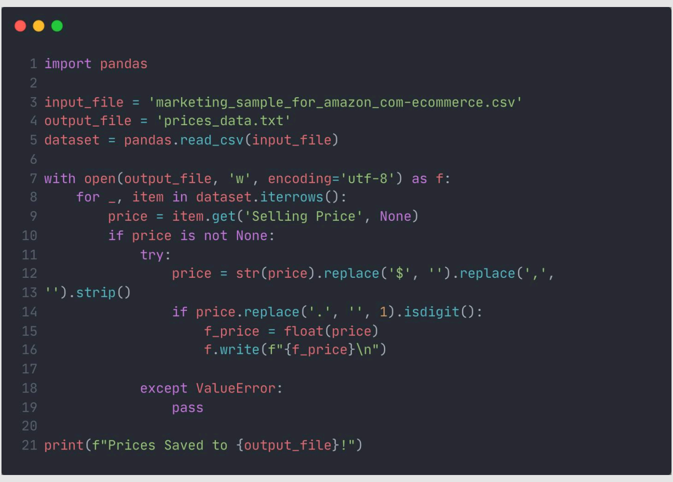
(CNET444) Distributed Systems Lab 4 Experiment	
Dr. Hussein Zangoti	**Hassan Ali Sharahili - 202201918**

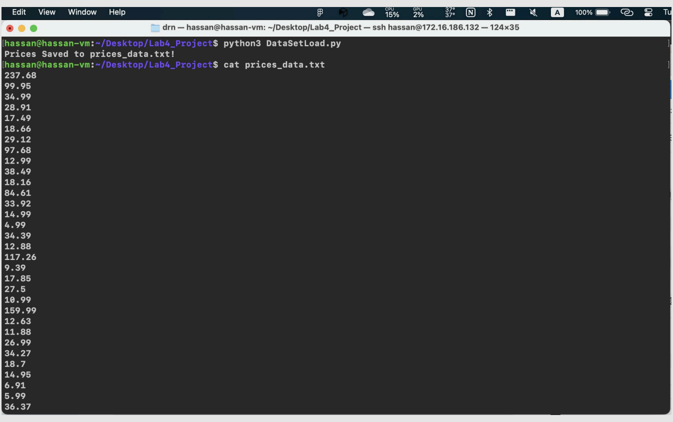
Lab Tasks:

Task 1 & 2: Download & Clean the dataset of amazon product prices + Show it

Clean the dataset by using this code (Extract Prices & Clean the non-number fields):



```python
import pandas

input_file = 'marketing_sample_for_amazon_com-ecommerce.csv'
output_file = 'prices_data.txt'
dataset = pandas.read_csv(input_file)

with open(output_file, 'w', encoding='utf-8') as f:
    for _, item in dataset.iterrows():
        price = item.get('Selling Price', None)
        if price is not None:
            try:
                price = str(price).replace('$', '').replace(',',
'').strip()
                if price.replace('.', '', 1).isdigit():
                    f_price = float(price)
                    f.write(f"{f_price}\n")

            except ValueError:
                pass

print(f"Prices Saved to {output_file}!")
```

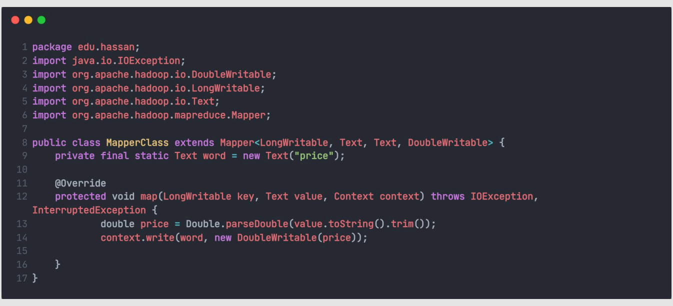


Edit
View
Window
Help
drn — hassan@hassan-vm: ~/Desktop/Lab4_Project — ssh hassan@172.16.186.132 — 124×35

```
hassan@hassan-vm:~/Desktop/Lab4_Project$ python3 DataSetLoad.py
Prices Saved to prices_data.txt!
hassan@hassan-vm:~/Desktop/Lab4_Project$ cat prices_data.txt
237.68
99.95
34.99
28.91
17.49
18.66
29.12
97.68
12.99
38.49
18.16
84.61
33.92
14.99
4.99
34.39
12.88
117.26
9.39
17.85
27.5
10.99
159.99
12.63
11.88
26.99
34.27
18.7
14.95
6.91
5.99
36.37
```

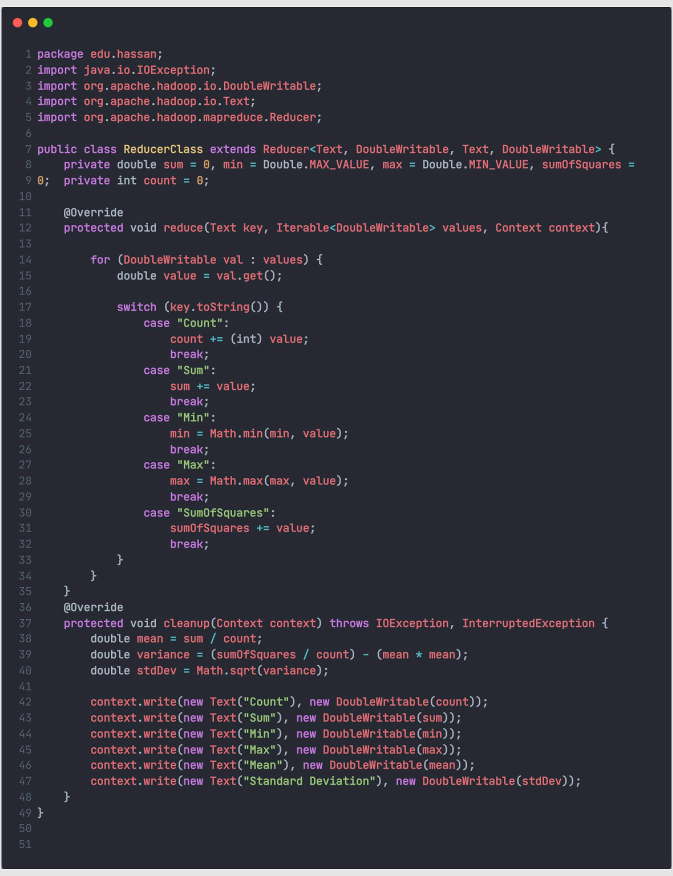
Task 3: Writing MapReduce Program (Mapper, Reducer, Driver, Combiner)

1. **Mapper Class:**



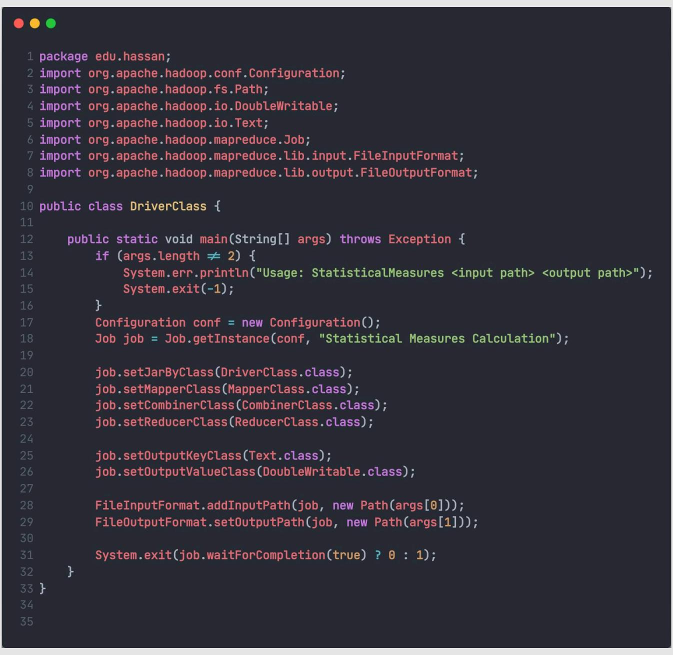
```java
package edu.hassan;
import java.io.IOException;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class MapperClass extends Mapper<LongWritable, Text, Text, DoubleWritable> {
    private final static Text word = new Text("price");

    @Override
    protected void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException {
            double price = Double.parseDouble(value.toString().trim());
            context.write(word, new DoubleWritable(price));

    }
}
```

2. **Reducer Class:**



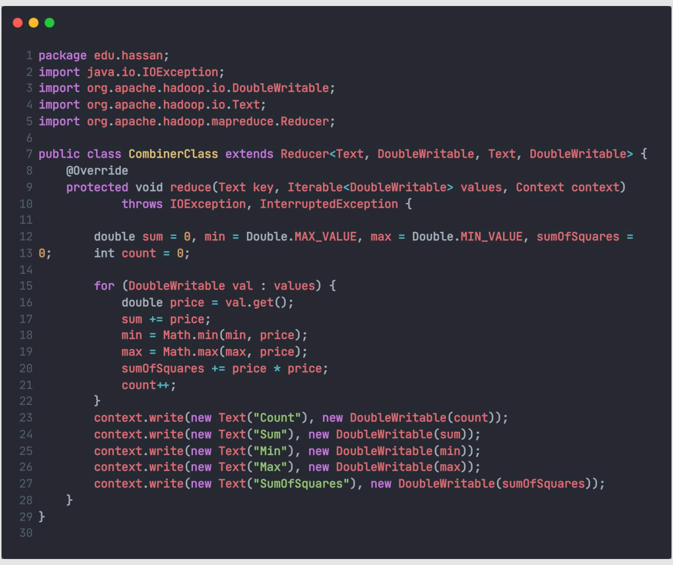
```java
package edu.hassan;
import java.io.IOException;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class ReducerClass extends Reducer<Text, DoubleWritable, Text, DoubleWritable> {
    private double sum = 0, min = Double.MAX_VALUE, max = Double.MIN_VALUE, sumOfSquares =
0;  private int count = 0;

    @Override
    protected void reduce(Text key, Iterable<DoubleWritable> values, Context context){

        for (DoubleWritable val : values) {
            double value = val.get();

            switch (key.toString()) {
                case "Count":
                    count += (int) value;
                    break;
                case "Sum":
                    sum += value;
                    break;
                case "Min":
                    min = Math.min(min, value);
                    break;
                case "Max":
                    max = Math.max(max, value);
                    break;
                case "SumOfSquares":
                    sumOfSquares += value;
                    break;
            }
        }
    }
    @Override
    protected void cleanup(Context context) throws IOException, InterruptedException {
        double mean = sum / count;
        double variance = (sumOfSquares / count) - (mean * mean);
        double stdDev = Math.sqrt(variance);

        context.write(new Text("Count"), new DoubleWritable(count));
        context.write(new Text("Sum"), new DoubleWritable(sum));
        context.write(new Text("Min"), new DoubleWritable(min));
        context.write(new Text("Max"), new DoubleWritable(max));
        context.write(new Text("Mean"), new DoubleWritable(mean));
        context.write(new Text("Standard Deviation"), new DoubleWritable(stdDev));
    }
}
```

3. **Driver Class:**



```java
package edu.hassan;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
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.output.FileOutputFormat;

public class DriverClass {

    public static void main(String[] args) throws Exception {
        if (args.length ≠ 2) {
            System.err.println("Usage: StatisticalMeasures <input path> <output path>");
            System.exit(-1);
        }
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "Statistical Measures Calculation");

        job.setJarByClass(DriverClass.class);
        job.setMapperClass(MapperClass.class);
        job.setCombinerClass(CombinerClass.class);
        job.setReducerClass(ReducerClass.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(DoubleWritable.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

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

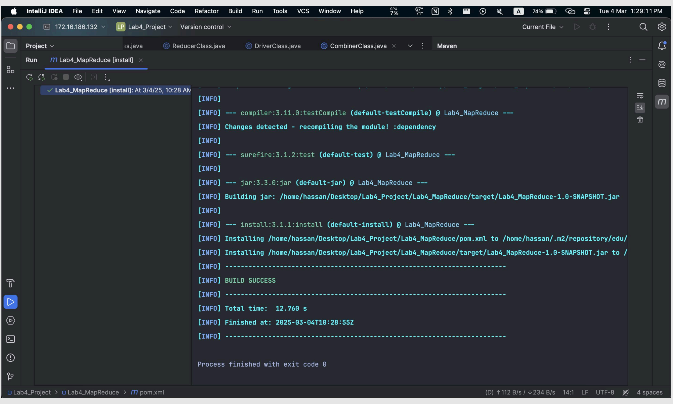
4. Combiner Class (Optional):



```java
package edu.hassan;
import java.io.IOException;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class CombinerClass extends Reducer<Text, DoubleWritable, Text, DoubleWritable> {
    @Override
    protected void reduce(Text key, Iterable<DoubleWritable> values, Context context)
            throws IOException, InterruptedException {

        double sum = 0, min = Double.MAX_VALUE, max = Double.MIN_VALUE, sumOfSquares =
0;      int count = 0;

        for (DoubleWritable val : values) {
            double price = val.get();
            sum += price;
            min = Math.min(min, price);
            max = Math.max(max, price);
            sumOfSquares += price * price;
            count++;
        }
        context.write(new Text("Count"), new DoubleWritable(count));
        context.write(new Text("Sum"), new DoubleWritable(sum));
        context.write(new Text("Min"), new DoubleWritable(min));
        context.write(new Text("Max"), new DoubleWritable(max));
        context.write(new Text("SumOfSquares"), new DoubleWritable(sumOfSquares));
    }
}
```

Task 4: Compile & Build JAR File

→ mvn clean
→ mvn install

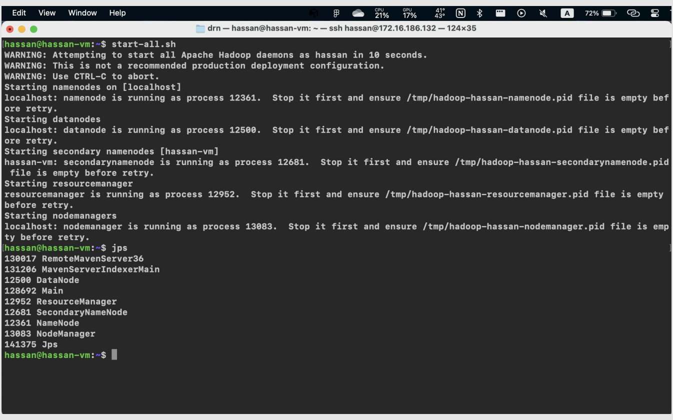


Lab4_MapReduce [install]: At 3/4/25, 10:28 AM

```
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ Lab4_MapReduce ---
[INFO] Changes detected - recompiling the module! :dependency
[INFO]
[INFO] --- surefire:3.1.2:test (default-test) @ Lab4_MapReduce ---
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ Lab4_MapReduce ---
[INFO] Building jar: /home/hassan/Desktop/Lab4_Project/Lab4_MapReduce/target/Lab4_MapReduce-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- install:3.1.1:install (default-install) @ Lab4_MapReduce ---
[INFO] Installing /home/hassan/Desktop/Lab4_Project/Lab4_MapReduce/pom.xml to /home/hassan/.m2/repository/edu/
[INFO] Installing /home/hassan/Desktop/Lab4_Project/Lab4_MapReduce/target/Lab4_MapReduce-1.0-SNAPSHOT.jar to /
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  12.760 s
[INFO] Finished at: 2025-03-04T10:28:55Z
[INFO] ------------------------------------------------------------------------

Process finished with exit code 0
```

Task 5: Preparing Hadoop + Create Directory & Upload the Input file

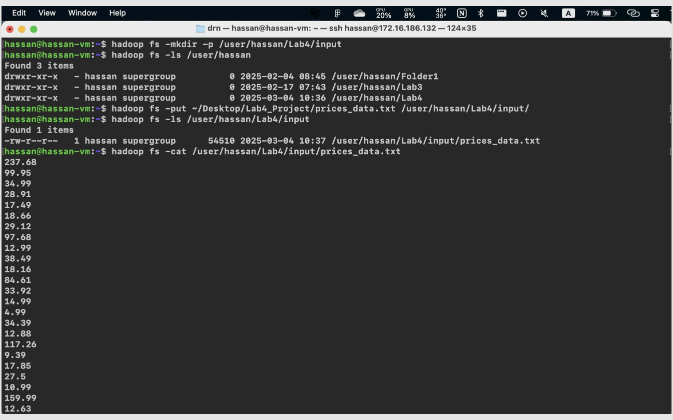
$ start-all.sh
$ jps



Edit View Window Help
drn — hassan@hassan-vm: ~ — ssh hassan@172.16.186.132 — 124×35

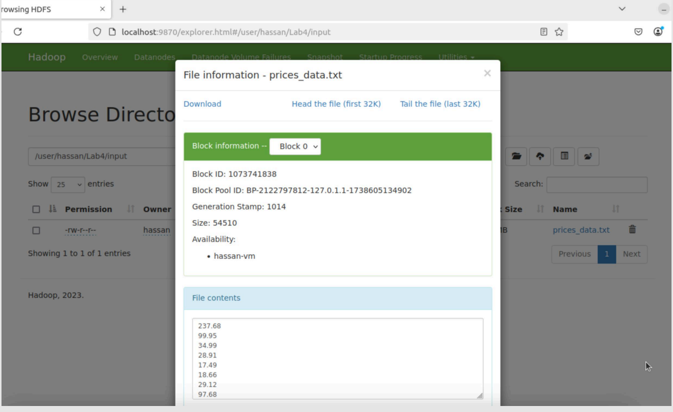
```
hassan@hassan-vm:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hassan in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
localhost: namenode is running as process 12361.  Stop it first and ensure /tmp/hadoop-hassan-namenode.pid file is empty bef
ore retry.
Starting datanodes
localhost: datanode is running as process 12500.  Stop it first and ensure /tmp/hadoop-hassan-datanode.pid file is empty bef
ore retry.
Starting secondary namenodes [hassan-vm]
hassan-vm: secondarynamenode is running as process 12681.  Stop it first and ensure /tmp/hadoop-hassan-secondarynamenode.pid
 file is empty before retry.
Starting resourcemanager
resourcemanager is running as process 12952.  Stop it first and ensure /tmp/hadoop-hassan-resourcemanager.pid file is empty
before retry.
Starting nodemanagers
localhost: nodemanager is running as process 13083.  Stop it first and ensure /tmp/hadoop-hassan-nodemanager.pid file is emp
ty before retry.
hassan@hassan-vm:~$ jps
130017 RemoteMavenServer36
131206 MavenServerIndexerMain
12500 DataNode
128692 Main
12952 ResourceManager
12681 SecondaryNameNode
12361 NameNode
13083 NodeManager
141375 Jps
hassan@hassan-vm:~$
```

$ hadoop fs -mkdir -p /user/hassan/Lab4/input
$ hadoop fs -ls /user/hassan
$ hadoop fs -put ~/Desktop/Lab4_Project/prices_data.txt /user/hassan/Lab4/input/
$ hadoop fs -ls /user/hassan/Lab4/input
$ hadoop fs -cat /user/hassan/Lab4/input/prices_data.txt



Edit View Window Help
drn — hassan@hassan-vm: ~ — ssh hassan@172.16.186.132 — 124×35

```
hassan@hassan-vm:~$ hadoop fs -mkdir -p /user/hassan/Lab4/input
hassan@hassan-vm:~$ hadoop fs -ls /user/hassan
Found 3 items
drwxr-xr-x   - hassan supergroup          0 2025-02-04 08:45 /user/hassan/Folder1
drwxr-xr-x   - hassan supergroup          0 2025-02-17 07:43 /user/hassan/Lab3
drwxr-xr-x   - hassan supergroup          0 2025-03-04 10:36 /user/hassan/Lab4
hassan@hassan-vm:~$ hadoop fs -put ~/Desktop/Lab4_Project/prices_data.txt /user/hassan/Lab4/input/
hassan@hassan-vm:~$ hadoop fs -ls /user/hassan/Lab4/input
Found 1 items
-rw-r--r--   1 hassan supergroup      54510 2025-03-04 10:37 /user/hassan/Lab4/input/prices_data.txt
hassan@hassan-vm:~$ hadoop fs -cat /user/hassan/Lab4/input/prices_data.txt
237.68
99.95
34.99
28.91
17.49
18.66
29.12
97.68
12.99
38.49
18.16
84.61
33.92
14.99
4.99
34.39
12.88
117.26
9.39
17.85
27.5
10.99
159.99
12.63
```



File information - prices_data.txt

Permission	Owner	Size	Name
-rw-r--r--	hassan	B	prices_data.txt

Showing 1 to 1 of 1 entries

Block ID: 1073741838
Block Pool ID: BP-2122797812-127.0.1.1-1738605134902
Generation Stamp: 1014
Size: 54510
Availability:
- hassan-vm

File contents

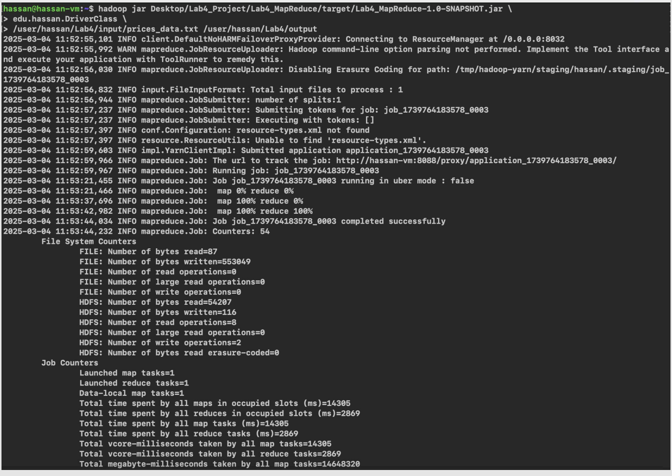
```
237.68
99.95
34.99
28.91
17.49
18.66
29.12
97.68
```

Task 6: Execute the MapReduce Program

$ hadoop jar Desktop/Lab4_Project/Lab4_MapReduce/target/Lab4_MapReduce-1.0-SNAPSHOT.jar \
> edu.hassan.DriverClass \
> /user/hassan/Lab4/input/prices_data.txt /user/hassan/Lab4/output

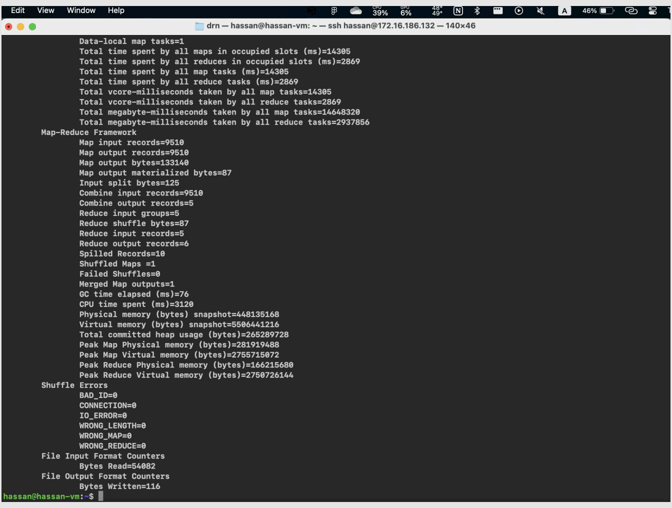
(1)

Edit View Window Help
drn — hassan@hassan-vm: ~ — ssh hassan@172.16.186.132 — 140×46



```
hassan@hassan-vm:~$ hadoop jar Desktop/Lab4_Project/Lab4_MapReduce/target/Lab4_MapReduce-1.0-SNAPSHOT.jar \
> edu.hassan.DriverClass \
> /user/hassan/Lab4/input/prices_data.txt /user/hassan/Lab4/output
2025-03-04 11:52:55,101 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2025-03-04 11:52:55,992 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface a
nd execute your application with ToolRunner to remedy this.
2025-03-04 11:52:56,030 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hassan/.staging/job_
1739764183578_0003
2025-03-04 11:52:56,832 INFO input.FileInputFormat: Total input files to process : 1
2025-03-04 11:52:56,944 INFO mapreduce.JobSubmitter: number of splits:1
2025-03-04 11:52:57,237 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1739764183578_0003
2025-03-04 11:52:57,237 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-03-04 11:52:57,397 INFO conf.Configuration: resource-types.xml not found
2025-03-04 11:52:57,397 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-03-04 11:52:59,603 INFO impl.YarnClientImpl: Submitted application application_1739764183578_0003
2025-03-04 11:52:59,966 INFO mapreduce.Job: The url to track the job: http://hassan-vm:8088/proxy/application_1739764183578_0003/
2025-03-04 11:52:59,967 INFO mapreduce.Job: Running job: job_1739764183578_0003
2025-03-04 11:53:21,455 INFO mapreduce.Job: Job job_1739764183578_0003 running in uber mode : false
2025-03-04 11:53:21,466 INFO mapreduce.Job:  map 0% reduce 0%
2025-03-04 11:53:37,696 INFO mapreduce.Job:  map 100% reduce 0%
2025-03-04 11:53:42,982 INFO mapreduce.Job:  map 100% reduce 100%
2025-03-04 11:53:44,034 INFO mapreduce.Job: Job job_1739764183578_0003 completed successfully
2025-03-04 11:53:44,232 INFO mapreduce.Job: Counters: 54
	File System Counters
		FILE: Number of bytes read=87
		FILE: Number of bytes written=553049
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=54207
		HDFS: Number of bytes written=116
		HDFS: Number of read operations=8
		HDFS: Number of large read operations=0
		HDFS: Number of write operations=2
		HDFS: Number of bytes read erasure-coded=0
	Job Counters
		Launched map tasks=1
		Launched reduce tasks=1
		Data-local map tasks=1
		Total time spent by all maps in occupied slots (ms)=14305
		Total time spent by all reduces in occupied slots (ms)=2869
		Total time spent by all map tasks (ms)=14305
		Total time spent by all reduce tasks (ms)=2869
		Total vcore-milliseconds taken by all map tasks=14305
		Total vcore-milliseconds taken by all reduce tasks=2869
		Total megabyte-milliseconds taken by all map tasks=14648320
```

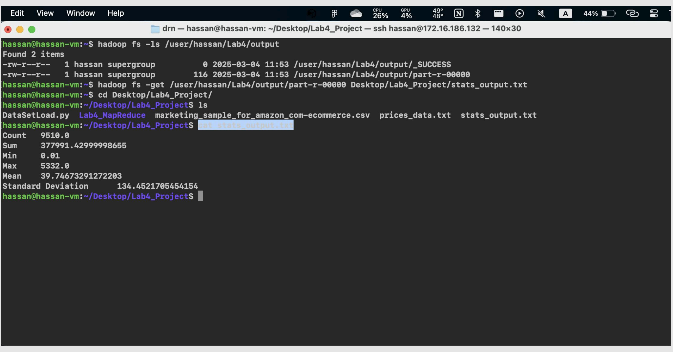
(2)


Edit
View
Window
Help
drn — hassan@hassan-vm: ~ — ssh hassan@172.16.186.132 — 140×46

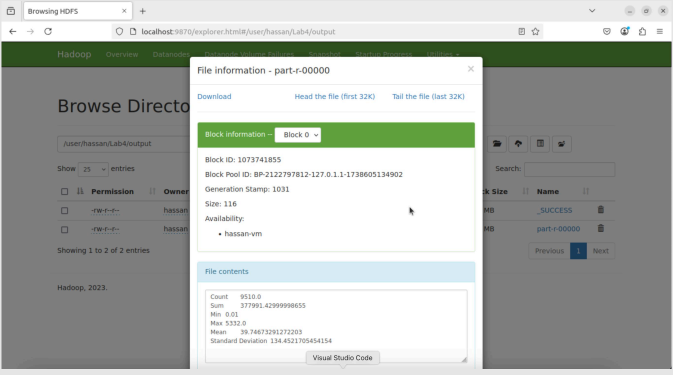
```
		Data-local map tasks=1
		Total time spent by all maps in occupied slots (ms)=14305
		Total time spent by all reduces in occupied slots (ms)=2869
		Total time spent by all map tasks (ms)=14305
		Total time spent by all reduce tasks (ms)=2869
		Total vcore-milliseconds taken by all map tasks=14305
		Total vcore-milliseconds taken by all reduce tasks=2869
		Total megabyte-milliseconds taken by all map tasks=14648320
		Total megabyte-milliseconds taken by all reduce tasks=2937856
	Map-Reduce Framework
		Map input records=9510
		Map output records=9510
		Map output bytes=133140
		Map output materialized bytes=87
		Input split bytes=125
		Combine input records=9510
		Combine output records=5
		Reduce input groups=5
		Reduce shuffle bytes=87
		Reduce input records=5
		Reduce output records=6
		Spilled Records=10
		Shuffled Maps =1
		Failed Shuffles=0
		Merged Map outputs=1
		GC time elapsed (ms)=76
		CPU time spent (ms)=3120
		Physical memory (bytes) snapshot=448135168
		Virtual memory (bytes) snapshot=5506441216
		Total committed heap usage (bytes)=265289728
		Peak Map Physical memory (bytes)=281919488
		Peak Map Virtual memory (bytes)=2755715072
		Peak Reduce Physical memory (bytes)=166215680
		Peak Reduce Virtual memory (bytes)=2750726144
	Shuffle Errors
		BAD_ID=0
		CONNECTION=0
		IO_ERROR=0
		WRONG_LENGTH=0
		WRONG_MAP=0
		WRONG_REDUCE=0
	File Input Format Counters
		Bytes Read=54082
	File Output Format Counters
		Bytes Written=116
hassan@hassan-vm:~$ 
```

Task 7: Show the output

```
$ hadoop fs -ls /user/hassan/Lab4/output
$ hadoop fs -get /user/hassan/Lab4/output/part-r-00000 Desktop/Lab4_Project/stats_output.txt
$ cat stats_output.txt
```


Edit
View
Window
Help
drn — hassan@hassan-vm: ~/Desktop/Lab4_Project — ssh hassan@172.16.186.132 — 140×30

```
hassan@hassan-vm:~$ hadoop fs -ls /user/hassan/Lab4/output
Found 2 items
-rw-r--r--   1 hassan supergroup          0 2025-03-04 11:53 /user/hassan/Lab4/output/_SUCCESS
-rw-r--r--   1 hassan supergroup        116 2025-03-04 11:53 /user/hassan/Lab4/output/part-r-00000
hassan@hassan-vm:~$ hadoop fs -get /user/hassan/Lab4/output/part-r-00000 Desktop/Lab4_Project/stats_output.txt
hassan@hassan-vm:~$ cd Desktop/Lab4_Project/
hassan@hassan-vm:~/Desktop/Lab4_Project$ ls
DataSetLoad.py  Lab4_MapReduce  marketing_sample_for_amazon_com-ecommerce.csv  prices_data.txt  stats_output.txt
hassan@hassan-vm:~/Desktop/Lab4_Project$ cat stats_output.txt
Count	9510.0
Sum	377991.42999998655
Min	0.01
Max	5332.0
Mean	39.74673291272203
Standard Deviation	134.4521705454154
hassan@hassan-vm:~/Desktop/Lab4_Project$ 
```



File information - part-r-00000

Block ID: 1073741855

Block Pool ID: BP-2122797812-127.0.1.1-1738605134902

Generation Stamp: 1031

Size: 116

Availability:

- hassan-vm

Permission	Owner	k Size	Name
-rw-r--r--	hassan	MB	_SUCCESS
-rw-r--r--	hassan	MB	part-r-00000

```
Count	9510.0
Sum	377991.42999998655
Min	0.01
Max	5332.0
Mean	39.74673291272203
Standard Deviation	134.4521705454154
```

Explanation of the Implementation

- This lab discusses about calculating statistical measures (Count, Sum, Min, Max, Mean, and Standard Deviation) from a price dataset of amazon products using the MapReduce mechanism and the Hadoop Distributed File System.
- First, I created a Java program that consists of four classes: one for Mapping, one for Combining, one for Reducing, and one for Running the job of statistical calculation. Each class implements a specific function for the MapReduce mechanism:
  - Mapper Class: Reads the input file and emits each price with a fixed key ("price").
  - Combiner Class: Performs local aggregation to reduce the amount of data shuffled to the Reducer.
  - Reducer Class: Aggregates the final results and calculates all the required statistical measures.
  - Driver Class: Configures and manages the MapReduce job execution.
- After developing the program, I built it into a JAR file and created an input file (prices_data.txt) containing the cleaned price data.
- The input file was uploaded to the Hadoop File System in a new directory "/Lab4/input" inside an existing directory created previously.
- Next, I executed the JAR program with the input file uploaded to HDFS. Hadoop distributed the statistical calculation function and generated an output file containing the calculated Count, Sum, Minimum, Maximum, Mean, and Standard Deviation of the prices.

- The System & Tools I'm using:
  - Macbook Air M1
  - VMware Fusion "Virtual Machine Hypervisor"
  - Ubuntu for Servers 22.04 "With ubuntu-desktop package"
  - Hadoop version 3.3.6
  - JDK version 11
  - IntelliJ IDEA with SSH feature (for remote programming)
  - GitHub (to upload the files on a remote repository)

Lab Files Repository on GitHub:

- https://github.com/drn-i/DS_Lab4