Large Scale Data Management

ASSIGNMENT 1

Stylianos Giagkos | f3352410

PART I

1. In this part you will familiarize yourself with building and submitting map-reduce applications for execution. Using the Vagrantfile provided with this project, you will initialize a virtual machine that offers you access to Java, Maven, and Hadoop. You will use Java and Maven to build your application and Hadoop to execute it. To initialize your virtual machine, you must have vagrant and virtualbox (or some other virtualization software) installed. If you do, you can initialize your virtual machine by executing the following command in your terminal:

To successfully complete the first part of this project you have to provide the output of the above command for an input file of your choice (other than MobyDick.txt). You should include in your report details about the file you used (URL where it can be found) and all the execution logs of the application as they were printed in your terminal. Keep in mind that you cannot use the same output location for multiple executions, so you should delete any temporary results.

Modifications to the Driver File and Vagrant File

To complete the first part of the project, it was necessary to make adjustments to both the provided Java driver file and the Vagrant configuration file to ensure proper execution.

Vagrant File

The Vagrant file was modified to facilitate the download of *The Brothers Karamazov* from the Gutenberg repository. Specifically, I adjusted the parameters responsible for retrieving and storing the text file, ensuring that the dataset was correctly fetched and accessible for processing.

```
  config.vm.provision "shell", inline: <<-SHELL
    apt-get update
    apt-get install -y wget docker-compose openjdk-8-jdk maven p7zip-full
    update-alternatives --set java /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
    cd /vagrant/docker-hadoop
    docker-compose up -d
    cd /vagrant/hadoop-mapreduce-examples/
    mvn clean install
    docker cp target/hadoop-map-reduce-examples-1.0-SNAPSHOT.jar namenode:/
    cd /vagrant
    wget https://www.gutenberg.org/cache/epub/28054/pg28054.txt -O TheBrothersKaramazov.txt
    docker cp TheBrothersKaramazov.txt namenode:/
    docker exec namenode hdfs dfs -mkdir -p /user/hdfs/input
    docker exec namenode hdfs dfs -put TheBrothersKaramazov.txt /user/hdfs/input/
    docker exec namenode rm TheBrothersKaramazov.txt
    rm TheBrothersKaramazov.txt
  SHELL
end
```

Driver.java

In the Driver.java file, input file directory was updated to correctly point to The Brothers Karamazov text file. This modification allowed the WordCount application to process the desired text and generate meaningful word frequency results.

These changes ensured seamless execution of the WordCount program with the specified dataset, aligning with the project requirements.

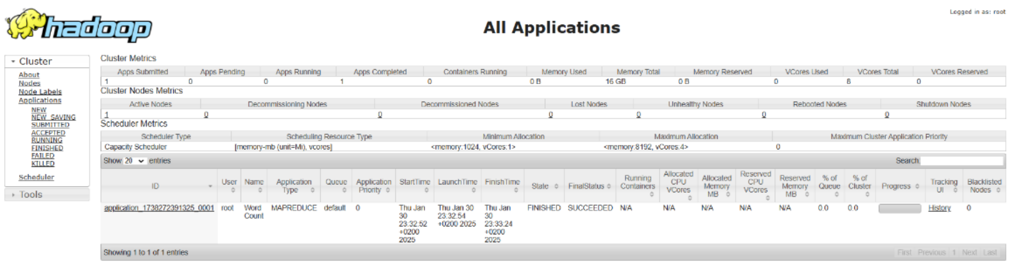
```
// set io paths
FileInputFormat.addInputPath(job, new Path("/user/hdfs/input/TheBrothersKaramazov.txt"));
FileOutputFormat.setOutputPath(job, new Path("/user/hdfs/output/"));
```

Results

The job was successfully executed, as confirmed by the Hadoop file system logs. Below, you can find the terminal logs that document the execution process, along with a sample of the generated output.

These results demonstrate that the WordCount application processed The Brothers Karamazov text file correctly, producing the expected word frequency counts.

HDFS Interface



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
1	0	0	1	0	0 B	16 GB	0 B	0	8	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
1	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1738272391325_0001	root	Word Count	MAPREDUCE	default	0	Thu Jan 30 23:32:52 +0200 2025	Thu Jan 30 23:32:54 +0200 2025	Thu Jan 30 23:33:24 +0200 2025	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History	0

Showing 1 to 1 of 1 entries

Log files

```
	File System Counters
		FILE: Number of bytes read=140308
		FILE: Number of bytes written=738909
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=2042118
		HDFS: Number of bytes written=342397
		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
		Rack-local map tasks=1
		Total time spent by all maps in occupied slots (ms)=24776
		Total time spent by all reduces in occupied slots (ms)=36152
		Total time spent by all map tasks (ms)=6194
		Total time spent by all reduce tasks (ms)=4519
		Total vcore-milliseconds taken by all map tasks=6194
		Total vcore-milliseconds taken by all reduce tasks=4519
		Total megabyte-milliseconds taken by all map tasks=25370624
		Total megabyte-milliseconds taken by all reduce tasks=37019648
	Map-Reduce Framework
		Map input records=37635
		Map output records=360406
		Map output bytes=3445554
		Map output materialized bytes=140300
		Input split bytes=115
		Combine input records=360406
		Combine output records=29994
		Reduce input groups=29994
		Reduce shuffle bytes=140300
		Reduce input records=29994
		Reduce output records=29994
		Spilled Records=59988
		Shuffled Maps =1
		Failed Shuffles=0
		Merged Map outputs=1
		Merged Map outputs=1
		GC time elapsed (ms)=174
		CPU time spent (ms)=2970
		Physical memory (bytes) snapshot=347836416
		Virtual memory (bytes) snapshot=13150445568
		Total committed heap usage (bytes)=230821888
		Peak Map Physical memory (bytes)=226885632
		Peak Map Virtual memory (bytes)=4955017216
		Peak Reduce Physical memory (bytes)=120950784
		Peak Reduce Virtual memory (bytes)=8195428352
	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=2042003
	File Output Format Counters
		Bytes Written=342397
```

Map Reduce Output

```
        6525
#28054] 1
$5,000) 1
($1     1
(801)   1
(Ah!    1
(Alyosha        4
(Alyosha,       1
(And    2
(As     1
(At     1
(Do     1
(Fenya  1
(Fyodor 1
(Grigory,       1
(He     5
(Here   1
(I      14
(Ippolit        1
(It     2
(It’s   1
(Laughter       1
(Madame 1
```

PART II

For the second part of this project you will develop your own map-reduce application. You will first download and go through your input file, with records of car sales: https://auebgr-my. sharepoint.com/:x:/g/personal/panagiotisliakos_aueb_gr/EXHUKfvONytBvZrjA3HbzzYBjihX_ K4LQNhKMB1O-c3qHA?e=fKpjbK This is a .csv file and you are going to use only some of its columns in your project. The .csv has a header (first line) which will help you identify the information you need. Your job is to produce a result file that provides for every seller and month pair (for example: “kia motors america inc:2024-12”) the car with the largest (sellingprice - mmr) difference, along with this “difference” and the average difference of all cars for the same seller and month pair (for example: “Kia Sorento LX: 1000, avg: 250.46).

Car sales Java Codes

As part of the second phase of the project, I developed three key Java files to implement the Hadoop MapReduce application for processing car sales data:

- CarSalesDriver.java – The main driver class that configures and initiates the MapReduce job.
- CarSalesMapper.java – The mapper class responsible for processing input records and emitting key-value pairs.
- CarSalesReducer.java – The reducer class that aggregates and processes the mapped data to generate the final output.

RESULTS

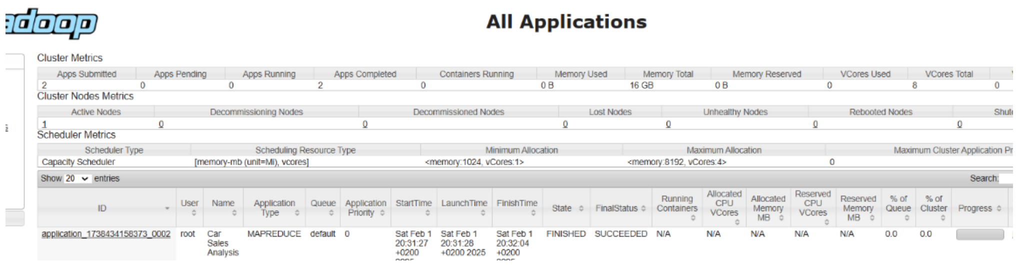
The job was successfully executed, as confirmed by the Hadoop file system logs. Below, you can find the terminal logs that document the execution process, along with a sample of the generated output.

This MapReduce job processes car sales data to generate a results file that, for each seller and month pair (e.g., *"kia motors america inc:2024-12"*), identifies:

- The car with the largest *(selling price - MMR)* difference.
- The specific difference value.
- The average difference across all cars for the same seller and month pair (e.g., *"Kia Sorento LX: 1000, avg: 250.46"*).

These results confirm the correctness of the data processing and ensure that the required metrics were computed accurately.

HDFS Interface


All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total
2	0	0	2	0	0 B	16 GB	0 B	0	8

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	0	0	0	0

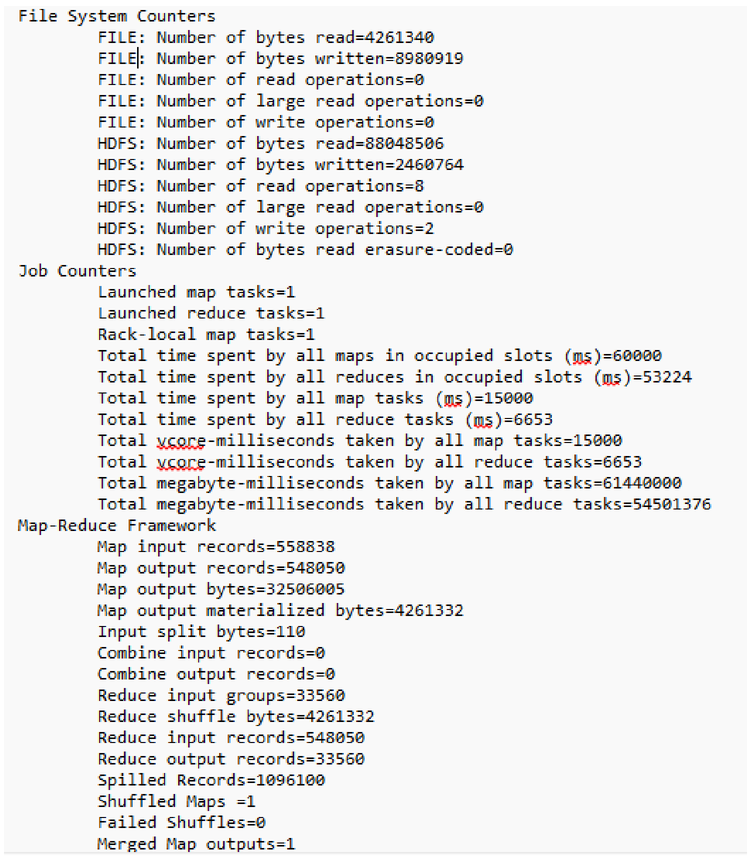
Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>

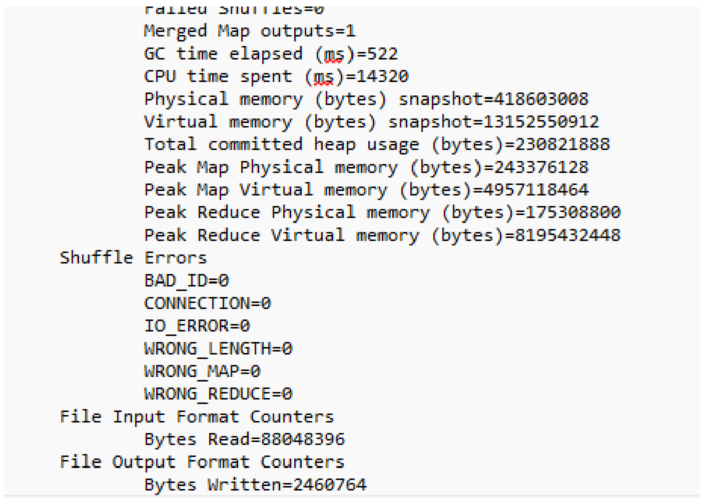
Show 20 entries — Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% of Queue	% of Cluster	Progress
application_1738434158373_0002	root	Car Sales Analysis	MAPREDUCE	default	0	Sat Feb 1 20:31:27 +0200	Sat Feb 1 20:31:28 +0200 2025	Sat Feb 1 20:32:04 +0200	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0	

Logs

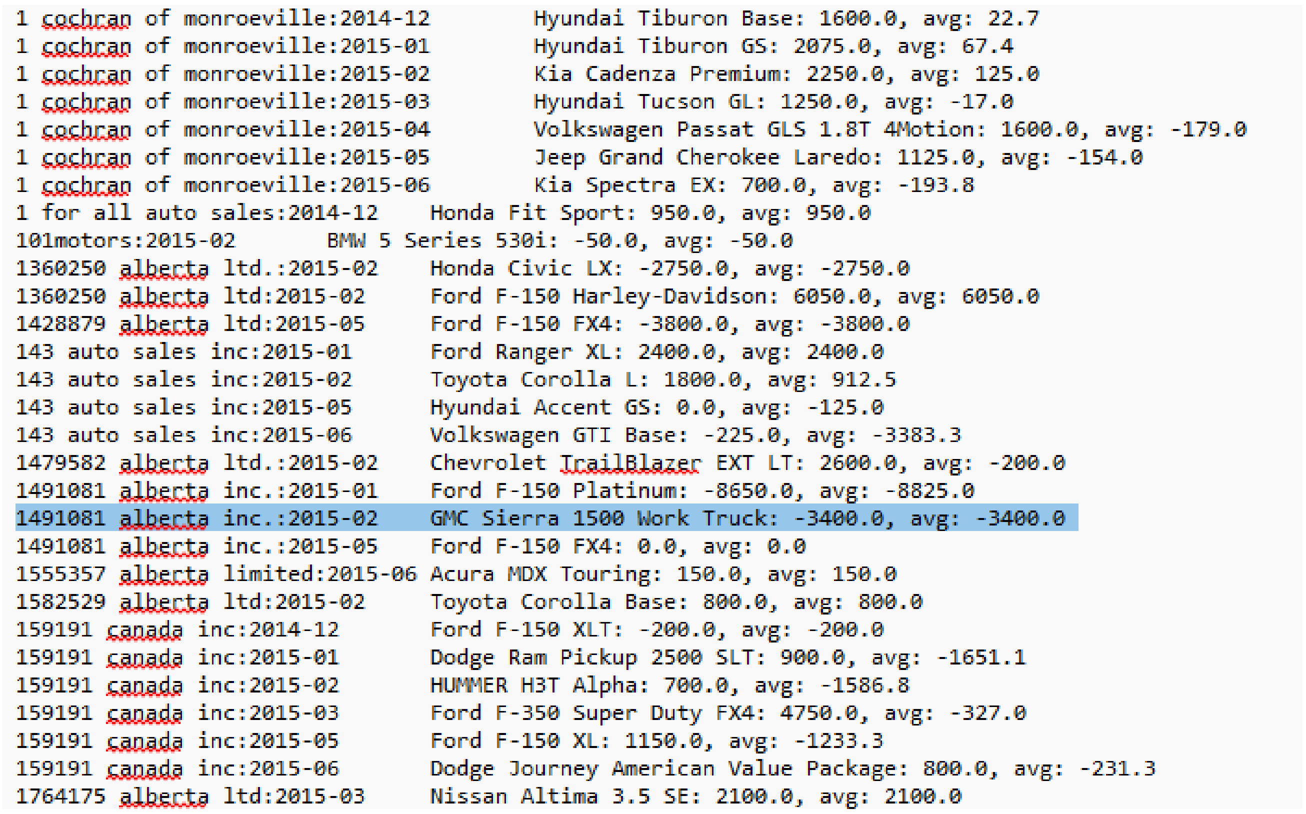


```
File System Counters
        FILE: Number of bytes read=4261340
        FILE: Number of bytes written=8980919
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=88048506
        HDFS: Number of bytes written=2460764
        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
        Rack-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=60000
        Total time spent by all reduces in occupied slots (ms)=53224
        Total time spent by all map tasks (ms)=15000
        Total time spent by all reduce tasks (ms)=6653
        Total vcore-milliseconds taken by all map tasks=15000
        Total vcore-milliseconds taken by all reduce tasks=6653
        Total megabyte-milliseconds taken by all map tasks=61440000
        Total megabyte-milliseconds taken by all reduce tasks=54501376
Map-Reduce Framework
        Map input records=558838
        Map output records=548050
        Map output bytes=32506005
        Map output materialized bytes=4261332
        Input split bytes=110
        Combine input records=0
        Combine output records=0
        Reduce input groups=33560
        Reduce shuffle bytes=4261332
        Reduce input records=548050
        Reduce output records=33560
        Spilled Records=1096100
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
```



```
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=522
                CPU time spent (ms)=14320
                Physical memory (bytes) snapshot=418603008
                Virtual memory (bytes) snapshot=13152550912
                Total committed heap usage (bytes)=230821888
                Peak Map Physical memory (bytes)=243376128
                Peak Map Virtual memory (bytes)=4957118464
                Peak Reduce Physical memory (bytes)=175308800
                Peak Reduce Virtual memory (bytes)=8195432448
        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=88048396
        File Output Format Counters
                Bytes Written=2460764
```

Output



```
1 cochran of monroeville:2014-12        Hyundai Tiburon Base: 1600.0, avg: 22.7
1 cochran of monroeville:2015-01        Hyundai Tiburon GS: 2075.0, avg: 67.4
1 cochran of monroeville:2015-02        Kia Cadenza Premium: 2250.0, avg: 125.0
1 cochran of monroeville:2015-03        Hyundai Tucson GL: 1250.0, avg: -17.0
1 cochran of monroeville:2015-04        Volkswagen Passat GLS 1.8T 4Motion: 1600.0, avg: -179.0
1 cochran of monroeville:2015-05        Jeep Grand Cherokee Laredo: 1125.0, avg: -154.0
1 cochran of monroeville:2015-06        Kia Spectra EX: 700.0, avg: -193.8
1 for all auto sales:2014-12    Honda Fit Sport: 950.0, avg: 950.0
101motors:2015-02       BMW 5 Series 530i: -50.0, avg: -50.0
1360250 alberta ltd.:2015-02    Honda Civic LX: -2750.0, avg: -2750.0
1360250 alberta ltd:2015-02     Ford F-150 Harley-Davidson: 6050.0, avg: 6050.0
1428879 alberta ltd:2015-05     Ford F-150 FX4: -3800.0, avg: -3800.0
143 auto sales inc:2015-01      Ford Ranger XL: 2400.0, avg: 2400.0
143 auto sales inc:2015-02      Toyota Corolla L: 1800.0, avg: 912.5
143 auto sales inc:2015-05      Hyundai Accent GS: 0.0, avg: -125.0
143 auto sales inc:2015-06      Volkswagen GTI Base: -225.0, avg: -3383.3
1479582 alberta ltd.:2015-02    Chevrolet TrailBlazer EXT LT: 2600.0, avg: -200.0
1491081 alberta inc.:2015-01    Ford F-150 Platinum: -8650.0, avg: -8825.0
1491081 alberta inc.:2015-02    GMC Sierra 1500 Work Truck: -3400.0, avg: -3400.0
1491081 alberta inc.:2015-05    Ford F-150 FX4: 0.0, avg: 0.0
1555357 alberta limited:2015-06 Acura MDX Touring: 150.0, avg: 150.0
1582529 alberta ltd:2015-02     Toyota Corolla Base: 800.0, avg: 800.0
159191 canada inc:2014-12       Ford F-150 XLT: -200.0, avg: -200.0
159191 canada inc:2015-01       Dodge Ram Pickup 2500 SLT: 900.0, avg: -1651.1
159191 canada inc:2015-02       HUMMER H3T Alpha: 700.0, avg: -1586.8
159191 canada inc:2015-03       Ford F-350 Super Duty FX4: 4750.0, avg: -327.0
159191 canada inc:2015-05       Ford F-150 XL: 1150.0, avg: -1233.3
159191 canada inc:2015-06       Dodge Journey American Value Package: 800.0, avg: -231.3
1764175 alberta ltd:2015-03     Nissan Altima 3.5 SE: 2100.0, avg: 2100.0
```