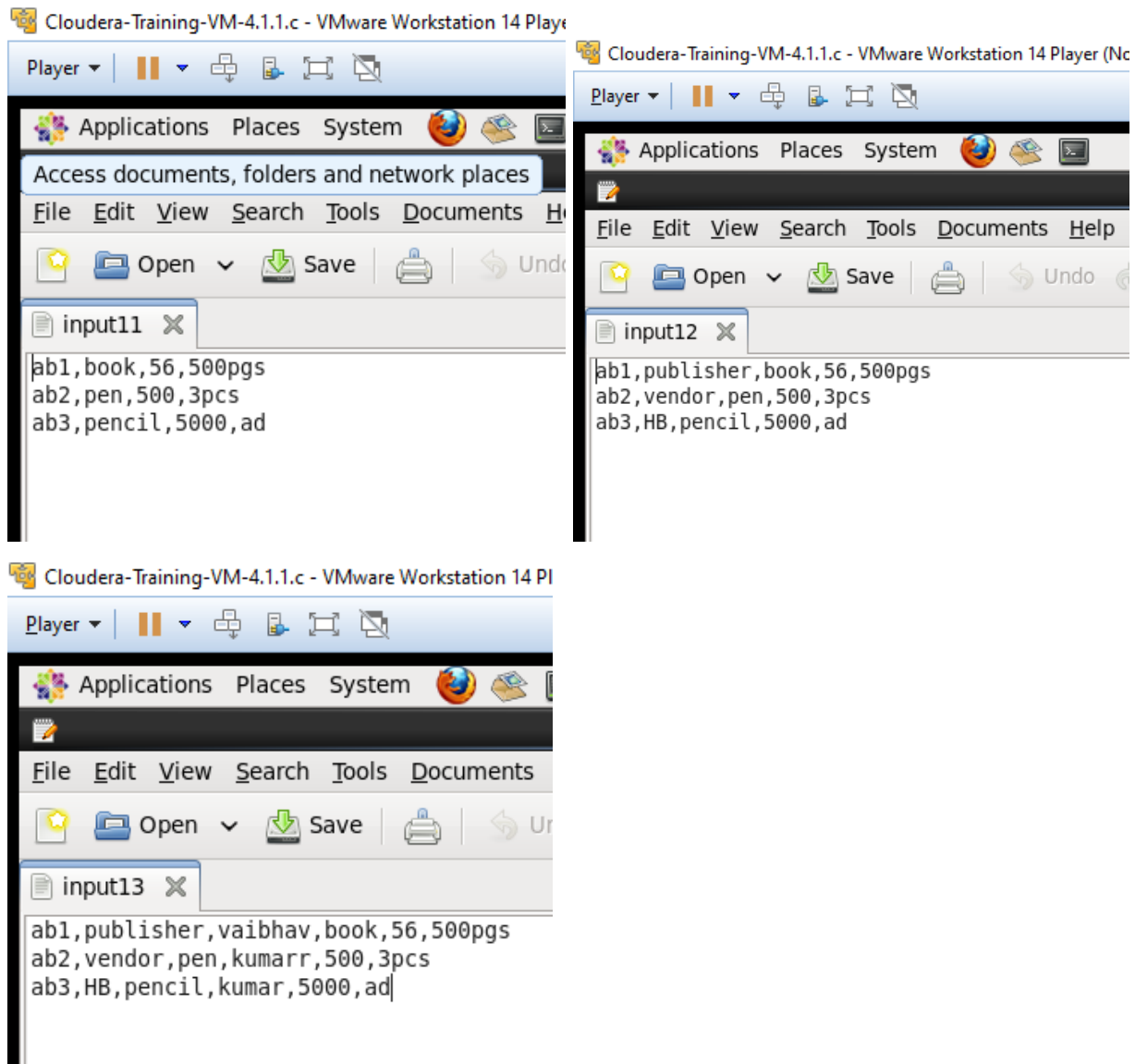


ASSIGNMENT

Multiple File Input

Problem Statement: Create 2 or 3 input files on your own , in which the data is present in different format. Write a program to process these files using different map class and perform any one aggregate function like sum, max, min etc. on it.



CODE:

```
import java.io.IOException;  
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.lib.input.MultipleInputs;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.util.GenericOptionsParser;
```

```
public class MultipleFileInp {
    public static class map1 extends Mapper <LongWritable, Text, Text,IntWritable>
    {
        @Override
        public void map(LongWritable key,Text value, Context con)
        throws IOException, InterruptedException{
            String line = value.toString();
            String[] line1=line.split(",");
            String category=line1[0];
            Text outputKey =new Text(category);
            int salary=Integer.parseInt(line1[2]);
            IntWritable outputValue=new IntWritable(salary);
            con.write(outputKey, outputValue);

        }
    }
    public static class map2 extends Mapper <LongWritable, Text, Text,IntWritable>
    {
        @Override
        public void map(LongWritable key,Text value, Context con)
        throws IOException, InterruptedException{
            String line = value.toString();
            String[] line1=line.split(",");
            String category=line1[0];
            Text outputKey =new Text(category);
            int salary=Integer.parseInt(line1[3]);
            IntWritable outputValue=new IntWritable(salary);
            con.write(outputKey, outputValue);

        }
    }
    public static class map3 extends Mapper <LongWritable, Text, Text,IntWritable>
```

```

{
    @Override
    public void map(LongWritable key,Text value, Context con)
    throws IOException, InterruptedException{
        String line = value.toString();
        String[] line1=line.split(",");
        String category=line1[0];
        Text outputKey =new Text(category);
        int salary=Integer.parseInt(line1[4]);
        IntWritable outputValue=new IntWritable(salary);
        con.write(outputKey, outputValue);

    }
}

public static class red extends Reducer<Text,IntWritable,Text,IntWritable>{
    public void reduce(Text category,Iterable<IntWritable> total_sal,Context con)throws
IOException,InterruptedException{
        int sum =0;
        for(IntWritable value:total_sal){
            sum+=value.get();
        }
        con.write(category, new IntWritable(sum));
    }
}
}

```

```

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

    Configuration c = new Configuration();
    String[]files = new GenericOptionsParser(c,args).getRemainingArgs();
    Path p1= new Path(files[0]);
    Path p2= new Path(files[1]);
    Path p3= new Path(files[2]);
    Path p4= new Path(files[3]);
    Job j =new Job(c,"multiple");
    j.setJarByClass(MultipleFileInp.class);
    j.setMapperClass(map1.class);
    j.setMapperClass(map2.class);
    j.setMapperClass(map3.class);
    j.setReducerClass(red.class);
    j.setMapOutputKeyClass(Text.class);
    j.setMapOutputValueClass(IntWritable.class);
    MultipleInputs.addInputPath(j, p1, TextInputFormat.class, map1.class);
    MultipleInputs.addInputPath(j, p2, TextInputFormat.class, map2.class);
}

```

```

    MultipleInputs.addInputPath(j, p3, TextInputFormat.class, map3.class);
    FileOutputFormat.setOutputPath(j, p4);
    System.exit(j.waitForCompletion(true)?0:1);
}
}

```

```

[training@localhost ~]$ hadoop jar /home/training/multifile.jar /user/training/vaibhav/input11 /user/training/vaibhav/input12 /user/training/vaibhav/input13 /user/training/multioutput5
22/08/29 11:22:54 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
22/08/29 11:22:54 INFO input.FileInputFormat: Total input paths to process : 1
22/08/29 11:22:54 WARN snappy.LoadSnappy: Snappy native library is available
22/08/29 11:22:54 INFO snappy.LoadSnappy: Snappy native library loaded
22/08/29 11:22:54 INFO input.FileInputFormat: Total input paths to process : 1
22/08/29 11:22:54 INFO input.FileInputFormat: Total input paths to process : 1
22/08/29 11:22:54 INFO mapred.JobClient: Running job: job_202208011504_0011
22/08/29 11:22:55 INFO mapred.JobClient: map 0% reduce 0%
22/08/29 11:22:59 INFO mapred.JobClient: map 66% reduce 0%
22/08/29 11:23:01 INFO mapred.JobClient: map 100% reduce 0%
22/08/29 11:23:03 INFO mapred.JobClient: map 100% reduce 100%
22/08/29 11:23:03 INFO mapred.JobClient: Job complete: job_202208011504_0011
22/08/29 11:23:03 INFO mapred.JobClient: Counters: 32
22/08/29 11:23:03 INFO mapred.JobClient: File System Counters
22/08/29 11:23:03 INFO mapred.JobClient: FILE: Number of bytes read=96
22/08/29 11:23:03 INFO mapred.JobClient: FILE: Number of bytes written=734280
22/08/29 11:23:03 INFO mapred.JobClient: FILE: Number of read operations=0
22/08/29 11:23:03 INFO mapred.JobClient: FILE: Number of large read operations=0
22/08/29 11:23:03 INFO mapred.JobClient: FILE: Number of write operations=0
22/08/29 11:23:03 INFO mapred.JobClient: HDFS: Number of bytes read=958
22/08/29 11:23:03 INFO mapred.JobClient: HDFS: Number of bytes written=27
22/08/29 11:23:03 INFO mapred.JobClient: HDFS: Number of read operations=6
22/08/29 11:23:03 INFO mapred.JobClient: HDFS: Number of large read operations=0
22/08/29 11:23:03 INFO mapred.JobClient: HDFS: Number of write operations=1
22/08/29 11:23:03 INFO mapred.JobClient: Job Counters
22/08/29 11:23:03 INFO mapred.JobClient: Launched map tasks=3
22/08/29 11:23:03 INFO mapred.JobClient: Launched reduce tasks=1
22/08/29 11:23:03 INFO mapred.JobClient: Data-local map tasks=3
22/08/29 11:23:03 INFO mapred.JobClient: Total time spent by all maps in occupied slots (ms)=8012
22/08/29 11:23:03 INFO mapred.JobClient: Total time spent by all reduces in occupied slots (ms)=3938
22/08/29 11:23:03 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0
22/08/29 11:23:03 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0
22/08/29 11:23:03 INFO mapred.JobClient: Map-Reduce Framework
22/08/29 11:23:03 INFO mapred.JobClient: Map input records=9
22/08/29 11:23:03 INFO mapred.JobClient: Map output records=9
22/08/29 11:23:03 INFO mapred.JobClient: Map output bytes=72
22/08/29 11:23:03 INFO mapred.JobClient: Input split bytes=732

```

File: [/user/training/multioutput5/part-r-00000](#)

Goto :

[Go back to dir listing](#)

[Advanced view/download options](#)

ab1	168
ab2	1500
ab3	15000

MAX

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
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.MultipleInputs;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class MultipleFileInp {
    public static class map1 extends Mapper<LongWritable,Text,Text,IntWritable>
    {
        public void map(LongWritable key, Text value, Context con) throws IOException,
        InterruptedException
        {
            String line = value.toString();
            String[] line1 = line.split(",");
            String name = line1[0];
            Text outputKey = new Text(name);
            int cost = Integer.parseInt(line1[2]);
            IntWritable outputValue = new IntWritable(cost);
            con.write(outputKey, outputValue);
        } }
    public static class Map2 extends Mapper<LongWritable, Text, Text, IntWritable>
    {
        public void map(LongWritable key, Text value, Context con) throws IOException,
        InterruptedException
        {
            String line = value.toString();
            String[] line1 = line.split(",");
            String name = line1[0];
            Text outputKey = new Text(name);
            int cost = Integer.parseInt(line1[3]);
            IntWritable outputValue = new IntWritable(cost);
            con.write(outputKey, outputValue);
        } }
    public static class Map3 extends Mapper<LongWritable, Text, Text, IntWritable>
```

```
{

    public void map(LongWritable key, Text value, Context con) throws IOException,
    InterruptedException
    {
        String line = value.toString();
        String[] line1 = line.split(",");
        String name = line1[0];
        Text outputKey = new Text(name);
        int cost = Integer.parseInt(line1[4]);
        IntWritable outputValue = new IntWritable(cost);
        con.write(outputKey, outputValue);
    }
}

public static class Red extends Reducer<Text, IntWritable, Text,IntWritable>
{
    public void reduce(Text name, Iterable<IntWritable>total_cost, Context con) throws IOException,
    InterruptedException
    {
        int sum = 0;
        int max = 0;
        for (IntWritable value : total_cost) {
            if (value.get() > max) {
                max = value.get();
            }
        }
        con.write(name,new IntWritable(sum));
    }
}

public static void main(String[] args) throws Exception{
    Configuration c = new Configuration();
    GenericOptionsParser parser = new GenericOptionsParser(c,args);
    String[] files = parser.getRemainingArgs();
    Path p1 = new Path(files[0]);
    Path p2 = new Path(files[1]);
    Path p3 = new Path(files[2]);
    Path p4 = new Path(files[3]);
    Job j = new Job(c,"multiple");
    j.setJarByClass(MultipleFileInp .class);
    j.setMapperClass(map1.class);
    j.setMapperClass(Map2.class);
    j.setMapperClass(Map3.class);
    j.setReducerClass(Red.class);
    j.setOutputKeyClass(Text.class);
```

```
j.setOutputValueClass(IntWritable.class);
MultipleInputs.addInputPath(j,p1,TextInputFormat.class,map1.class);
MultipleInputs.addInputPath(j, p2, TextInputFormat.class,Map2.class);
MultipleInputs.addInputPath(j, p3, TextInputFormat.class,Map3.class);
FileOutputFormat.setOutputPath(j,p4);
System.exit(j.waitForCompletion(true)?0:1);}}
```

```
[training@localhost ~]$ hadoop jar /home/training/multifilemax.jar /user/training/vaibhav/input11 /user/training/vaibhav/input12 /user/training/vaibhav/input13 /user/training/multimax
22/08/29 11:42:39 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
22/08/29 11:42:39 INFO input.FileInputFormat: Total input paths to process : 1
22/08/29 11:42:39 WARN snappy.LoadSnappy: Snappy native library is available
22/08/29 11:42:39 INFO snappy.LoadSnappy: Snappy native library loaded
22/08/29 11:42:39 INFO input.FileInputFormat: Total input paths to process : 1
22/08/29 11:42:39 INFO input.FileInputFormat: Total input paths to process : 1
22/08/29 11:42:40 INFO mapred.JobClient: Running job: job_202208011504_0012
22/08/29 11:42:40 INFO mapred.JobClient: map 0% reduce 0%
22/08/29 11:42:44 INFO mapred.JobClient: map 66% reduce 0%
22/08/29 11:42:46 INFO mapred.JobClient: map 100% reduce 0%
22/08/29 11:42:48 INFO mapred.JobClient: map 100% reduce 100%
22/08/29 11:42:48 INFO mapred.JobClient: Job complete: job_202208011504_0012
22/08/29 11:42:48 INFO mapred.JobClient: Counters: 32
22/08/29 11:42:48 INFO mapred.JobClient: File System Counters
22/08/29 11:42:48 INFO mapred.JobClient: FILE: Number of bytes read=96
22/08/29 11:42:48 INFO mapred.JobClient: FILE: Number of bytes written=734240
22/08/29 11:42:48 INFO mapred.JobClient: FILE: Number of read operations=0
22/08/29 11:42:48 INFO mapred.JobClient: FILE: Number of large read operations=0
22/08/29 11:42:48 INFO mapred.JobClient: FILE: Number of write operations=0
22/08/29 11:42:48 INFO mapred.JobClient: HDFS: Number of bytes read=958
22/08/29 11:42:48 INFO mapred.JobClient: HDFS: Number of bytes written=18
22/08/29 11:42:48 INFO mapred.JobClient: HDFS: Number of read operations=6
22/08/29 11:42:48 INFO mapred.JobClient: HDFS: Number of large read operations=0
22/08/29 11:42:48 INFO mapred.JobClient: HDFS: Number of write operations=1
22/08/29 11:42:48 INFO mapred.JobClient: Job Counters
22/08/29 11:42:48 INFO mapred.JobClient: Launched map tasks=3
22/08/29 11:42:48 INFO mapred.JobClient: Launched reduce tasks=1
22/08/29 11:42:48 INFO mapred.JobClient: Data-local map tasks=3
22/08/29 11:42:48 INFO mapred.JobClient: Total time spent by all maps in occupied slots (ms)=7836
```

[Go back to dir listing](#)

[Advanced view/download options](#)

Ab1	570
Ab1	1670
ab2	600
ab2	1500
ab3	5500
ab3	16100