
    }
}

// collect the result, highest rank first - you will need to finish this up
Job resultJob = Job.getInstance(getConf(),"final result");
resultJob.setJarByClass(HadoopPageRank.class);
resultJob.setMapperClass(HadoopPageRankResultMapper.class);
resultJob.setOutputKeyClass(Text.class);
resultJob.setOutputValueClass(Text.class);
FileInputFormat.setInputPaths(resultJob, outputPath);
FileOutputFormat.setOutputPath(resultJob,new Path(basePath, "result"));

if (!resultJob.waitForCompletion(true))
{
    return -1;
}

return 0;
}

public static void main(String[] args) throws Exception
{
    int exitCode = ToolRunner.run(new HadoopPageRank(), args);
    System.exit(exitCode);
}
}

```

### ***HadoopPageRankResultMapper (Sorting Stage)***

```

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Mapper.Context;

import java.io.IOException;

public class HadoopPageRankResultMapper extends Mapper<LongWritable, Text,
Text, Text>
{

```

```

public void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {

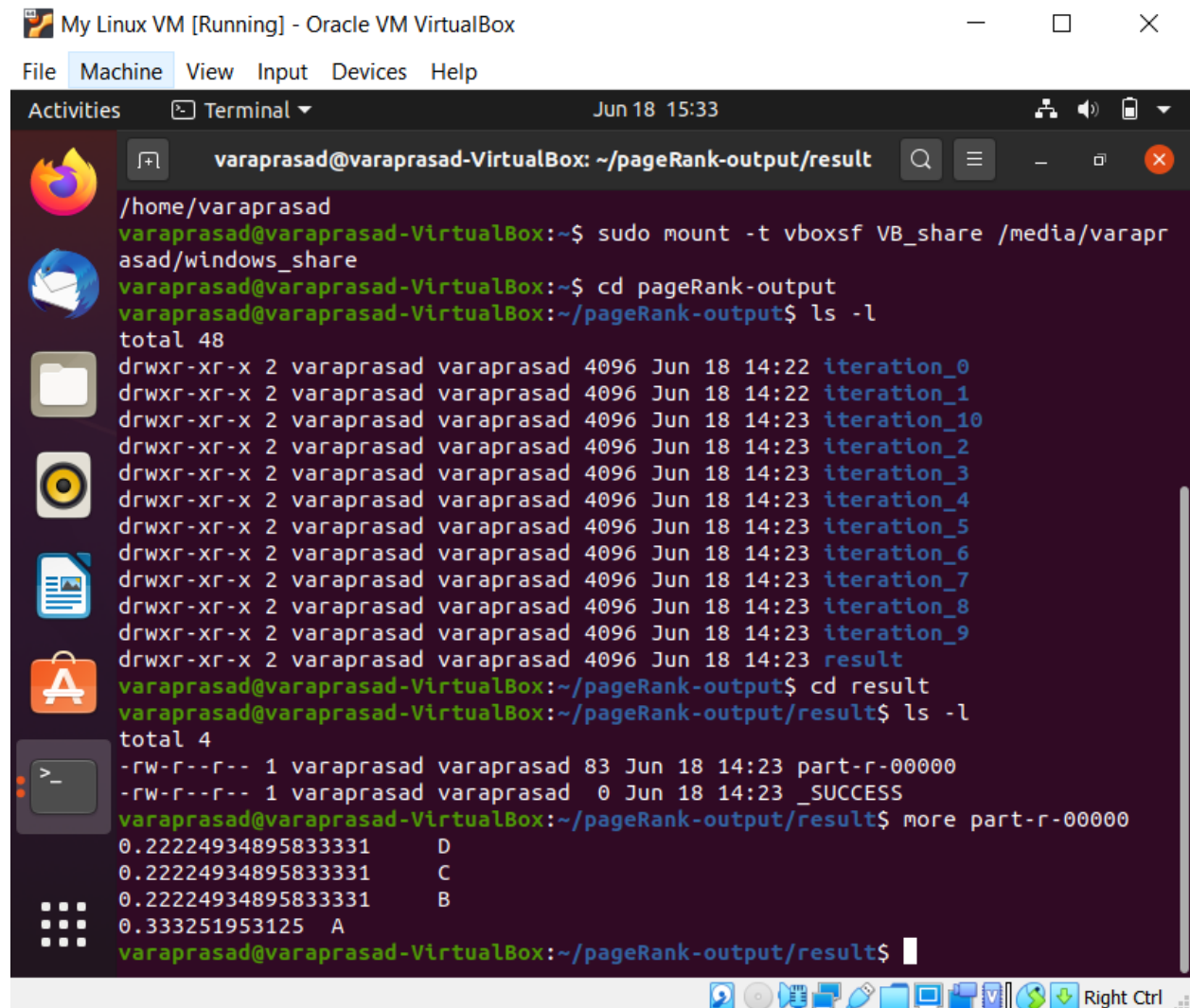
    if ( value == null || value.charAt(0) == '#' ) {
        return;
    }

    int tabIdx1 = value.find("\t");

    String nodeA = Text.decode(value.getBytes(), 0, tabIdx1 );
    String nodeB = Text.decode(value.getBytes(), tabIdx1 + 1,
value.getLength() - (tabIdx1 + 1));
    String [] pageResult = nodeB.split("\t");
    String result = pageResult[0];
    context.write(new Text(result), new Text(nodeA));

}
}

```



The screenshot shows a terminal window titled "My Linux VM [Running] - Oracle VM VirtualBox". The terminal is running as the user "varaprasad" in the directory "~/pageRank-output/result". The user has executed the following commands and received the following output:

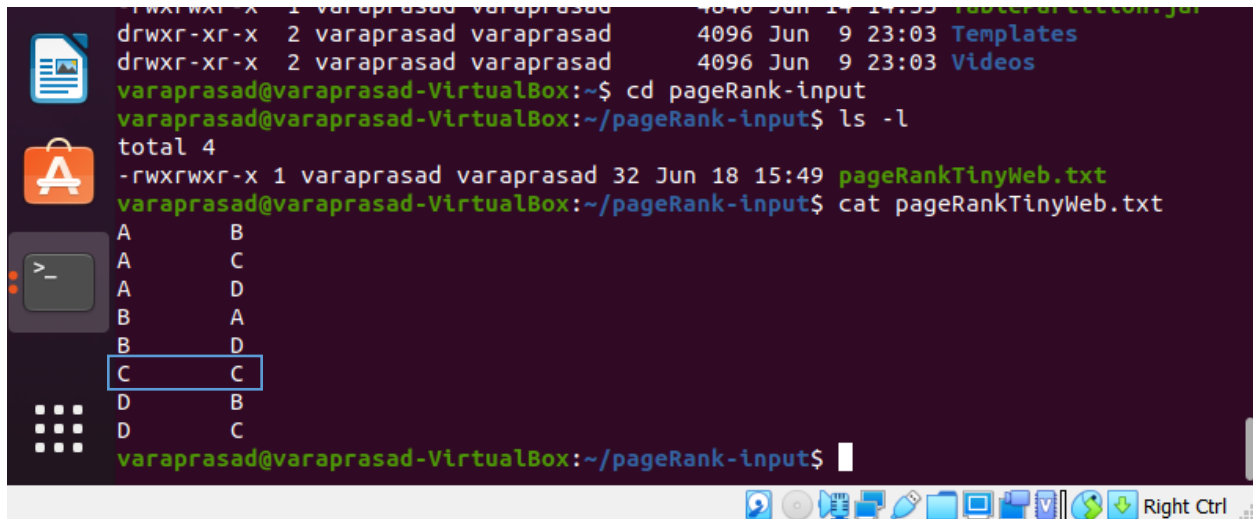
```

varaprasad@varaprasad-VirtualBox: ~/pageRank-output/result
varaprasad@varaprasad-VirtualBox:~$ sudo mount -t vboxsf VB_share /media/varaprasad/windows_share
varaprasad@varaprasad-VirtualBox:~$ cd pageRank-output
varaprasad@varaprasad-VirtualBox:~/pageRank-output$ ls -l
total 48
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:22 iteration_0
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:22 iteration_1
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 iteration_10
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 iteration_2
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 iteration_3
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 iteration_4
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 iteration_5
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 iteration_6
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 iteration_7
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 iteration_8
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 iteration_9
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 14:23 result
varaprasad@varaprasad-VirtualBox:~/pageRank-output$ cd result
varaprasad@varaprasad-VirtualBox:~/pageRank-output/result$ ls -l
total 4
-rw-r--r-- 1 varaprasad varaprasad 83 Jun 18 14:23 part-r-00000
-rw-r--r-- 1 varaprasad varaprasad 0 Jun 18 14:23 _SUCCESS
varaprasad@varaprasad-VirtualBox:~/pageRank-output/result$ more part-r-00000
0.22224934895833331 D
0.22224934895833331 C
0.22224934895833331 B
0.333251953125 A
varaprasad@varaprasad-VirtualBox:~/pageRank-output/result$

```

## Task 2:

To avoid the spider-trap, we need to alter the calculations in the MainReducer Job. This is the part of the we are actually passing the results to the result phase. For observing the spider trap We need to change the input so that the we inject the Spider-Trap.



The screenshot shows a terminal window with the following commands and output:

```
varaprasad@varaprasad-VirtualBox:~$ cd pageRank-input
varaprasad@varaprasad-VirtualBox:~/pageRank-input$ ls -l
total 4
-rwxrwxr-x 1 varaprasad varaprasad 32 Jun 18 15:49 pageRankTinyWeb.txt
varaprasad@varaprasad-VirtualBox:~/pageRank-input$ cat pageRankTinyWeb.txt
A      B
A      C
A      D
B      A
B      D
C      C
D      B
D      C
```

The input file contains a directed graph with 4 nodes (A, B, C, D). The edges are: A to B, A to C, A to D, B to A, B to D, C to C (self-loop), D to B, and D to C. The self-loop on node C is highlighted with a red box, indicating the spider-trap.

We made the spider-trap by modifying the direction from C – C.

Modifications in the `HadoopPageRankMainJobReducer.java` file

```
import java.io.IOException;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class HadoopPageRankMainJobReducer extends Reducer<Text, Text, Text, Text>
{
    public void reduce(Text key, Iterable<Text> values, Context context) throws
    IOException, InterruptedException
    {
        double beta = 0.8;
        if ( values == null )
        {
            return;
        }

        String links = "";
        double receivedContribution = 0.0;

        for (Text value : values) {

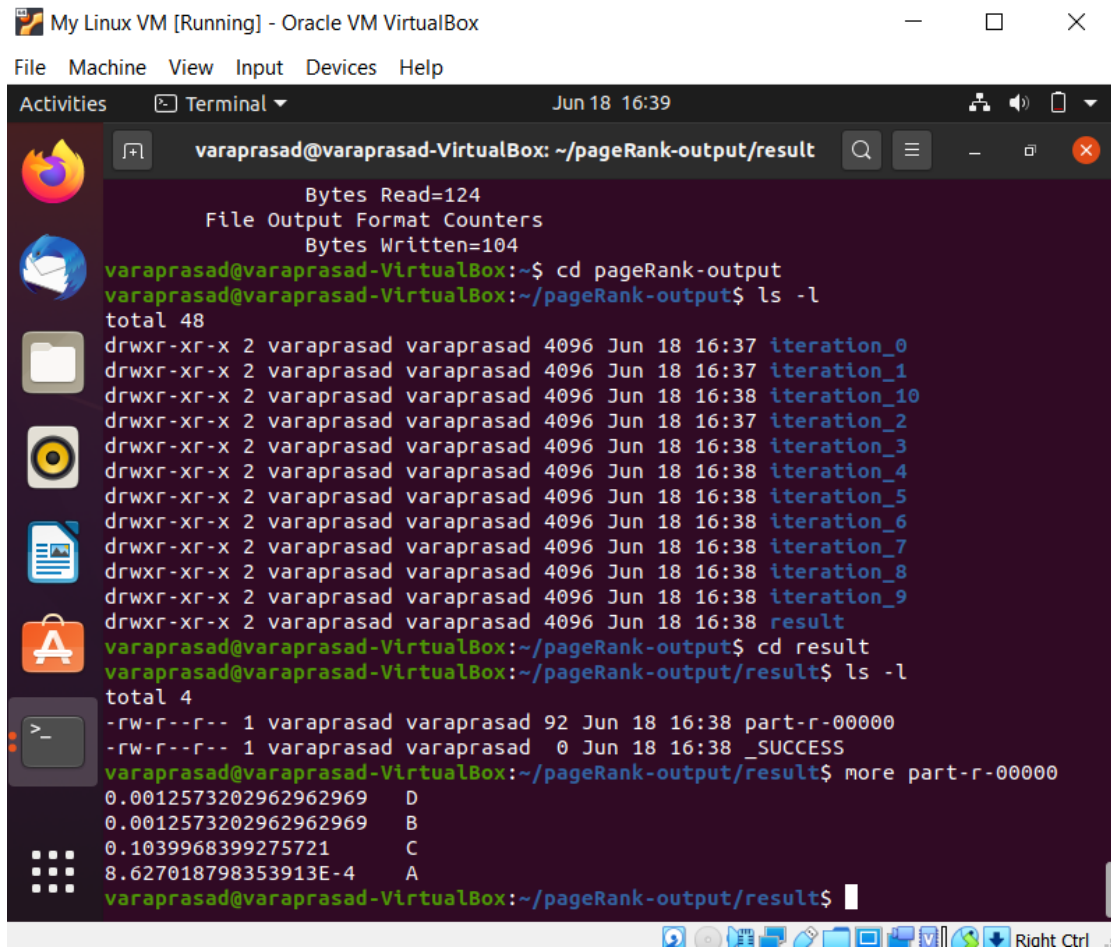
            String content = value.toString();
            if (content.startsWith("|"))
            {
                links += content.substring("|".length());
            }
        }
    }
}
```

```

        else
        {
            receivedContribution += Double.parseDouble(content);
        }
    }

    double newPageRank = ((receivedContribution* beta) + (1-beta)*(1/4));
    context.write(key, new Text(newPageRank + "\t" + links));
}
}

```

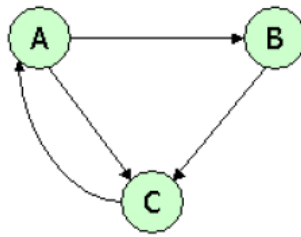


```

My Linux VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jun 18 16:39
varaprasad@varaprasad-VirtualBox: ~/pageRank-output/result
Bytes Read=124
File Output Format Counters
Bytes Written=104
varaprasad@varaprasad-VirtualBox:~$ cd pageRank-output
varaprasad@varaprasad-VirtualBox:~/pageRank-output$ ls -l
total 48
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:37 iteration_0
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:37 iteration_1
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:38 iteration_10
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:37 iteration_2
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:38 iteration_3
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:38 iteration_4
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:38 iteration_5
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:38 iteration_6
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:38 iteration_7
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:38 iteration_8
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:38 iteration_9
drwxr-xr-x 2 varaprasad varaprasad 4096 Jun 18 16:38 result
varaprasad@varaprasad-VirtualBox:~/pageRank-output$ cd result
varaprasad@varaprasad-VirtualBox:~/pageRank-output/result$ ls -l
total 4
-rw-r--r-- 1 varaprasad varaprasad 92 Jun 18 16:38 part-r-00000
-rw-r--r-- 1 varaprasad varaprasad 0 Jun 18 16:38 _SUCCESS
varaprasad@varaprasad-VirtualBox:~/pageRank-output/result$ more part-r-00000
0.0012573202962962969 D
0.0012573202962962969 B
0.1039968399275721 C
8.627018798353913E-4 A
varaprasad@varaprasad-VirtualBox:~/pageRank-output/result$

```

**Question -2:** Given TinyWeb is as below.



Hence from the above Tiny Web can build our Stochastic Column Matrix that describes the probabilities of the next page to be visited from the initial.

Observed Stochastic column Matrix is: 3\*3 Matrix

$$\begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \end{bmatrix}$$

Given  $a=b=c=1$ . So the column matrix i. e the initial PageRank 3\*1 Matrix is  $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

So, the First iteration gives us the 3\*1 Matrix that is:  $\begin{bmatrix} 1 \\ 1/2 \\ 3/2 \end{bmatrix}$

After 2<sup>nd</sup> iteration the resultant Matrix is:  $\begin{bmatrix} 3/2 \\ 1/2 \\ 1 \end{bmatrix}$

After 3<sup>rd</sup> iteration the obtained Matrix Is  $\begin{bmatrix} 1 \\ 3/4 \\ 5/4 \end{bmatrix}$

After 4<sup>th</sup> iteration  $\begin{bmatrix} 5/4 \\ 1/2 \\ 5/4 \end{bmatrix}$

After the 5<sup>th</sup> iteration the resultant column matrix is  $\begin{bmatrix} 5/4 \\ 5/8 \\ 5/4 \end{bmatrix}$

Hence the answer would be  $b = \frac{1}{2}$ . Option C.