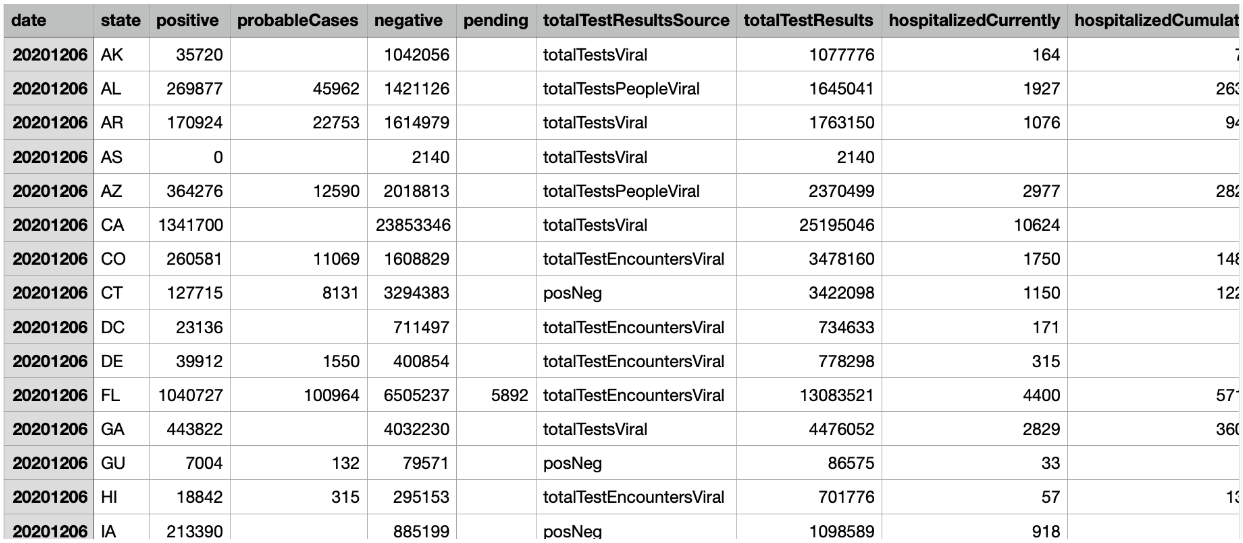
Project Data Ingestion Report

Name: Rajan Shantanu Chaturvedi NetID: rsc9044

Background:

I have a data set which contains all the Covid test related information of the different states. I have written a mapreduce program to find the cumulative tests counts of each state for the year of 2020, which can be combined with total positive cases data to find out any relevant insights.

Data:



date	state	positive	probableCases	negative	pending	totalTestResultsSource	totalTestResults	hospitalizedCurrently	hospitalizedCumulat
20201206	AK	35720		1042056		totalTestsViral	1077776	164	7
20201206	AL	269877	45962	1421126		totalTestsPeopleViral	1645041	1927	263
20201206	AR	170924	22753	1614979		totalTestsViral	1763150	1076	94
20201206	AS	0		2140		totalTestsViral	2140		
20201206	AZ	364276	12590	2018813		totalTestsPeopleViral	2370499	2977	282
20201206	CA	1341700		23853346		totalTestsViral	25195046	10624	
20201206	CO	260581	11069	1608829		totalTestEncountersViral	3478160	1750	148
20201206	CT	127715	8131	3294383		posNeg	3422098	1150	122
20201206	DC	23136		711497		totalTestEncountersViral	734633	171	
20201206	DE	39912	1550	400854		totalTestEncountersViral	778298	315	
20201206	FL	1040727	100964	6505237	5892	totalTestEncountersViral	13083521	4400	571
20201206	GA	443822		4032230		totalTestsViral	4476052	2829	360
20201206	GU	7004	132	79571		posNeg	86575	33	
20201206	HI	18842	315	295153		totalTestEncountersViral	701776	57	13
20201206	IA	213390		885199		posNeg	1098589	918	

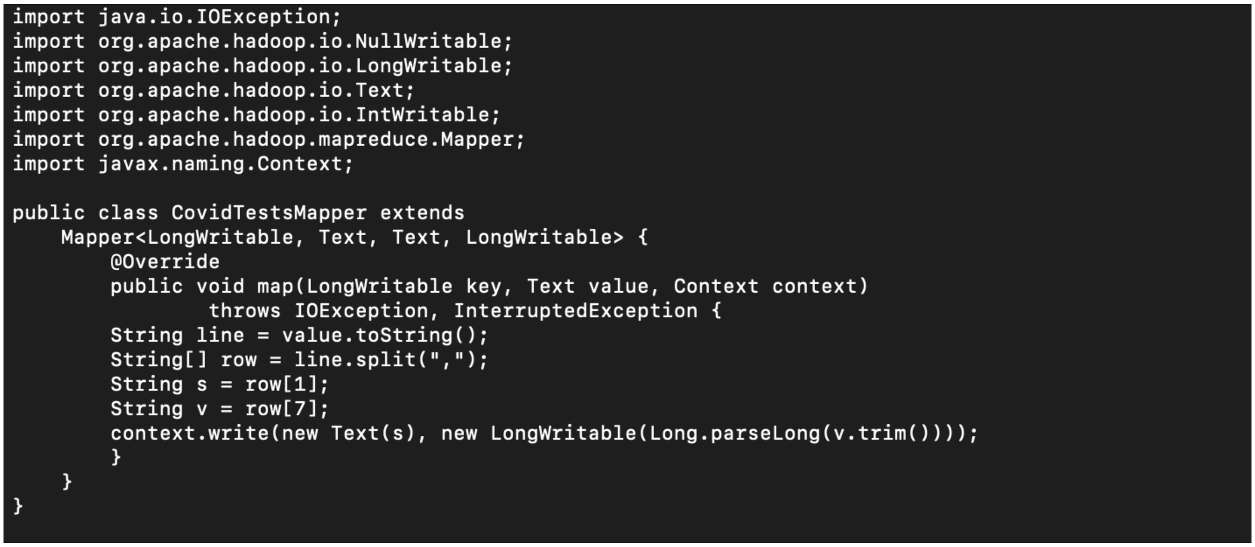
Target:

To find out the number Covid test took place in the different states of the USA in 2020.

MapReduce Program:

a) First I have written a mapper which is mapping totalTestResults of each state as a key-value pair.
Eg: (AK, 1077776)

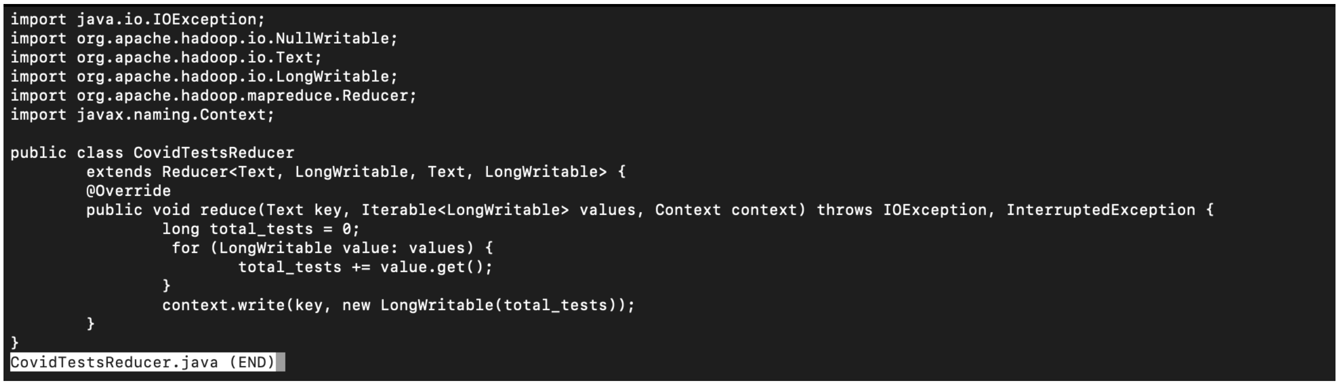
Code:



```java
import java.io.IOException;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.mapreduce.Mapper;
import javax.naming.Context;

public class CovidTestsMapper extends
    Mapper<LongWritable, Text, Text, LongWritable> {
        @Override
        public void map(LongWritable key, Text value, Context context)
                throws IOException, InterruptedException {
        String line = value.toString();
        String[] row = line.split(",");
        String s = row[1];
        String v = row[7];
        context.write(new Text(s), new LongWritable(Long.parseLong(v.trim())));
        }
    }
}
```

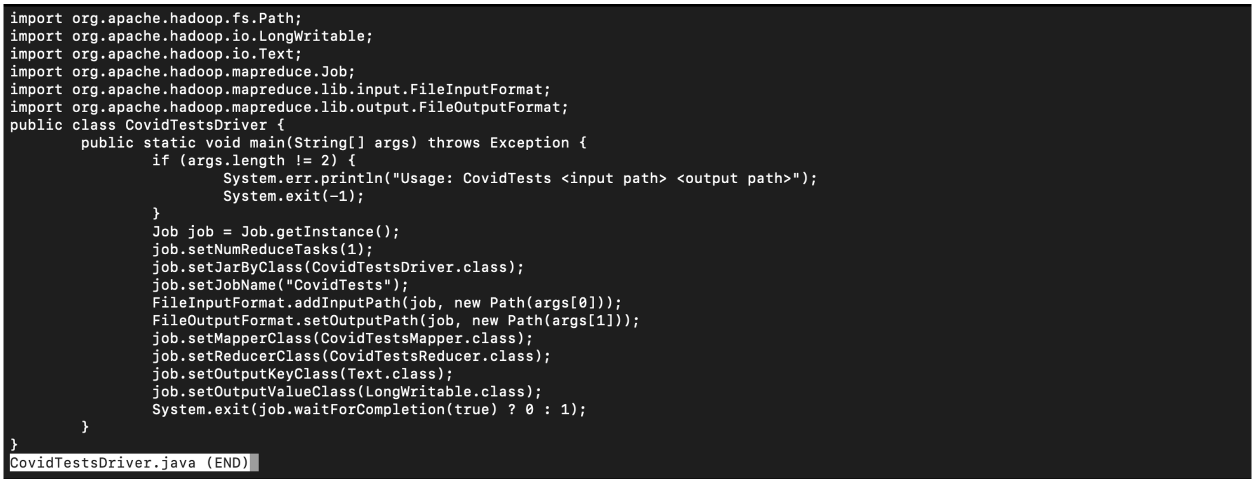
b) Then, I wrote a reducer program to add all values of the same keys-values pairs.
Eg: (AK, 50, 60, 90) → (AK, 200)

Code:



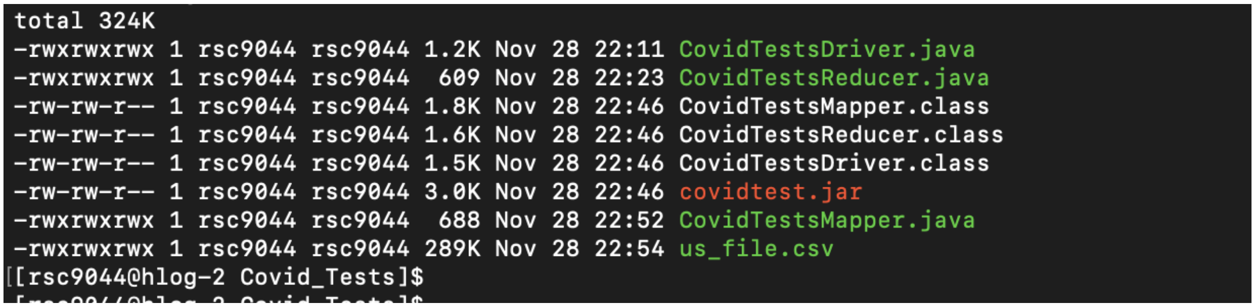
```java
import java.io.IOException;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.mapreduce.Reducer;
import javax.naming.Context;

public class CovidTestsReducer
        extends Reducer<Text, LongWritable, Text, LongWritable> {
        @Override
        public void reduce(Text key, Iterable<LongWritable> values, Context context) throws IOException, InterruptedException {
                long total_tests = 0;
                 for (LongWritable value: values) {
                        total_tests += value.get();
                }
                context.write(key, new LongWritable(total_tests));
        }
}
CovidTestsReducer.java (END)
```

c) Finally, I wrote the driver code to define mapper and reducer class and output data types and run the MapReduce job.



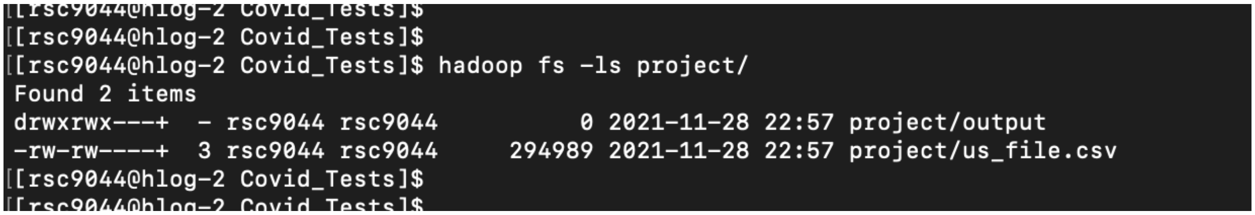
```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class CovidTestsDriver {
        public static void main(String[] args) throws Exception {
                if (args.length != 2) {
                        System.err.println("Usage: CovidTests <input path> <output path>");
                        System.exit(-1);
                }
                Job job = Job.getInstance();
                job.setNumReduceTasks(1);
                job.setJarByClass(CovidTestsDriver.class);
                job.setJobName("CovidTests");
                FileInputFormat.addInputPath(job, new Path(args[0]));
                FileOutputFormat.setOutputPath(job, new Path(args[1]));
                job.setMapperClass(CovidTestsMapper.class);
                job.setReducerClass(CovidTestsReducer.class);
                job.setOutputKeyClass(Text.class);
                job.setOutputValueClass(LongWritable.class);
                System.exit(job.waitForCompletion(true) ? 0 : 1);
        }
}
CovidTestsDriver.java (END)
```

d) After this I compiled all the Java files and created a Jar file.



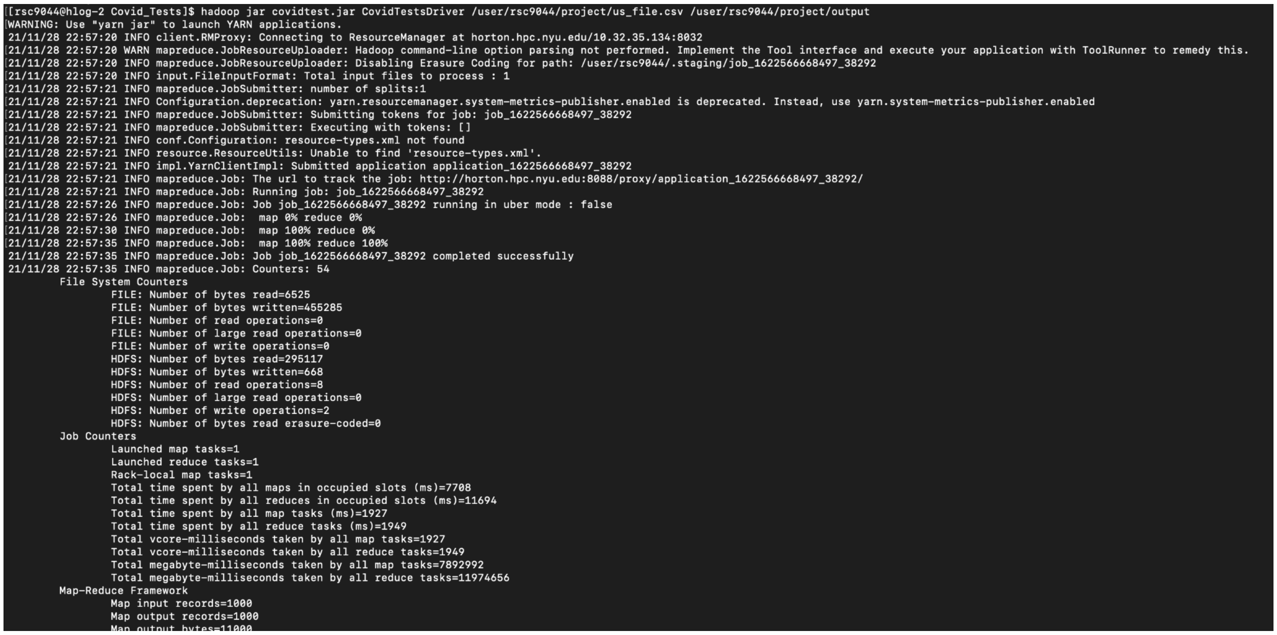
```
total 324K
-rwxrwxrwx 1 rsc9044 rsc9044 1.2K Nov 28 22:11 CovidTestsDriver.java
-rwxrwxrwx 1 rsc9044 rsc9044  609 Nov 28 22:23 CovidTestsReducer.java
-rw-rw-r-- 1 rsc9044 rsc9044 1.8K Nov 28 22:46 CovidTestsMapper.class
-rw-rw-r-- 1 rsc9044 rsc9044 1.6K Nov 28 22:46 CovidTestsReducer.class
-rw-rw-r-- 1 rsc9044 rsc9044 1.5K Nov 28 22:46 CovidTestsDriver.class
-rw-rw-r-- 1 rsc9044 rsc9044 3.0K Nov 28 22:46 covidtest.jar
-rwxrwxrwx 1 rsc9044 rsc9044  688 Nov 28 22:52 CovidTestsMapper.java
-rwxrwxrwx 1 rsc9044 rsc9044 289K Nov 28 22:54 us_file.csv
[[rsc9044@hlog-2 Covid_Tests]$
[[rsc9044@hlog-2 Covid_Tests]$
```

e) Then I transferred the Data file into the HDFS.

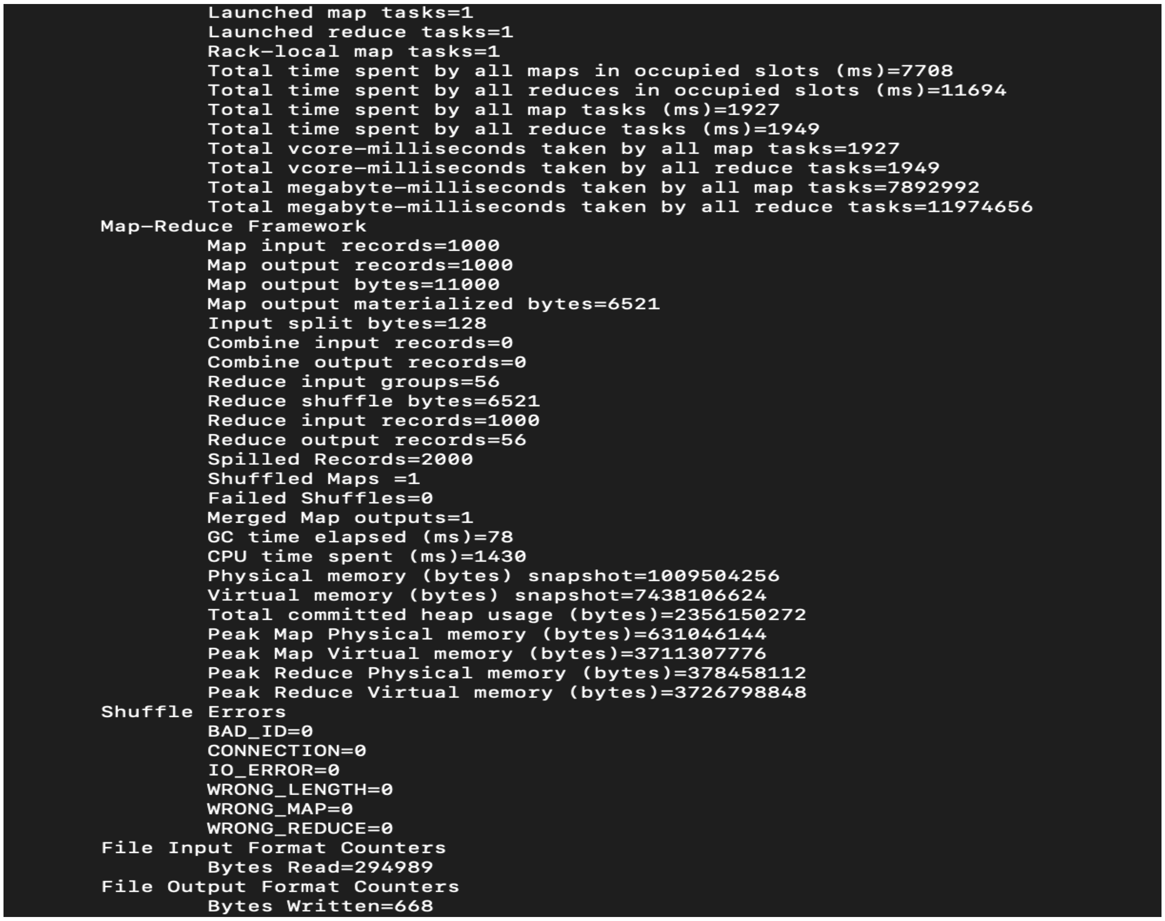


```
[[rsc9044@hlog-2 Covid_Tests]$
[[rsc9044@hlog-2 Covid_Tests]$
[[rsc9044@hlog-2 Covid_Tests]$ hadoop fs -ls project/
Found 2 items
drwxrwx---+  - rsc9044 rsc9044          0 2021-11-28 22:57 project/output
-rw-rw----+  3 rsc9044 rsc9044     294989 2021-11-28 22:57 project/us_file.csv
[[rsc9044@hlog-2 Covid_Tests]$
[[rsc9044@hlog-2 Covid_Tests]$
```

f) Then I ran the mapreduce job to get the output. Map Reduce job ran successfully.

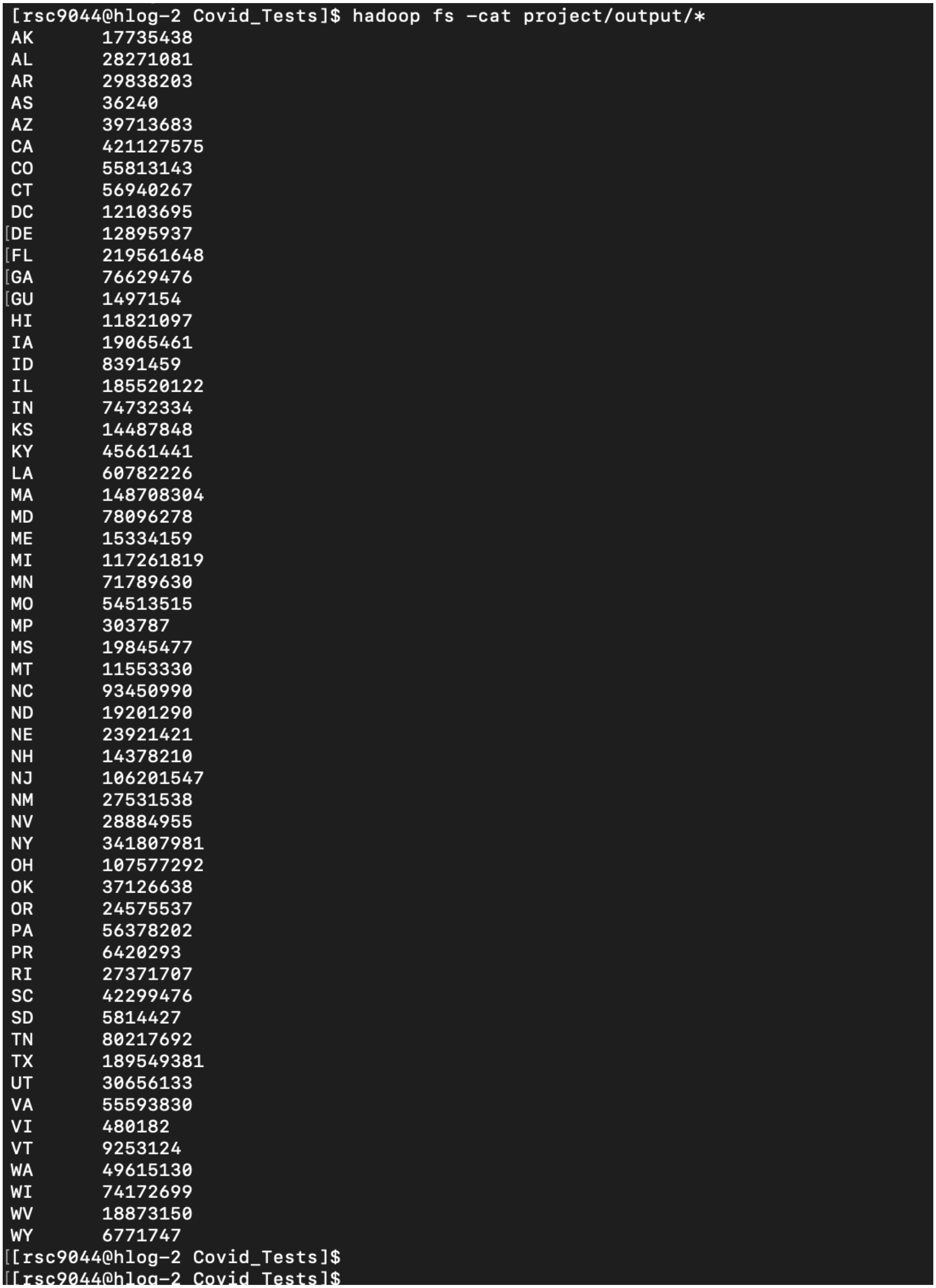


```
[[rsc9044@hlog-2 Covid_Tests]$ hadoop jar covidtest.jar CovidTestsDriver /user/rsc9044/project/us_file.csv /user/rsc9044/project/output
[WARNING: Use "yarn jar" to launch YARN applications.
 21/11/28 22:57:20 INFO client.RMProxy: Connecting to ResourceManager at horton.hpc.nyu.edu/10.32.35.134:8032
[21/11/28 22:57:20 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
[21/11/28 22:57:20 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /user/rsc9044/.staging/job_1622566668497_38292
[21/11/28 22:57:20 INFO input.FileInputFormat: Total input files to process : 1
[21/11/28 22:57:21 INFO mapreduce.JobSubmitter: number of splits:1
[21/11/28 22:57:21 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
[21/11/28 22:57:21 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1622566668497_38292
[21/11/28 22:57:21 INFO mapreduce.JobSubmitter: Executing with tokens: []
[21/11/28 22:57:21 INFO conf.Configuration: resource-types.xml not found
[21/11/28 22:57:21 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
 21/11/28 22:57:21 INFO impl.YarnClientImpl: Submitted application application_1622566668497_38292
[21/11/28 22:57:21 INFO mapreduce.Job: The url to track the job: http://horton.hpc.nyu.edu:8088/proxy/application_1622566668497_38292/
[21/11/28 22:57:21 INFO mapreduce.Job: Running job: job_1622566668497_38292
[21/11/28 22:57:26 INFO mapreduce.Job: Job job_1622566668497_38292 running in uber mode : false
[21/11/28 22:57:26 INFO mapreduce.Job:  map 0% reduce 0%
[21/11/28 22:57:30 INFO mapreduce.Job:  map 100% reduce 0%
[21/11/28 22:57:35 INFO mapreduce.Job:  map 100% reduce 100%
 21/11/28 22:57:35 INFO mapreduce.Job: Job job_1622566668497_38292 completed successfully
 21/11/28 22:57:35 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=6525
                FILE: Number of bytes written=455285
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=295117
                HDFS: Number of bytes written=668
                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)=7708
                Total time spent by all reduces in occupied slots (ms)=11694
                Total time spent by all map tasks (ms)=1927
                Total time spent by all reduce tasks (ms)=1949
                Total vcore-milliseconds taken by all map tasks=1927
                Total vcore-milliseconds taken by all reduce tasks=1949
                Total megabyte-milliseconds taken by all map tasks=7892992
                Total megabyte-milliseconds taken by all reduce tasks=11974656
        Map-Reduce Framework
                Map input records=1000
                Map output records=1000
                Map output bytes=11000
```



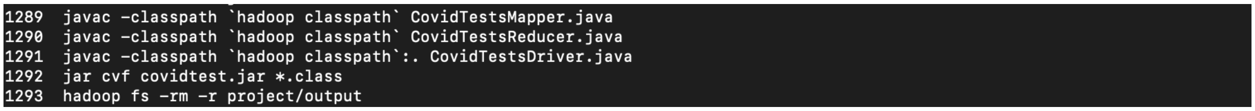
```
                Launched map tasks=1
                Launched reduce tasks=1
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=7708
                Total time spent by all reduces in occupied slots (ms)=11694
                Total time spent by all map tasks (ms)=1927
                Total time spent by all reduce tasks (ms)=1949
                Total vcore-milliseconds taken by all map tasks=1927
                Total vcore-milliseconds taken by all reduce tasks=1949
                Total megabyte-milliseconds taken by all map tasks=7892992
                Total megabyte-milliseconds taken by all reduce tasks=11974656
        Map-Reduce Framework
                Map input records=1000
                Map output records=1000
                Map output bytes=11000
                Map output materialized bytes=6521
                Input split bytes=128
                Combine input records=0
                Combine output records=0
                Reduce input groups=56
                Reduce shuffle bytes=6521
                Reduce input records=1000
                Reduce output records=56
                Spilled Records=2000
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=78
                CPU time spent (ms)=1430
                Physical memory (bytes) snapshot=1009504256
                Virtual memory (bytes) snapshot=7438106624
                Total committed heap usage (bytes)=2356150272
                Peak Map Physical memory (bytes)=631046144
                Peak Map Virtual memory (bytes)=3711307776
                Peak Reduce Physical memory (bytes)=378458112
                Peak Reduce Virtual memory (bytes)=3726798848
        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=294989
        File Output Format Counters
                Bytes Written=668
```

Output:

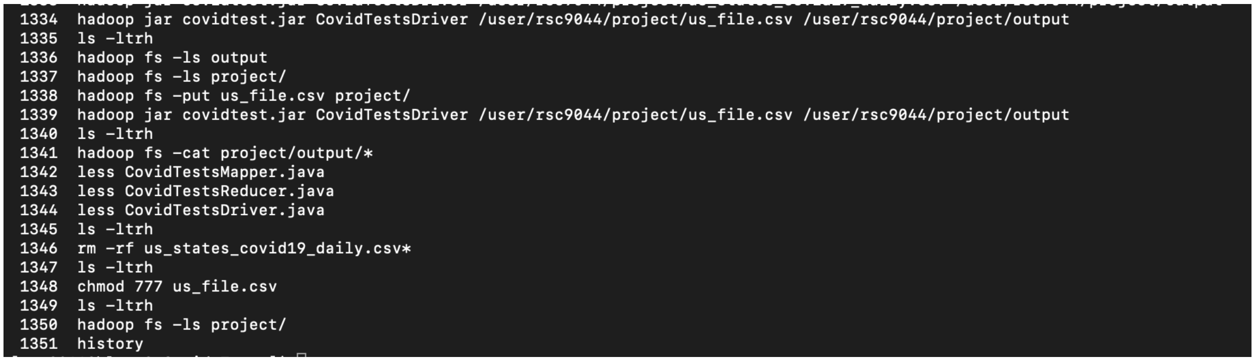


```
[rsc9044@hlog-2 Covid_Tests]$ hadoop fs -cat project/output/*
AK      17735438
AL      28271081
AR      29838203
AS      36240
AZ      39713683
CA      421127575
CO      55813143
CT      56940267
DC      12103695
[DE      12895937
[FL      219561648
[GA      76629476
[GU      1497154
HI      11821097
IA      19065461
ID      8391459
IL      185520122
IN      74732334
KS      14487848
KY      45661441
LA      60782226
MA      148708304
MD      78096278
ME      15334159
MI      117261819
MN      71789630
MO      54513515
MP      303787
MS      19845477
MT      11553330
NC      93450990
ND      19201290
NE      23921421
NH      14378210
NJ      106201547
NM      27531538
NV      28884955
NY      341807981
OH      107577292
OK      37126638
OR      24575537
PA      56378202
PR      6420293
RI      27371707
SC      42299476
SD      5814427
TN      80217692
TX      189549381
UT      30656133
VA      55593830
VI      480182
VT      9253124
WA      49615130
WI      74172699
WV      18873150
WY      6771747
[[rsc9044@hlog-2 Covid_Tests]$
[[rsc9044@hlog-2 Covid_Tests]$
```

Command History:



```
1289  javac -classpath `hadoop classpath` CovidTestsMapper.java
1290  javac -classpath `hadoop classpath` CovidTestsReducer.java
1291  javac -classpath `hadoop classpath`:. CovidTestsDriver.java
1292  jar cvf covidtest.jar *.class
1293  hadoop fs -rm -r project/output
```



```
 1334  hadoop jar covidtest.jar CovidTestsDriver /user/rsc9044/project/us_file.csv /user/rsc9044/project/output
 1335  ls -ltrh
 1336  hadoop fs -ls output
 1337  hadoop fs -ls project/
 1338  hadoop fs -put us_file.csv project/
 1339  hadoop jar covidtest.jar CovidTestsDriver /user/rsc9044/project/us_file.csv /user/rsc9044/project/output
 1340  ls -ltrh
 1341  hadoop fs -cat project/output/*
 1342  less CovidTestsMapper.java
 1343  less CovidTestsReducer.java
 1344  less CovidTestsDriver.java
 1345  ls -ltrh
 1346  rm -rf us_states_covid19_daily.csv*
 1347  ls -ltrh
 1348  chmod 777 us_file.csv
 1349  ls -ltrh
 1350  hadoop fs -ls project/
 1351  history
```

Notes:

- In the project proposal, it is mentioned that I will be doing Covid Spending Data Visualization. But later on I realized that the USA Government Covid spending data set is too large to work with. Hence, i went with Covid Testing dataset which is manage enough to work with.
- For Simplicity, with this report and codes i am attaching small portion of dataset so that attachment file should not be very large in size.