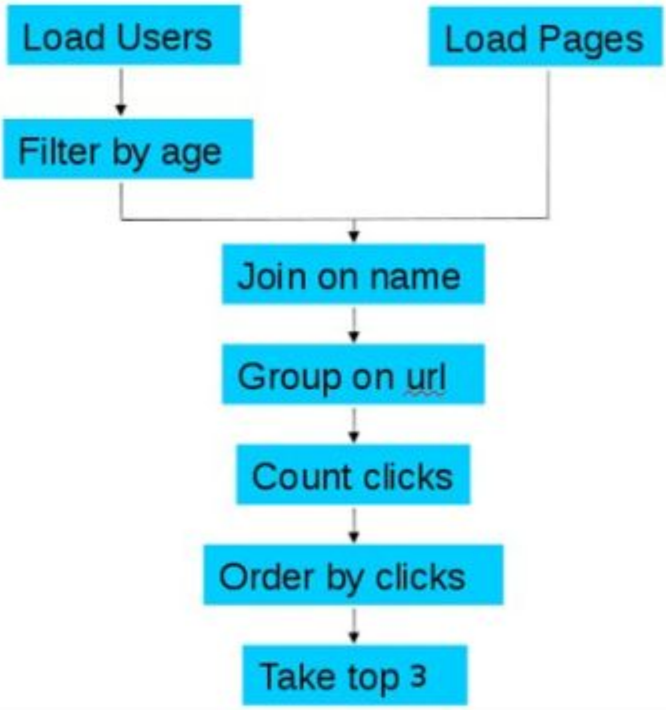
HW6   ECE677

Name: CAN ZENG

Inputs: User data in a file and Website data in another file

Outputs: Top3 most visited pages by users aged between 18 - 25



1. Approaches and Implementation

Create two samples files: SampleUsers.txt, SampleWebpages.txt

Create the HW6.java

The Following executing steps showed:

```
#Using psftp to upload the file
open jaspardzc@aim.engr.arizona.edu 2224
lcd C:/Users/jaspe_000/Desktop/677HW6/
put README.txt
put SampleUsers.txt
put SampleWebpages.txt
put Users.txt
put Webpages.txt

#Step0:
ssh aim.engr.arizona.edu 2224
mkdir -p workspace/wordcount
nano WordCount.java
cd ..
cd ..
```

```
#Step1-transfer:
hfs -put SampleUsers.txt /users/jaspardzc/
hfs -put SampleWebpages.txt /users/jaspardzc/

#Step2-testing the example
javac -classpath `hadoop classpath` -d workspace/wordcount workspace/wordcount/WordCount.java
jar -cvf workspace/wordcount/WordCount.jar -C workspace/wordcount/ .
hadoop jar workspace/wordcount/WordCount.jar org.apache.hadoop.mapreduce.WordCount /users/jaspardzc/SampleUsers.txt /users/jaspardzc/SampleUsers.txt_out
hfs -cat /users/jaspardzc/SampleUsers.txt_out/part-r-00000

#Step3-start
mkdir -p HW/HW6
touch test.txt
hfs -put test.txt /users/jaspardzc/
hfs -ls /users/jaspardzc/


#Step4-compile:
javac -classpath `hadoop classpath` -d HW/HW6 HW/HW6/HW6.java
jar -cvf HW/HW6/HW6.jar -C HW/HW6/ .
hadoop jar HW/HW6/HW6.jar HW6

#hint: alternatively, can add above three commands to a script file name EXE, and use “sh EXE” to execute the whole process would save a lot of time

#Step5-check the result:
hfs -cat /users/jaspardzc/output/JoinedOnName/part-r-00000
hfs -cat /users/jaspardzc/output/GroupByURL/part-r-00000
hfs -cat /users/jaspardzc/output/FilteredWebpages.txt/part-m-00000
hfs -cat /users/jaspardzc/output/FilteredUsers.txt/part-m-00000
hfs -cat /users/jaspardzc/output/SoringByClicks/part-r-00000

#Step6-get and store the result in local node:
hfs -get /users/jaspardzc/output
```

2. Results and Analysis

Results are contained in the output file.

```
cd output/SortingByClicks
more part-r-00000
```

The results showed the TOP3 most visited pages by users aged between 18 - 25.

```
  chron.com/        27
  google.cn/        23
  spotify.com/      22
```

#More details are included in the output folder

3. Source Code

Source code is contained in the src file.

*********HW6.java*********

```
import java.io.IOException;
import java.util.StringTokenizer;
import java.util.ArrayList;
import java.util.Date ;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.input.KeyValueTextInputFormat;


public class HW6 {

//Load Users, filtered by age
public static class UsersMapper extends Mapper<Object, Text, Text, Text>{

         public void map(Object key, Text UsersFile, Context context
         ) throws IOException, InterruptedException {

           String inputstring = UsersFile.toString();
           int space_tag = inputstring.indexOf(" ");

           String Username = inputstring.substring(0, space_tag);
           String Ages = inputstring.substring(space_tag + 1);
           int age = Integer.parseInt(Ages);

           if(age>=18 && age<=25)
           {
           context.write(new Text(Username), new Text("0 " + Ages));
```

```
                }
        }
}

//Check whether Users has been mapped or not, if not go to join
public static class CheckMapper extends Mapper<Text, Text, Text, Text>{
            public void map(Text name, Text value, Context context
               ) throws IOException, InterruptedException {
                  context.write(name, value);
      }
  }


//Load Webpages
public static class WebpagesMapper extends Mapper<Object, Text, Text, Text>{
            public void map(Object key, Text WebpagesFile, Context context
               ) throws IOException, InterruptedException {

                  String inputstring = WebpagesFile.toString();
                  int space_tag = inputstring.indexOf(" ");
                  String Username = inputstring.substring(0, space_tag);
                  String Webs = inputstring.substring(space_tag + 1);
                  context.write(new Text(Username), new Text("1 " + Webs));
      }
  }


//Join on name
public static class JoinOnName extends Reducer<Text,Text,Text,Text> {

      public void reduce(Text name,Iterable <Text> values,Context context)
                  throws IOException, InterruptedException {
                     ArrayList<String> UserList= new ArrayList<String>();
                     ArrayList<String> WebpagesList = new ArrayList<String>();
                           for(Text tempvalues :values) {
                              String stringvalue = tempvalues.toString();
                              if(stringvalue.charAt(0) == '0')
                                       UserList.add(stringvalue.substring(1));
                              else
                                       WebpagesList.add(stringvalue.substring(1));
                           }

                 for(String Tempage:UserList)
                 {
```

```
             for(String Tempweb:WebpagesList)
                 {
                    String tempJoin = name+"|"+Tempage +"|"+Tempweb;
                    context.write(new Text("NULL"), new Text(tempJoin));
                 }
         }

    }
 }


//Seperate the joined data
 public static class SepJoinedData extends Mapper<Text, Text, Text, IntWritable>{

         private final static IntWritable one = new IntWritable(1);
          public void map(Text nullkey, Text JoinedData, Context context
        ) throws IOException, InterruptedException {
          String inputstring = JoinedData.toString();
          int firstcommaIndex = inputstring.indexOf('|')+1;
          int SecondcommaIndex = inputstring.indexOf('|',firstcommaIndex)+1;
          String Webs = inputstring.substring(SecondcommaIndex);
          context.write(new Text(Webs), one);
   }
 }

//sum of numbers of same Webpages, count clicks
public static class CountClicks extends Reducer<Text,IntWritable,Text,IntWritable> {

   private IntWritable result = new IntWritable();

   public void reduce(Text Webs,Iterable <IntWritable> values,Context context)
          throws IOException, InterruptedException {
                  int sum = 0;
         for (IntWritable val : values) {
              sum += val.get();
          }
          result.set(sum);
          context.write(Webs, result);

    }
}

//Load in the numbers of Webpages
 public static class LoadWebpagesNum extends Mapper<Text, Text, IntWritable, Text>{
      public void map(Text name, Text clicks, Context context
        ) throws IOException, InterruptedException {
          String Sclicks = clicks.toString();
          // From top to bottom
          context.write(new IntWritable(-Integer.parseInt(Sclicks)), name);
   }
 }

//Sorting, order by clicks
 public static class WebpagesSorting extends Reducer<IntWritable,Text,Text,IntWritable> {

   int count=0;
   public void reduce(IntWritable Clicks,Iterable <Text> values,Context context)
          throws IOException, InterruptedException {

        for (Text val : values) {
          if(count<3){
                  int result = -1*Clicks.get();
                  String Sclicks = result + "";
                  context.write(val, new IntWritable(Integer.parseInt(Sclicks)));
           }
          count++;
    }
  }
}
//output the final result, top3 Webpages clicks

//main function
  public static void main(String[] args) throws Exception {
    long start = new Date().getTime();
    Configuration conf0 = new Configuration();
//Load in Users, filtered by age
        Job UsersFilteredJob = Job.getInstance(conf0, "UsersMapper");
        UsersFilteredJob.setJarByClass(HW6.class);
        UsersFilteredJob.setMapperClass(UsersMapper.class);
        UsersFilteredJob.setInputFormatClass(TextInputFormat.class);
        UsersFilteredJob.setMapOutputKeyClass(Text.class);
        UsersFilteredJob.setMapOutputValueClass(Text.class);
        UsersFilteredJob.setOutputKeyClass(Text.class);
        UsersFilteredJob.setOutputValueClass(Text.class);
        FileInputFormat.addInputPath(UsersFilteredJob,                      new
Path("/users/jaspardzc/Users.txt"));
        FileOutputFormat.setOutputPath(UsersFilteredJob,                    new
Path("/users/jaspardzc/output/FilteredUsers.txt"));
```

```
            UsersFilteredJob.setNumReduceTasks(0);
            UsersFilteredJob.waitForCompletion(true);
//Load in Webpages
            Job WebpagesJob = Job.getInstance(conf0, "WebpagesJob");
            WebpagesJob.setJarByClass(HW6.class);
            WebpagesJob.setMapperClass(WebpagesMapper.class);
            WebpagesJob.setInputFormatClass(TextInputFormat.class);
            WebpagesJob.setMapOutputKeyClass(Text.class);
            WebpagesJob.setMapOutputValueClass(Text.class);
            WebpagesJob.setOutputKeyClass(Text.class);
            WebpagesJob.setOutputValueClass(Text.class);
            FileInputFormat.addInputPath(WebpagesJob,                                   new
Path("/users/jaspardzc/Webpages.txt"));
            FileOutputFormat.setOutputPath(WebpagesJob,                                 new
Path("/users/jaspardzc/output/FilteredPages.txt"));
            WebpagesJob.setNumReduceTasks(0);
            WebpagesJob.waitForCompletion(true);

//Join on Name
            Job JoinOnNameJob = Job.getInstance(conf0, "JoinOnName");
            JoinOnNameJob.setJarByClass(HW6.class);
            JoinOnNameJob.setMapperClass(CheckMapper.class);
            JoinOnNameJob.setReducerClass(JoinOnName.class);
            JoinOnNameJob.setInputFormatClass(KeyValueTextInputFormat.class);
            JoinOnNameJob.setMapOutputKeyClass(Text.class);
            JoinOnNameJob.setMapOutputValueClass(Text.class);
            JoinOnNameJob.setOutputKeyClass(Text.class);
            JoinOnNameJob.setOutputValueClass(Text.class);
            FileInputFormat.addInputPath(JoinOnNameJob,                                 new
Path("/users/jaspardzc/output/FilteredUsers.txt/part-m-00000"));
            FileInputFormat.addInputPath(JoinOnNameJob,                                 new
Path("/users/jaspardzc/output/FilteredWebpages.txt/part-m-00000"));
            FileOutputFormat.setOutputPath(JoinOnNameJob,                               new
Path("/users/jaspardzc/output/JoinedOnName"));
            JoinOnNameJob.waitForCompletion(true);

//Group on URL: Seperate Joined Data && Webpages reducing
            Job URLMapReduceJob = Job.getInstance(conf0, "URLMapReduce");
            URLMapReduceJob.setJarByClass(HW6.class);
            URLMapReduceJob.setMapperClass(SepJoinedData.class);
            URLMapReduceJob.setReducerClass(CountClicks.class);
            URLMapReduceJob.setInputFormatClass(KeyValueTextInputFormat.class);
            URLMapReduceJob.setMapOutputKeyClass(Text.class);
            URLMapReduceJob.setMapOutputValueClass(IntWritable.class);
```

```
		URLMapReduceJob.setOutputKeyClass(Text.class);
		URLMapReduceJob.setOutputValueClass(IntWritable.class);
		FileInputFormat.addInputPath(URLMapReduceJob,                    new
Path("/users/jaspardzc/output/JoinedOnName/part-r-00000"));
		FileOutputFormat.setOutputPath(URLMapReduceJob,                  new
Path("/users/jaspardzc/output/GroupByURL"));
		URLMapReduceJob.waitForCompletion(true);

//Sorting the top3 Webpages
		Job WebpagesSortingJob = Job.getInstance(conf0, "LoadInWebpagesNum");
		WebpagesSortingJob.setJarByClass(HW6.class);
		WebpagesSortingJob.setMapperClass(LoadWebpagesNum.class);
		WebpagesSortingJob.setReducerClass(WebpagesSorting.class);
		WebpagesSortingJob.setInputFormatClass(KeyValueTextInputFormat.class);
		WebpagesSortingJob.setMapOutputKeyClass(IntWritable.class);
		WebpagesSortingJob.setMapOutputValueClass(Text.class);
		WebpagesSortingJob.setOutputKeyClass(Text.class);
		WebpagesSortingJob.setOutputValueClass(IntWritable.class);
		FileInputFormat.addInputPath(WebpagesSortingJob,
new Path("/users/jaspardzc/output/GroupByURL/part-r-00000"));
		FileOutputFormat.setOutputPath(WebpagesSortingJob,
new Path("/users/jaspardzc/output/SortingByClicks"));
		WebpagesSortingJob.waitForCompletion(true);

		long end = new Date().getTime();
		System.out.println("The total time elapsed: "+(end-start) + "ms");

		}
/*
//Also output the bot3 Webpages
	Job WebpagesSortingJob = Job.getInstance(conf0, "LoadWebpagesNum");

*/

}
//Done
```