

230340325068 - Surya Dev Singh Jamwal - (Set A)

MapRed 15

Hive 15m

PySpark 10m

Q1. - MapReduce

Java code:

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.DoubleWritable;
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.Reducer.Context;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class AvgClosingPrice {
    public static class MappingClass extends Mapper<LongWritable, Text,
Text, DoubleWritable>{
        public void map(LongWritable Key, Text value, Context context) throws
IOException, InterruptedException {
            String[] str= value.toString().split(",");
            double closing = Double.parseDouble(str[6]);
            context.write(new Text(str[1]),new DoubleWritable(closing));
        }
    }
    public static class ReducingClass extends
Reducer<Text, DoubleWritable, Text, DoubleWritable> {
        public void reduce(Text key, Iterable<DoubleWritable> values, Context context)
throws IOException, InterruptedException {
            DoubleWritable resultval = new DoubleWritable();
            double sum = 0;
            double count = 0;
            for (DoubleWritable val : values) {
                sum +=val.get();
                count++;
            }
            double avg = sum/count;
            resultval.set(avg);
            context.write(key, resultval);
        }
    }
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
```

```

    Job job = Job.getInstance(conf, " Average Closing Price ");
    job.setJarByClass(AvgClosingPrice.class);
    job.setMapperClass(MappingClass.class);
    job.setReducerClass(ReducingClass.class);
    job.setNumReduceTasks(1);
    job.setMapOutputKeyClass(Text.class);
    job.setMapOutputValueClass(DoubleWritable.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);
}
}

```

FTP upload for JAR file and NYSE.csv dataset:

```
[bigdatalab456422@ip-10-1-1-204 ~]$ ls -l ModEndAvgClosingPrice.jar
NYSE.csv
```

Screenshot:

```
[bigdatalab456422@ip-10-1-1-204 ~]$ ls -l ModEndAvgClosingPrice.jar NYSE.csv
-rw-rw-r-- 1 bigdatalab456422 bigdatalab456422      4070 Jun 10 05:48 ModEndAvgClosingPrice.jar
-rw-rw-r-- 1 bigdatalab456422 bigdatalab456422 40990862 Jun 10 05:41 NYSE.csv
[bigdatalab456422@ip-10-1-1-204 ~]$ █
```

Jar File Contents:

```
[bigdatalab456422@ip-10-1-1-204 ~]$ jar tvf ModEndAvgClosingPrice.jar
```

Screenshot:

```
[bigdatalab456422@ip-10-1-1-204 ~]$ jar tvf ModEndAvgClosingPrice.jar
 25 Sat Jun 10 11:18:14 UTC 2023 META-INF/MANIFEST.MF
 640 Sat Jun 10 10:40:44 UTC 2023 .classpath
 385 Sat Jun 10 10:39:02 UTC 2023 .project
2245 Sat Jun 10 11:18:04 UTC 2023 AvgClosingPrice$MappingClass.class
2442 Sat Jun 10 11:18:04 UTC 2023 AvgClosingPrice$ReducingClass.class
1830 Sat Jun 10 11:18:04 UTC 2023 AvgClosingPrice.class
[bigdatalab456422@ip-10-1-1-204 ~]$
```

Hadoop MapReduce Command using Jar file and java class:

```
[bigdatalab456422@ip-10-1-1-204 ~]$ hadoop jar
ModEndAvgClosingPrice.jar AvgClosingPrice NYSE.csv ModEnd/q1
```

Screenshots:

```
[bigdatalab456422@ip-10-1-1-204 ~]$ hadoop jar ModEndAvgClosingPrice.jar AvgClosingPrice NYSE.csv ModEnd/q1
WARNING: Use "yarn jar" to launch YARN applications.
23/06/10 05:52:12 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
23/06/10 05:52:13 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
23/06/10 05:52:13 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /user/bigdatalab456422/.staging/job_1685754149182_3594
23/06/10 05:52:14 INFO input.FileInputFormat: Total input files to process : 1
23/06/10 05:52:15 INFO mapreduce.JobSubmitter: number of splits:1
23/06/10 05:52:15 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
23/06/10 05:52:16 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1685754149182_3594
23/06/10 05:52:16 INFO mapreduce.JobSubmitter: Executing with tokens: []
23/06/10 05:52:16 INFO conf.Configuration: resource-types.xml not found
23/06/10 05:52:16 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
23/06/10 05:52:16 INFO impl.YarnClientImpl: Submitted application application_1685754149182_3594
23/06/10 05:52:16 INFO mapreduce.Job: The url to track the job: http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1685754149182_3594/
23/06/10 05:52:16 INFO mapreduce.Job: Running job: job_1685754149182_3594
23/06/10 05:52:33 INFO mapreduce.Job: Job job_1685754149182_3594 running in uber mode : false
23/06/10 05:52:33 INFO mapreduce.Job: map 0% reduce 0%
23/06/10 05:53:39 INFO mapreduce.Job: map 100% reduce 0%
23/06/10 05:54:11 INFO mapreduce.Job: map 100% reduce 100%
23/06/10 05:54:13 INFO mapreduce.Job: Job job_1685754149182_3594 completed successfully
23/06/10 05:54:13 INFO mapreduce.Job: Counters: 54
  File System Counters
    FILE: Number of bytes read=2782149
    FILE: Number of bytes written=6010257
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=40990977
    HDFS: Number of bytes written=4564
    HDFS: Number of read operations=0
    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)=18385
    Total time spent by all reduces in occupied slots (ms)=30058
    Total time spent by all map tasks (ms)=18385
    Total time spent by all reduce tasks (ms)=30058
    Total vcore-milliseconds taken by all map tasks=18385
    Total vcore-milliseconds taken by all reduce tasks=30058
    Total megabyte-milliseconds taken by all map tasks=18826240
    Total megabyte-milliseconds taken by all reduce tasks=30779392
  Map-Reduce Framework
    Map input records=735026
    Map output records=735026
    Map output bytes=8781587
    Map output materialized bytes=2782145
    Input split bytes=115
    Combine input records=0
    Combine output records=0
    Reduce input groups=203
    Reduce shuffle bytes=2782145
    Reduce input records=735026
    Reduce output records=203
    Spilled Records=1470052
    Shuffled Maps =1
    Failed Shuffles=0
    Merged Map outputs=1
    GC time elapsed (ms)=895
    CPU time spent (ms)=7780
    Physical memory (bytes) snapshot=896126976
    Virtual memory (bytes) snapshot=5186215936
    Total committed heap usage (bytes)=1061158912
    Peak Map Physical memory (bytes)=636690432
    Peak Map Virtual memory (bytes)=2587029504
    Peak Reduce Physical memory (bytes)=259436544
    Peak Reduce Virtual memory (bytes)=2599186432
  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=40990962
  File Output Format Counters
    Bytes Written=4564
[bigdatalab456422@ip-10-1-1-204 ~]$
```

Screenshots for Result in Hue:

Result Path: /user/bigdatalab456422/ModEnd/q1/part-r-00000

Home

/ user / bigdatalab456422 / ModEnd / q1

Trash

<input type="checkbox"/>	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	f		bigdatalab456422	bigdatalab456422	drwxr-xr-x	June 09, 2023 10:52 PM
<input type="checkbox"/>	.		bigdatalab456422	bigdatalab456422	drwxr-xr-x	June 09, 2023 10:54 PM
<input type="checkbox"/>	_SUCCESS	0 bytes	bigdatalab456422	bigdatalab456422	-rw-r--	June 09, 2023 10:54 PM
<input type="checkbox"/>	part-r-00000	4.5 KB	bigdatalab456422	bigdatalab456422	-rw-r--	June 09, 2023 10:54 PM

Show 45 of 2 items

Page 1 of 1

Back

Home

Page 1 to 2 of 2

Edit file

Refresh

View as binary

Download

Last modified
06/10/2023 11:24 AM
+05:30

User
bigdatalab456422

Group
bigdatalab456422

Size
4.46 KB

Mode
100644

/ user / bigdatalab456422 / ModEnd / q1 / **part-r-00000**

AA	51.91159715913803
AAI	10.268777015001355
AAN	19.548878615457568
AAP	44.167920310981586
AAR	18.982074468005067
AAV	12.323024390243903
AB	30.259244767970902
ABA	25.813885209713032
ABB	12.447982090589811
ABC	47.1888748995449
ABD	15.45613132209405
ABG	15.147817540322613
ABK	50.77681332763779
ABM	24.313192677629377
ABR	18.152958620689645
ABT	47.68227111636185
ABV	31.5187637028015
ABVT	48.865052631579005
ABX	22.349152784388313
ACC	25.148551774076783
ACE	42.218229755178974
ACF	16.171186174391213

Q2 - Hive

```
[bigdatalab456422@ip-10-1-1-204 ~]$ hive
```

```
[bigdatalab456422@ip-10-1-1-204 ~]$ hive
WARNING: Use "yarn jar" to launch YARN applications.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/jars/log4j-slf4j-impl-2.8.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/jars/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
2023-06-10 06:12:45,524 main WARN JNDI lookup class is not available because this JRE does not support JNDI. JNDI string lookups will not be available, continuing configuration. Ignoring java.lang.ClassNotFoundException: org.apache.logging.log4j.core.lookup.JndiLookup

Logging initialized using configuration in jar:file:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/jars/hive-common-2.1.1-cdh6.2.1.jar!/hive-log4j2.properties As
ync: false

WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
```

```
hive> set hive.cli.print.current.db = true ;
```

```
hive> set hive.cli.print.current.db = true ;
hive (default)>
```

```
hive (default)> CREATE DATABASE modend68 ;
```

```
hive (default)> CREATE DATABASE modend68 ;
OK
Time taken: 1.667 seconds
hive (default)>
```

```
hive (default)> USE modend68 ;
```

```
hive (default)> USE modend68 ;
OK
Time taken: 0.177 seconds
hive (modend68)>
```

```
hive (modend68)> SHOW TABLES ;
```

```
hive (modend68)> SHOW TABLES ;
OK
Time taken: 0.174 seconds
hive (modend68)>
```

```
hive (modend68)> CREATE TABLE airport(
    airport_id INT, name STRING,
    city STRING, country STRING,
    iata_code STRING, icao_code STRING,
    longitude DOUBLE, latitude DOUBLE,
    altitude INT,
    timezone DOUBLE, daylight STRING, tzregion STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED
AS TEXTFILE;
```

```
hive (modend68)> CREATE TABLE airport(
>   airport_id INT, name STRING,
>   city STRING, country STRING,
>   iata_code STRING, icao_code STRING,
>   longitude DOUBLE, latitude DOUBLE,
>   altitude INT,
>   timezone DOUBLE, daylight STRING, tzregion STRING)
> ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 0.414 seconds
```

```
hive (modend68)> LOAD DATA LOCAL INPATH 'airports_mod.dat' OVERWRITE
INTO TABLE airport;
```

```
hive (modend68)> LOAD DATA LOCAL INPATH 'airports_mod.dat' OVERWRITE INTO TABLE airport;
Loading data to table modend68.airport
OK
Time taken: 1.102 seconds
hive (modend68)>
```

```
hive (modend68)> CREATE TABLE airlines(
    airline_id INT,  airline_name STRING, alias STRING,
    airline_iata  STRING, airline_icao  STRING,
    callsign STRING,
    reg_country STRING, active_status  STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS
TEXTFILE;
```

```
hive (modend68)> CREATE TABLE airlines(
>
> airline_id INT,  airline_name STRING, alias STRING,
>
> airline_iata  STRING, airline_icao  STRING,
>
> callsign STRING,
>
> reg_country STRING, active_status  STRING)
> ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 0.115 seconds
hive (modend68)>
```

```
hive (modend68)> LOAD DATA LOCAL INPATH 'Final_airlines' OVERWRITE
INTO TABLE airlines;
```

```
hive (modend68)> LOAD DATA LOCAL INPATH 'Final_airlines' OVERWRITE INTO TABLE airlines;
Loading data to table modend68.airlines
OK
Time taken: 0.792 seconds
hive (modend68)>
```

```
hive (modend68)> CREATE TABLE routes(
    airline_iata STRING, airline_id INT,
    src_airport_iata STRING, src_airport_id INT,
    dest_airport_iata STRING, dest_airport_id INT,
    codeshare STRING, stops INT, equipment STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS
TEXTFILE;
```

```
hive (modend68)> CREATE TABLE routes(
>
> airline_iata STRING, airline_id INT,
>
> src_airport_iata STRING, src_airport_id INT,
>
> dest_airport_iata STRING, dest_airport_id INT,
>
> codeshare STRING, stops INT, equipment STRING)
> ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 0.099 seconds
hive (modend68)>
```

```
hive (modend68)> LOAD DATA LOCAL INPATH 'routes.dat' OVERWRITE INTO
TABLE routes;
```

```
hive (modend68)> LOAD DATA LOCAL INPATH 'routes.dat' OVERWRITE INTO TABLE routes;
Loading data to table modend68.routes
OK
Time taken: 0.732 seconds
hive (modend68)>
```

Hive Q1. Which airports have the highest altitude ?

Soln →

```
hive (modend68)> SELECT country, max(altitude) FROM airport GROUP BY country;
```

```
hive (modend68)> SELECT country, max(altitude) FROM airport GROUP BY country;Query ID = bigdatalab456422_20230610070518_e2d575f9-8085-4dda-95af-98fbc3629812
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
23/06/10 07:05:19 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
23/06/10 07:05:19 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
Starting Job = job_1685754149182_4032, Tracking URL = http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1685754149182_4032/
Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job -kill job_1685754149182_4032
```

Hive Q2. How many routes are operated by active airlines from the United States ?

Soln →

```
hive (modend68)> SELECT DISTINCT(airline_name) FROM airlines al
JOIN routes r ON al.airline_id = r.airline_id
JOIN airport ap1 ON al.airport_id = r.src_airport_id
WHERE TRIM(UPPER(ap1.country)) = 'UNITED STATES';
```

```
hive (modend68)> SELECT DISTINCT(airline_name) FROM airlines al
>
> JOIN routes r ON al.airline_id = r.airline_id
>
> JOIN airport ap1 ON ap1.airport_id = r.src_airport_id
>
> WHERE TRIM(UPPER(ap1.country)) = 'UNITED STATES';
Query ID = bigdatalab456422_20230610070606_5b51c9b0-2701-4479-be7d-977f9b457721
Total jobs = 1
SLF4J: Class path contains multiple SLF4J bindings.SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/jars/log4j-slf4j-impl-2.8.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]

MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
23/06/10 07:06:16 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
23/06/10 07:06:16 INFO client.RMProxy: Connecting to ResourceManager at ip-10-1-1-204.ap-south-1.compute.internal/10.1.1.204:8032
Starting Job = job_1685754149182_4035, Tracking URL = http://ip-10-1-1-204.ap-south-1.compute.internal:6066/proxy/application_1685754149182_4035/
Kill Command = /opt/cloudera/parcels/CDH-6.2.1-1.cdh6.2.1.p0.1425774/lib/hadoop/bin/hadoop job -kill job_1685754149182_4035
```

Hive Q3. Which airlines operate routes that have less than 3 stops number of stops top 10 alphabetically?

Soln →

```
hive (modend68)> SELECT DISTINCT(airline_name) FROM airlines al
JOIN routes r ON al.airline_id = r.airline_id
WHERE r.stops < 3 LIMIT 10;
```

Hive Q4. How many airlines have a specific IATA code 'W9'?

Soln →

```
hive (modend68)> SELECT count(airline_id) FROM airlines WHERE
airline_iata = 'W9';
```

Hive Q5. Find the airlines that operate routes with a specific equipment as 'AN4' and codeshare enabled

Soln →

```
hive (modend68)> SELECT DISTINCT(airline_name) FROM airlines al  
JOIN routes r ON al.airline_id = r.airline_id  
WHERE equipment = 'AN4' AND TRIM(UPPER(codeshare)) = 'Y';
```


Q3 - PySpark

```
[bigdatalab456422@ip-10-1-1-204 ~]$ pyspark
```

```
from pyspark.sql.types import StructType, StringType, IntegerType,
LongType, DoubleType

airline_schema = StructType().add("Year", IntegerType(),
True).add("Quarter", IntegerType(), True).add("ARPS", DoubleType(),
True).add("Booked_seats", LongType(), True)
print(airline_schema)

df1 = spark.read.format("csv").option("header",
"False").schema(airline_schema).load("hdfs://nameservice1/user/bigdat
alab456422/airlines.csv")

df1.printSchema();
df1.count();
df1.show();

df1.registerTempTable("airline_table")
```

PySpark Q1. What is the total revenue generated in each year?

Soln →

```
yearly_revenue_mn = spark.sql("SELECT Year, ROUND(sum(ARPS *
Booked_seats)/1000000, 2) FROM airline_table GROUP BY Year")

yearly_revenue_mn.show()
```

PySpark Q2. Which year had the highest average revenue per seat?

Soln →

```
highest_rev_mn_year = spark.sql("SELECT max(total_rev) FROM (SELECT
sum(ARPS * Booked_seats)/1000000 AS total_rev FROM airline_table
GROUP BY Year) AS tbl1")

highest_rev_mn_year.show()
```

PySpark Q3. What is the total number of booked seats for each quarter in a given year?

Soln →

```
total_seats = spark.sql("SELECT Quarter, Booked_seats FROM
airline_table GROUP BY Quarter")
```

```
total_seats.show()
```