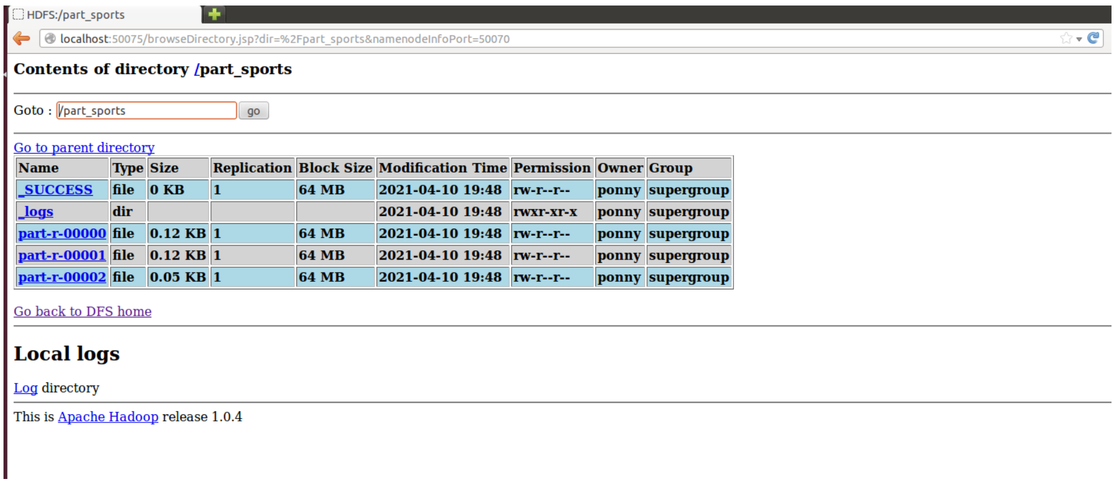
Lab - 5.4.2021

Name: Desai Dhyani Dhaval

19bce1218

1. The dataset contains the person information which has the name, country, and sports liked. Partition the dataset based on the sport name. Write the name, country and sport name details in the files. Also display the count of the records in every partition.



HDFS:/part_sports

localhost:50075/browseDirectory.jsp?dir=%2Fpart_sports&namenodeInfoPort=50070

Contents of directory /part_sports

Goto : /part_sports go

Go to parent directory

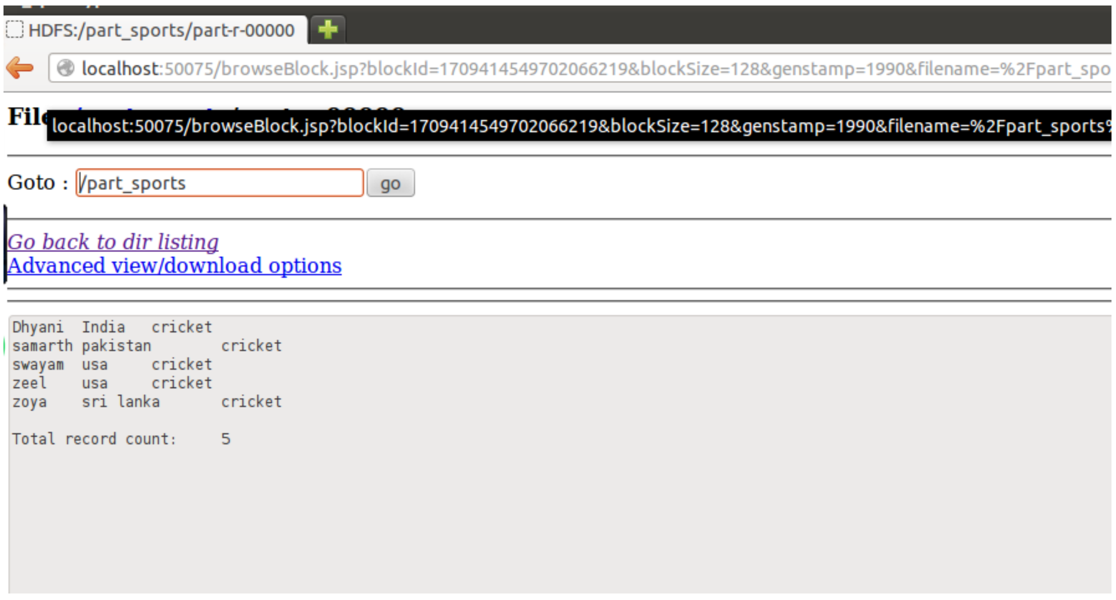
Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
_SUCCESS	file	0 KB	1	64 MB	2021-04-10 19:48	rw-r--r--	ponny	supergroup
_logs	dir				2021-04-10 19:48	rwxr-xr-x	ponny	supergroup
part-r-00000	file	0.12 KB	1	64 MB	2021-04-10 19:48	rw-r--r--	ponny	supergroup
part-r-00001	file	0.12 KB	1	64 MB	2021-04-10 19:48	rw-r--r--	ponny	supergroup
part-r-00002	file	0.05 KB	1	64 MB	2021-04-10 19:48	rw-r--r--	ponny	supergroup

Go back to DFS home

Local logs

Log directory

This is Apache Hadoop release 1.0.4



HDFS:/part_sports/part-r-00000

localhost:50075/browseBlock.jsp?blockId=1709414549702066219&blockSize=128&genstamp=1990&filename=%2Fpart_spo

Fil

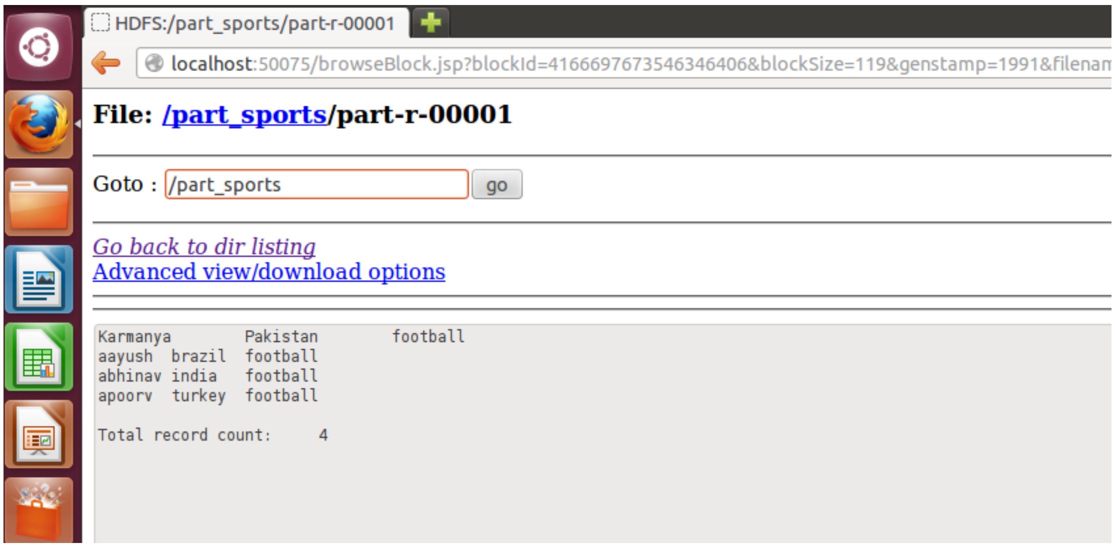
localhost:50075/browseBlock.jsp?blockId=1709414549702066219&blockSize=128&genstamp=1990&filename=%2Fpart_sports

Goto : /part_sports go

Go back to dir listing
Advanced view/download options

```
Dhyani  India   cricket
samarth pakistan        cricket
swayam  usa     cricket
zeel    usa     cricket
zoya    sri lanka       cricket

Total record count:     5
```


HDFS:/part_sports/part-r-00001
localhost:50075/browseBlock.jsp?blockId=4166697673546346406&blockSize=119&genstamp=1991&filenam
go

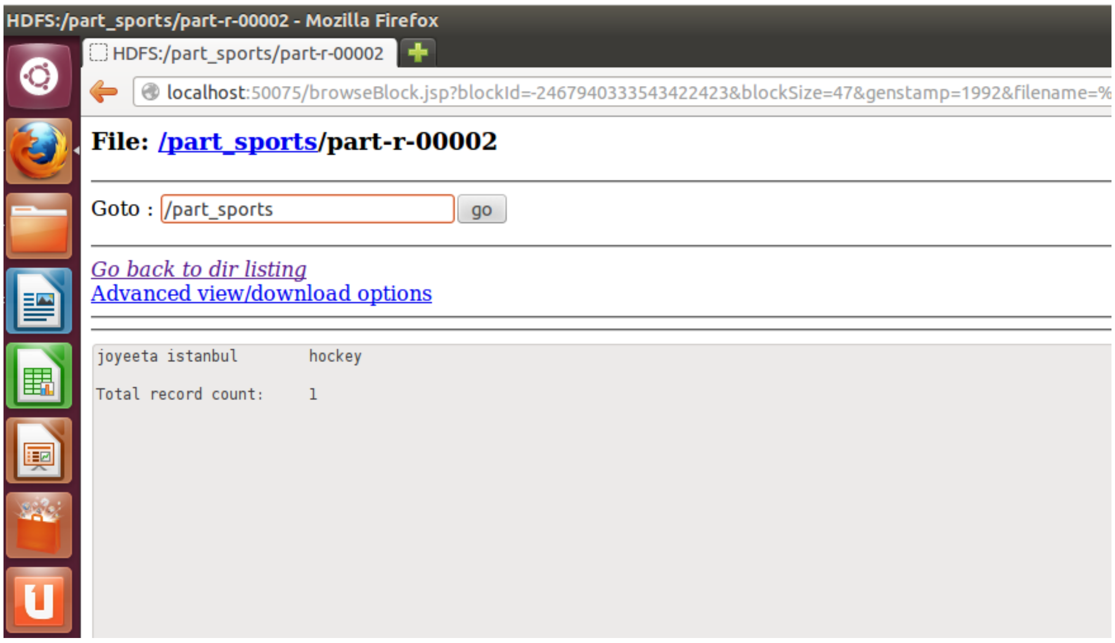
File: /part_sports/part-r-00001

Goto : /part_sports

Go back to dir listing
Advanced view/download options

```
Karmanya        Pakistan        football
aayush  brazil  football
abhinav india   football
apoorv  turkey  football

Total record count:     4
```


HDFS:/part_sports/part-r-00002 - Mozilla Firefox
HDFS:/part_sports/part-r-00002
localhost:50075/browseBlock.jsp?blockId=-2467940333543422423&blockSize=47&genstamp=1992&filename=%
go

File: /part_sports/part-r-00002

Goto : /part_sports

Go back to dir listing
Advanced view/download options

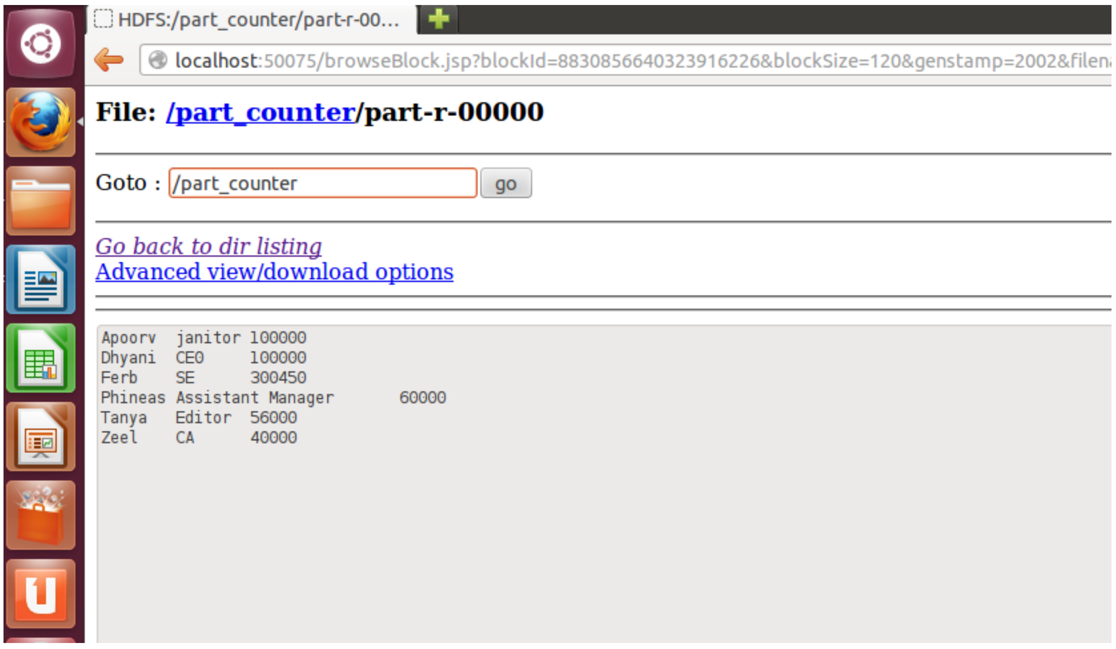
```
joyeeta istanbul        hockey

Total record count:     1
```

CODE

```
import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.Reducer.Context;
import org.apache.hadoop.mapreduce.lib.input.*;
import org.apache.hadoop.mapreduce.lib.output.*;
public class sports {
public static class Map extends Mapper<LongWritable, Text, Text,
Text>
{
public void map(LongWritable key, Text value, Context context) throws
IOException,
InterruptedException {
String[] line = value.toString().split(",");try{
context.write(new Text(line[0]+"\t"+line[1]), new Text(line[2]));
}
catch(Exception e){
}
}
}
public static class dpart3 extends Partitioner<Text,Text>
{
 public int getPartition(Text key,Text value,int nr)
 {
 String a=value.toString();
if(a.equalsIgnoreCase("cricket"))
 return 0;
if(a.equalsIgnoreCase("football"))
 return 1;
 else
 return 2;
 }}
public static class Reduce extends Reducer<Text, Text, Text, Text> {
int a=0;
public void reduce(Text key, Iterable<Text> values, Context context)
throws IOException, InterruptedException {
for(Text b:values){
context.write(key,b);
a=a+1;}
}
public void cleanup(Context context)throws IOException,
InterruptedException
{
context.write(new Text("\nTotal record count:"),new
Text(String.valueOf(a)));
 }
}
public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
Job job = new Job(conf, "sports");
job.setJarByClass(sports.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(Text.class);
job.setMapperClass(Map.class);
job.setReducerClass(Reduce.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
job.setPartitionerClass(dpart3.class);
job.setNumReduceTasks(3);
job.waitForCompletion(true);
}
}
```

2. Salary details are maintained in the text file. It has name, experience (0-20), designation and salary (10,000 to 1,00,000). Create the user defined counters. Count the number of persons having more than 10 years of experience and write the details of name, designation and salary satisfying the condition in HDFS. Also count the number of persons earning the salary greater than 30,000.


HDFS:/part_counter/part-r-00...
localhost:50075/browseBlock.jsp?blockId=8830856640323916226&blockSize=120&genstamp=2002&filen
Goto :
go
Go back to dir listing
Advanced view/download options

File: /part_counter/part-r-00000

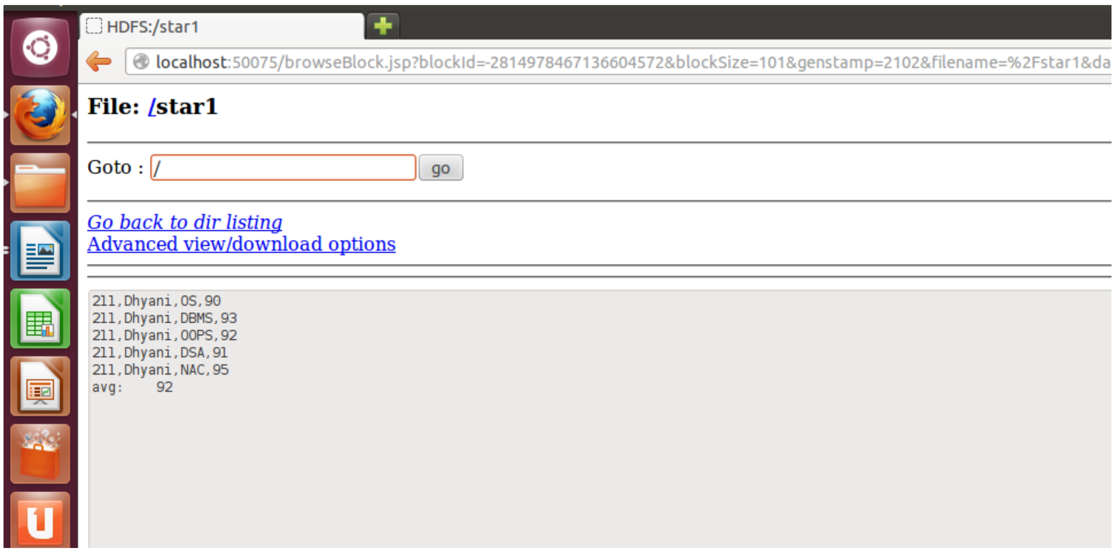
Goto : /part_counter

```
Apoorv  janitor 100000
Dhyani  CEO     100000
Ferb    SE      300450
Phineas Assistant Manager       60000
Tanya   Editor  56000
Zeel    CA      40000
```

CODE

```
import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.Reducer.Context;
import org.apache.hadoop.mapreduce.lib.input.*;
import org.apache.hadoop.mapreduce.lib.output.*;
public class count {
public enum ct
  {
  count_exp_more_10,sal_gr8r_30000;
};
public static class Map extends Mapper<LongWritable, Text, Text,
IntWritable>
{
public void map(LongWritable key, Text value, Context context) throws
IOException,
InterruptedException {
String[] line = value.toString().split(",");
int exp=Integer.parseInt(line[1]);
int i=Integer.parseInt(line[3]);
if(exp>10)
{
context.getCounter(ct.count_exp_more_10).increment(1);
context.write(new Text(line[0]+"\t"+line[2]),new IntWritable(i));
}
if(i > 30000)
{
 context.getCounter(ct.sal_gr8r_30000).increment(1);
 }}
}
public static class Reduce extends Reducer<Text, IntWritable, Text,
IntWritable> {
int b=0;String a,c;
Text wrd1=new Text();
IntWritable res1=new IntWritable();
public void reduce(Text key, IntWritable values, Context context)
throws IOException, InterruptedException {
context.write(key, values);
}
}
public static void main(String[] args) throws Exception {
Configuration conf = new Configuration();
Job job = new Job(conf, "count");
job.setJarByClass(count.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
job.setMapperClass(Map.class);
job.setReducerClass(Reduce.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
FileInputFormat.addInputPath(job, new
Path(args[0]));FileOutputFormat.setOutputPath(job, new Path(args[1]));
job.waitForCompletion(true);
Counters cn=job.getCounters();
cn.findCounter(ct.count_exp_more_10).getValue();
cn.findCounter(ct.sal_gr8r_30000).getValue();
}
}
```

3. Create a text file for students details(student id,student name, course name, marks for 5 courses) in HDFS. Get a particular student details using side data configuration. Display the student details in hdfs. Also display the total and average of the student's marks in hdfs.


HDFS:/star1
localhost:50075/browseBlock.jsp?blockId=-2814978467136604572&blockSize=101&genstamp=2102&filename=%2Fstar1&da
Goto : /
go
Go back to dir listing
Advanced view/download options

File: /star1

```
211,Dhyani,OS,90
211,Dhyani,DBMS,93
211,Dhyani,OOPS,92
211,Dhyani,DSA,91
211,Dhyani,NAC,95
avg:    92
```

Code

```
import java.util.Scanner;
import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.*;
import org.apache.hadoop.mapreduce.lib.output.*;
public class student {
  public static class Map extends Mapper<LongWritable, Text, Text,
IntWritable>
 {

        public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
            String[] line = value.toString().split(",");
            String name1;
            name1 =line[0]+ context.getConfiguration().get("n1")+line[2];

            IntWritable b = new IntWritable(Integer.parseInt(line[2]));

            context.write(new Text(name1),b);
```

```
      }
 }



   public static class Reduce extends Reducer<Text, IntWritable, Text,
IntWritable> {
        int avg=0;
          public void reduce(Text key, Iterable<IntWritable>value,
Context context) throws IOException, InterruptedException {
               int sum=0,cnt=0;

               for (IntWritable val : value) {
                   sum += val.get();
                   cnt++;
                   context.write(key,val);
                }
                avg=sum/cnt;



         }


   public void cleanup(Context context)throws IOException,
InterruptedException
   {
     context.write(new Text("avg: "), new IntWritable(avg));
   }
   }


   public static void main(String[] args) throws Exception {
         Configuration conf = new Configuration();
         conf.set("n1",args[2]);
         Job job = new Job(conf, "partion");
         job.setJarByClass(student.class);
         job.setOutputKeyClass(Text.class);
         job.setOutputValueClass(IntWritable.class);
         job.setMapperClass(Map.class);
         job.setReducerClass(Reduce.class);
         job.setInputFormatClass(TextInputFormat.class);
         job.setOutputFormatClass(TextOutputFormat.class);
         FileInputFormat.addInputPath(job, new Path(args[0]));
         FileOutputFormat.setOutputPath(job, new Path(args[1]));
         job.waitForCompletion(true);
     }
     }
```