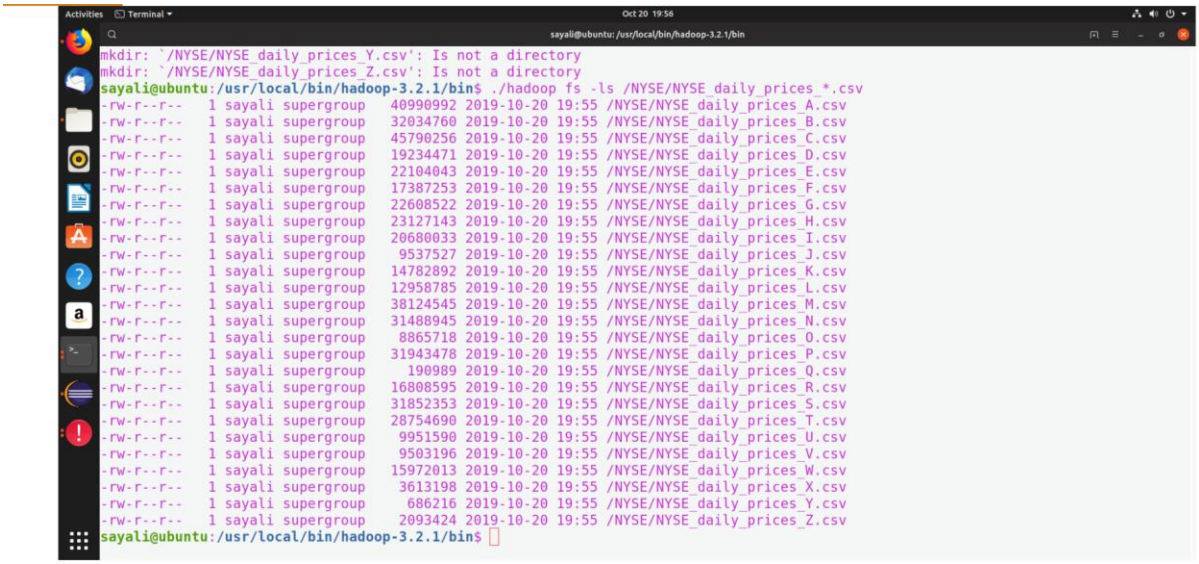
Map reduce on NYSE dataset

1] Write a MapReduce to find the Max price of stock_price_high for each stock. Capture the running time programmatically.

Copy all the csv files into different HDFS

./hadoop fs -copyFromLocal /home/sayali/Downloads/NYSE/NYSE_daily_prices_*.csv /NYSE



Mapper:

```
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import com.google.common.base.Splitter;
import com.google.common.collect.Lists;

public class NyseMapper extends Mapper<LongWritable, Text, Text, FloatWritable>
{        @Override
         public void map(LongWritable key, Text value, Context context)
         throws IOException, InterruptedException {
                  String line = value.toString();
                   try {
                if (key.get() == 0 && value.toString().contains("header") /*Some condition satisfying it is header*/)
                   return;
                else {
                 List<String> items = Lists.newArrayList(Splitter.on(',').split(line));
                    String stock = items.get(1);
                    Float closePrice = Float.parseFloat(items.get(4));
                 context.write(new Text(stock), new FloatWritable(closePrice));
                }
              } catch (Exception e) {
```

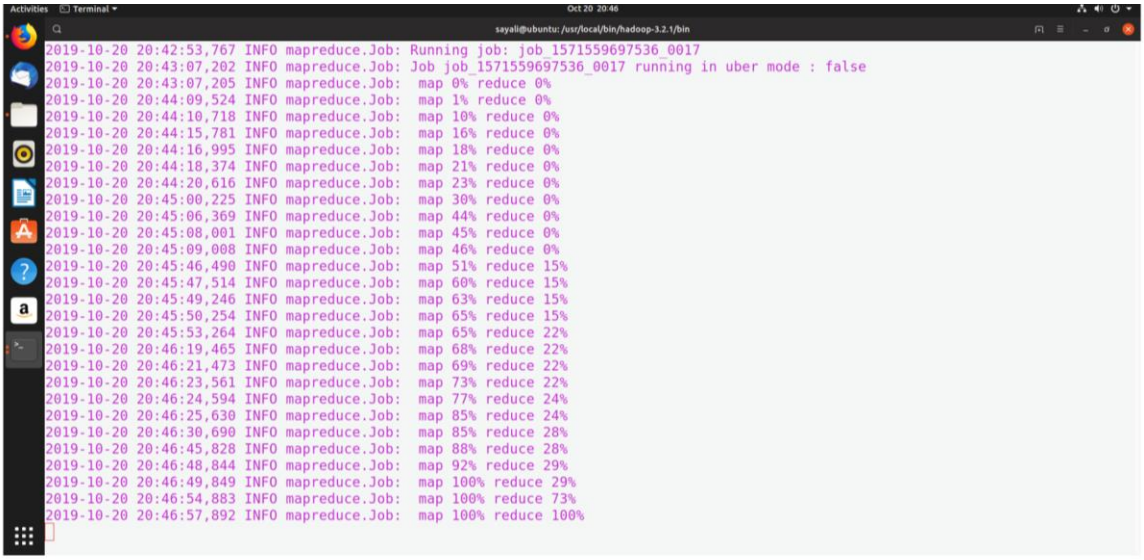
```
                    e.printStackTrace(); }
```

Reducer:

```
package com.sayali.NYSEPart4;
import java.io.IOException;
import org.apache.hadoop.io.FloatWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class NyseReducer extends Reducer<Text, FloatWritable, Text, FloatWritable> {
 public void reduce(Text key, Iterable<FloatWritable> values, Context context)
throws IOException, InterruptedException {

float maxHighPrice = Float.MIN_VALUE;
  //Iterate all high prices and calculate maximum
    for (FloatWritable value : values) {
    maxHighPrice = Math.max(maxHighPrice, value.get());
            }
 //Write output
context.write(key, new FloatWritable(maxHighPrice));
}  }
```

OUTPUT:

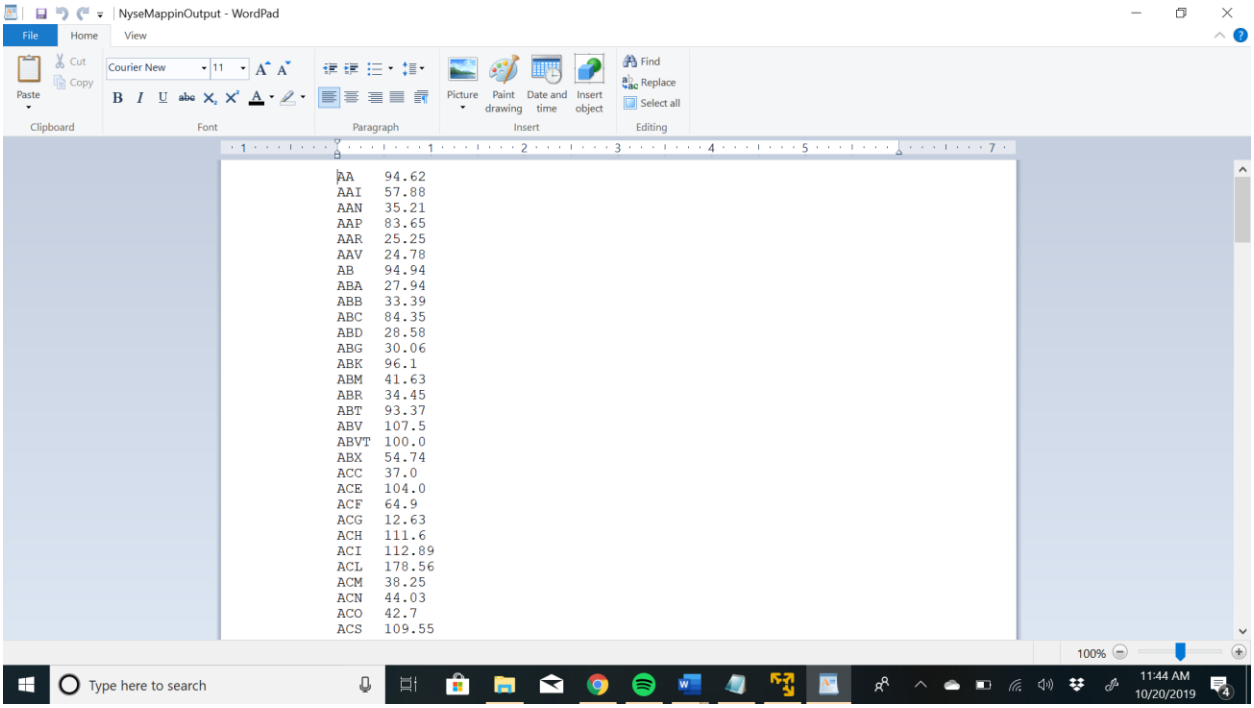
./hadoop jar /home/sayali/Desktop/nyse.jar com.sayali.NYSEPart4.App /NYSE/NYSE_daily_prices_*.csv /maxStockPrice

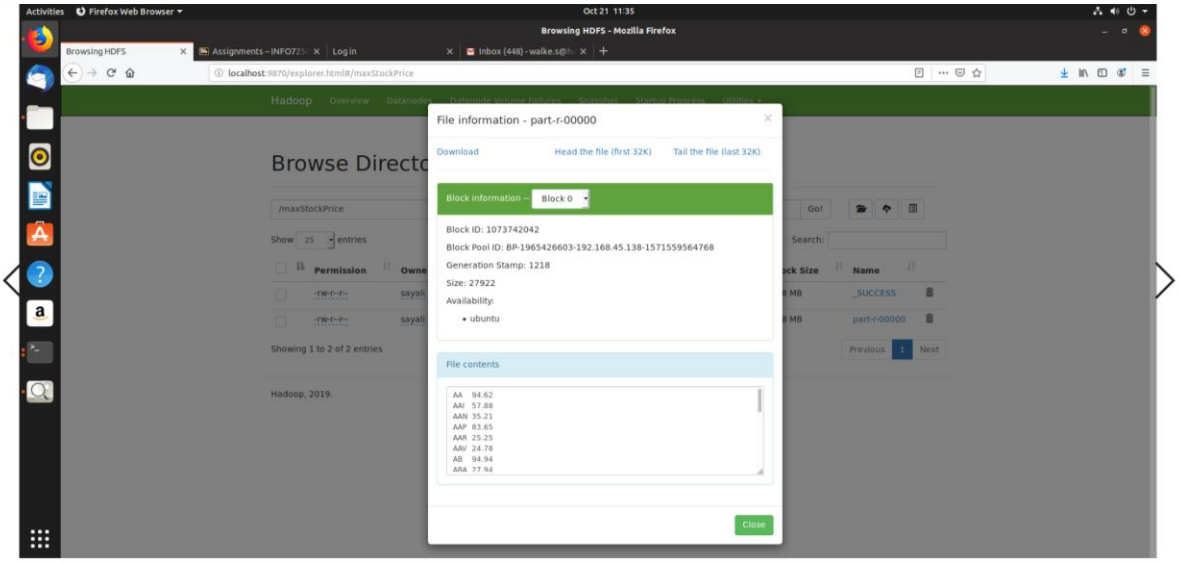


The Map reduce job when NYSE was not merged i.e DailyPrices_A to DailyPrices_Z as separate:

The time required to finish the job can be calculated from time log shown in following screenshot

47:57 – 43:07 = 4 mins 50 sec





THE NYSE merged into single file:

JAVA Code to merge:

```
package sayali.LoadToHdfs;
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataInputStream;
import org.apache.hadoop.fs.FSDataOutputStream;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
 public class HDFSFileWrite {
```

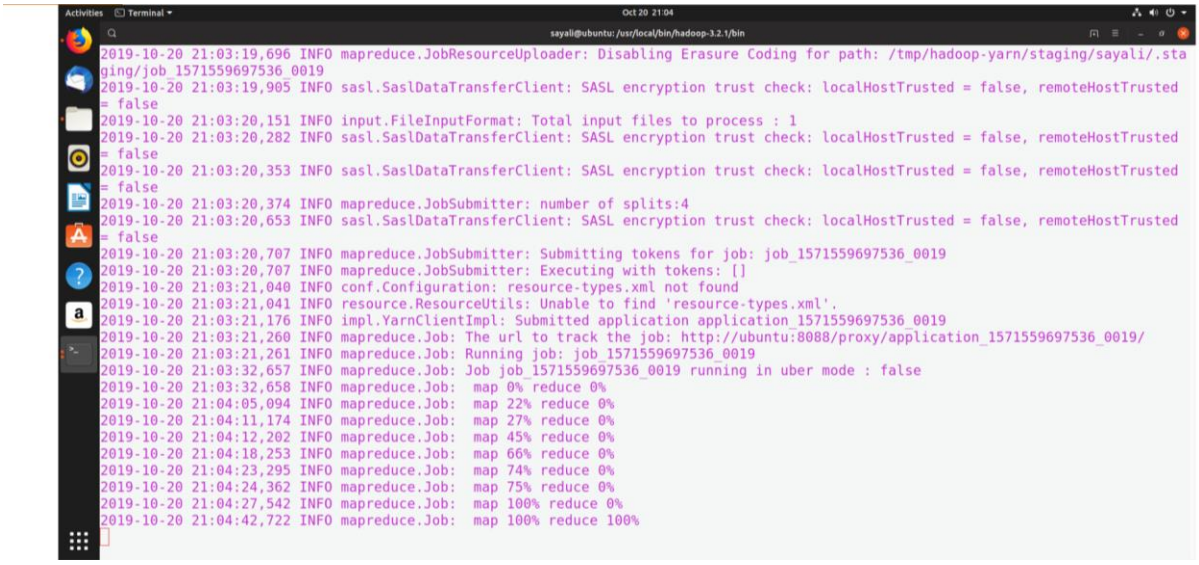
```java
  public static void main(String[] args) {
      Configuration conf = new Configuration();
      try {
         FileSystem fs = FileSystem.get(conf);
         // Hadoop DFS Path - Input & Output file
         Path inFile = new Path(args[0]);
         Path outFile = new Path(args[1]);
         // Verification
         if (!fs.exists(inFile)) {
            System.out.println("Input file not found");
            throw new IOException("Input file not found");
         }
         if (fs.exists(outFile)) {
            System.out.println("Output file already exists");
            throw new IOException("Output file already exists");
         }

         // open and read from file
         FSDataInputStream in = fs.open(inFile);
         // Create file to write
         FSDataOutputStream out = fs.create(outFile);
          byte buffer[] = new byte[256];
         try {
            int bytesRead = 0;
            while ((bytesRead = in.read(buffer)) > 0) {
               out.write(buffer, 0, bytesRead);
             }
         } catch (IOException e) {
            System.out.println("Error while copying file");
         } finally {
            in.close();
            out.close();
         }

      } catch (IOException e) {
         // TODO Auto-generated catch block
         e.printStackTrace();
      }
   }
}
```

```
sayali@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -ls /NYSEMerged
-rw-r--r--   1 sayali supergroup  511085627 2019-10-20 20:39 /NYSEMerged
sayali@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -ls /NYSEMerged/
-rw-r--r--   1 sayali supergroup  511085627 2019-10-20 20:39 /NYSEMerged
sayali@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -head /NYSEMerged
2019-10-20 20:41:11,480 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted
= false
exchange,stock_symbol,date,stock_price_open,stock_price_high,stock_price_low,stock_price_close,stock_volume,stock_price_adj_close
NYSE,AEA,2010-02-08,4.42,4.42,4.21,4.24,205500,4.24
NYSE,AEA,2010-02-05,4.42,4.54,4.22,4.41,194300,4.41
NYSE,AEA,2010-02-04,4.55,4.69,4.39,4.42,233800,4.42
NYSE,AEA,2010-02-03,4.65,4.69,4.50,4.55,182100,4.55
NYSE,AEA,2010-02-02,4.74,5.00,4.62,4.66,222700,4.66
NYSE,AEA,2010-02-01,4.84,4.92,4.68,4.75,194800,4.75
NYSE,AEA,2010-01-29,4.97,5.05,4.76,4.83,222900,4.83
NYSE,AEA,2010-01-28,5.12,5.22,4.81,4.98,283100,4.98
NYSE,AEA,2010-01-27,4.82,5.16,4.79,5.09,243500,5.09
NYSE,AEA,2010-01-26,5.18,5.18,4.81,4.84,554800,4.84
NYSE,AEA,2010-01-25,5.42,5.48,5.20,5.22,257300,5.22
NYSE,AEA,2010-01-22,5.52,5.59,5.31,5.37,260800,5.37
NYSE,AEA,2010-01-21,5.67,5.74,5.37,5.51,264300,5.51
NYSE,AEA,2010-01-20,5.65,5.70,5.53,5.66,244600,5.66
NYSE,AEA,2010-01-19,5.54,5.70,5.54,5.69,368000,5.69
NYSE,AEA,2010-01-15,5.48,5.55,5.33,5.54,435500,5.54
NYSE,AEA,2010-01-14,5.41,5.50,5.39,5.41,272200,5.41
NYSE,AEA,2sayali@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$
```

```
sayali@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -tail /NYSEMerged
2019-10-20 20:41:32,213 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted
= false
27.10,28.00,27.09,27.78,1697200,27.78
NYSE,ZMH,2001-08-31,26.80,27.20,26.50,27.20,1447500,27.20
NYSE,ZMH,2001-08-30,26.40,27.30,26.40,26.95,2443000,26.95
NYSE,ZMH,2001-08-29,26.02,26.75,25.95,26.40,2020500,26.40
NYSE,ZMH,2001-08-28,26.10,26.30,25.86,26.02,2378300,26.02
NYSE,ZMH,2001-08-27,26.60,26.80,26.30,26.66,1461800,26.66
NYSE,ZMH,2001-08-24,26.25,26.72,26.05,26.69,1909700,26.69
NYSE,ZMH,2001-08-23,27.15,27.19,26.73,26.75,2343000,26.75
NYSE,ZMH,2001-08-22,27.10,27.26,26.40,27.26,2549800,27.26
NYSE,ZMH,2001-08-21,27.20,27.44,27.10,27.34,1870500,27.34
NYSE,ZMH,2001-08-20,27.65,27.72,26.80,27.20,2066600,27.20
NYSE,ZMH,2001-08-17,27.70,28.00,26.53,28.00,1747200,28.00
NYSE,ZMH,2001-08-16,28.03,28.10,27.30,28.10,2132100,28.10
NYSE,ZMH,2001-08-15,27.75,28.40,27.70,28.08,2622100,28.08
NYSE,ZMH,2001-08-14,27.30,28.10,27.20,27.70,3667700,27.70
NYSE,ZMH,2001-08-13,27.75,27.85,26.85,27.30,4809800,27.30
NYSE,ZMH,2001-08-10,28.00,28.35,27.43,28.00,3193700,28.00
NYSE,ZMH,2001-08-09,28.10,28.45,27.55,28.36,2996900,28.36
sayali@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$
```

The map reduce job on merged file. The time taken for map reduce job on merged file can be calculated as follows:

04:42 – 03:32 = 1 min 10 secs



```
2019-10-20 21:03:19,696 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/sayali/.sta
ging/job_1571559697536_0019
2019-10-20 21:03:19,905 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted
= false
2019-10-20 21:03:20,151 INFO input.FileInputFormat: Total input files to process : 1
2019-10-20 21:03:20,282 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted
= false
2019-10-20 21:03:20,353 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted
= false
2019-10-20 21:03:20,374 INFO mapreduce.JobSubmitter: number of splits:4
2019-10-20 21:03:20,653 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted
= false
2019-10-20 21:03:20,707 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1571559697536_0019
2019-10-20 21:03:20,707 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-10-20 21:03:21,040 INFO conf.Configuration: resource-types.xml not found
2019-10-20 21:03:21,041 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2019-10-20 21:03:21,176 INFO impl.YarnClientImpl: Submitted application application_1571559697536_0019
2019-10-20 21:03:21,260 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1571559697536_0019/
2019-10-20 21:03:21,261 INFO mapreduce.Job: Running job: job_1571559697536_0019
2019-10-20 21:03:32,657 INFO mapreduce.Job: Job job_1571559697536_0019 running in uber mode : false
2019-10-20 21:03:32,658 INFO mapreduce.Job:  map 0% reduce 0%
2019-10-20 21:04:05,094 INFO mapreduce.Job:  map 22% reduce 0%
2019-10-20 21:04:11,174 INFO mapreduce.Job:  map 27% reduce 0%
2019-10-20 21:04:12,202 INFO mapreduce.Job:  map 45% reduce 0%
2019-10-20 21:04:18,253 INFO mapreduce.Job:  map 66% reduce 0%
2019-10-20 21:04:23,295 INFO mapreduce.Job:  map 74% reduce 0%
2019-10-20 21:04:24,362 INFO mapreduce.Job:  map 75% reduce 0%
2019-10-20 21:04:27,542 INFO mapreduce.Job:  map 100% reduce 0%
2019-10-20 21:04:42,722 INFO mapreduce.Job:  map 100% reduce 100%
```

Hence, the time taken for merged file (1 min 10 secs) is much less than time taken for individual files (4 mins 50 secs)