## BDA | LAB 4

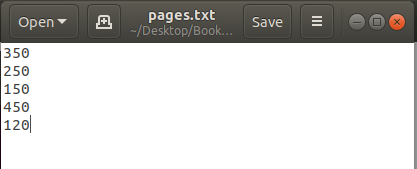
**Aim: Write a map-reduce program to count the frequencies of words from a distributed storage source and understand the phases involved in map-reduce programming.**

**To practice Map Reduce programming in Hadoop using Java Coding.**

**Step. 1.1: Create a file named 'pages.txt' in the local file system. Store line by line content**

**as shown below. Each line data represents the number of pages of a sample book.**

* **350**
* **250**
* **150**
* **450**
* **120**

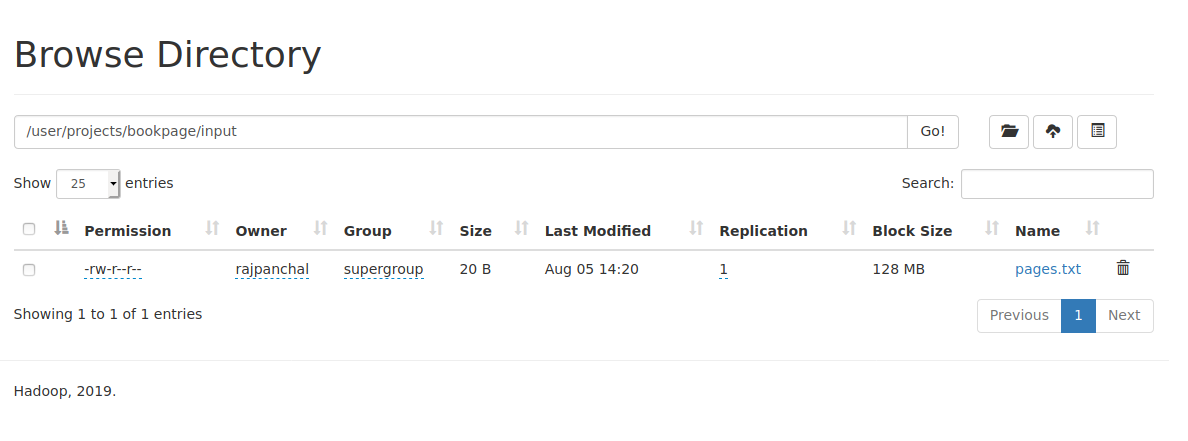
****

**Step. 1.2: Put the file from the local file system to hdfs with a folder named 'input'. Confirm the presence of above data.**

Use following command :

****

Confirm the presence of above data.



**Step. 1.3: Write map and reduce functions to split the books into the following two**

**categories:**

**(a) Big Books**

**(b) Small Books**

**Books which have more than 300 pages should be in the big book category.**

**Books which have less than 300 pages should be in the small book category.**

**Count the number of books in each category.**

**Step : 1**

Code : BookCount.java

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.\*;

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;

public class BookCount

{

public static class BookMapper extends Mapper < LongWritable,

Text, Text, IntWritable >

{

private final static IntWritable one = new IntWritable(1);

private Text book = new Text();

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

{

int x = Integer.parseInt(value.toString());

if(x>300)

{

book.set("Big Books");

context.write(book, one);

}

else

{

book.set("Small Books");

context.write(book, one);

}

}

}

public static class SumReducer extends Reducer < Text, IntWritable,

Text, IntWritable >

{

private IntWritable result = new IntWritable();

public void reduce(Text key, Iterable < IntWritable > values, Context context) throws

IOException, InterruptedException

{

int total = 0;

for (IntWritable val: values)

{

total += val.get();

}

result. set (total);

context. write(key, result);

}

}

public static void main(String[] args) throws Exception

{

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "Book count");

job. setJarByClass(BookCount.class);

job. setMapperClass(BookMapper.class);

job. setCombinerClass(SumReducer.class);

job. setReducerClass(SumReducer.class);

job. setOutputKeyClass(Text.class);

job. setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

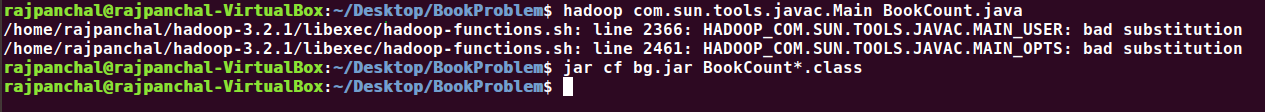
System. exit (job.waitForCompletion(true) ? 0 : 1);

}

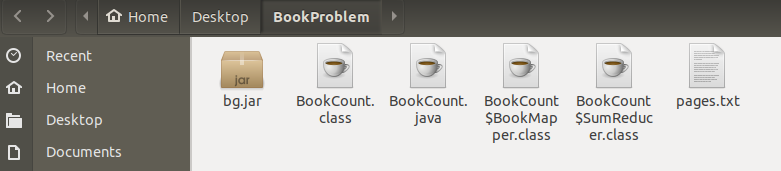
}

**Step 2:**

Run the following command:

****

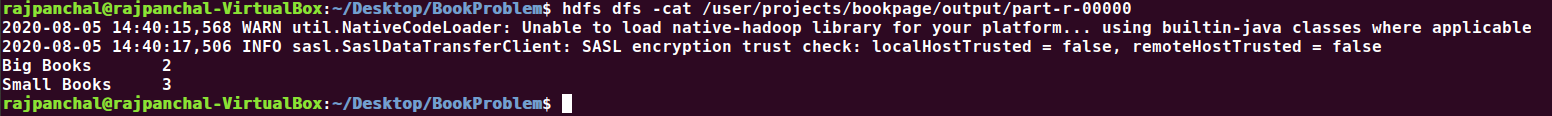
Locate the folder and you may see below files.



Run jar file on Hadoop using below command.



See the output of the Map reduce program by using cat command.



You can also see the output by Web GUI.

