Big Data Report

Project Description

I choose the dataset of Airlines in 2008. It is a large that contains some information about year, month, departure time, arrive time, delay time and so on.

I have used the Hadoop, Hbase, Hive, Pig do analysis it.

I will illustrate in the following part:

1. Hadoop:

A: HDFS

Firstly, I used Hadoop Shell to make a directory called Final:

```
Zhuowens-MacBook-Pro:bin mumu$ ./hadoop fs -mkdir /Final
```

Then, I used Hadoop Shell (-copyFromLocal) to upload the Airline Dataset to HDFS:

```
Zhuowens-MacBook-Pro:bin mumu$ ./hadoop fs -put /Users/mumu/Downloads/2008.csv /Final/
15/12/13 12:37:21 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platfo
es where applicable
```

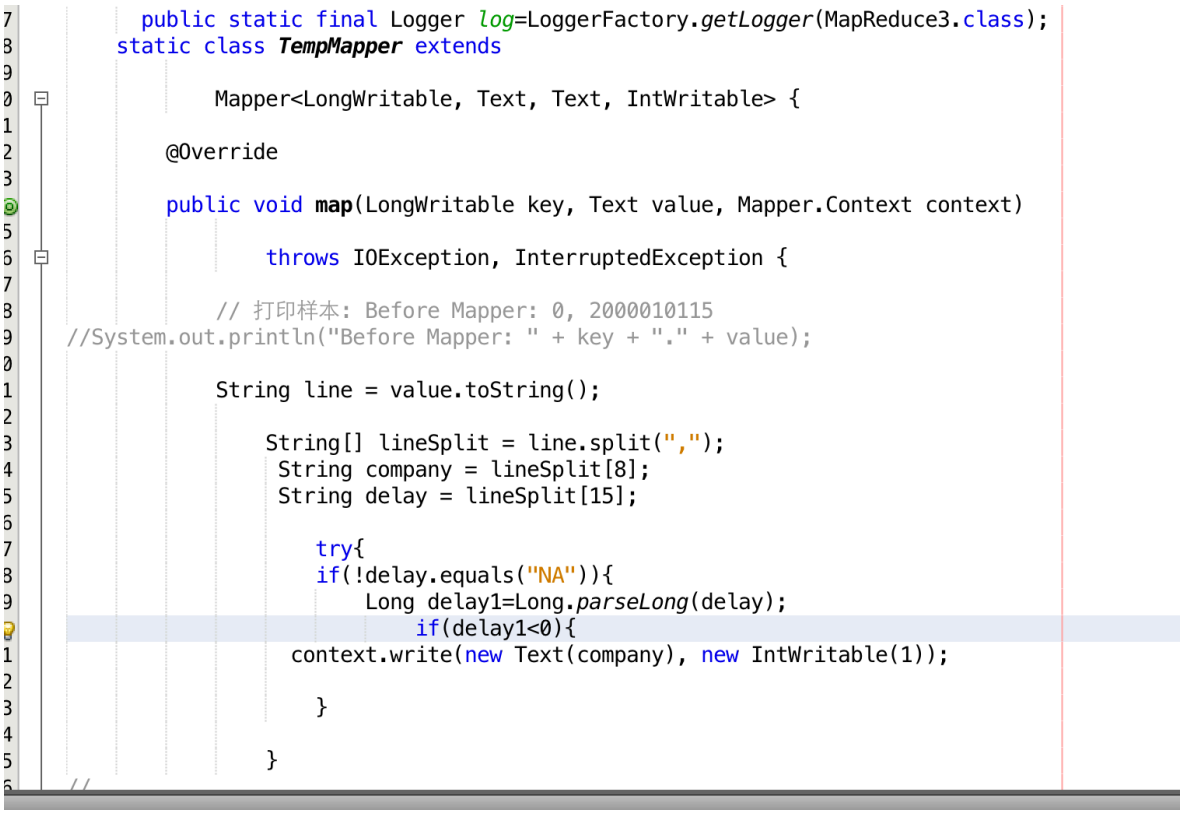
2. MapReduce

I used Java in Netbeans to do mapreduce to implement following functons.

A: For each uniqueCarrier, How many flights are not delay, flight arrived in time.

The screenshots of MapReduce in Java.

Map Function:



```
        public static final Logger log=LoggerFactory.getLogger(MapReduce3.class);
    static class TempMapper extends

            Mapper<LongWritable, Text, Text, IntWritable> {

        @Override

        public void map(LongWritable key, Text value, Mapper.Context context)

                throws IOException, InterruptedException {

            // 打印样本: Before Mapper: 0, 2000010115
//System.out.println("Before Mapper: " + key + "." + value);

            String line = value.toString();

                String[] lineSplit = line.split(",");
                 String company = lineSplit[8];
                 String delay = lineSplit[15];

                    try{
                    if(!delay.equals("NA")){
                        Long delay1=Long.parseLong(delay);
                            if(delay1<0){
                  context.write(new Text(company), new IntWritable(1));

                    }

                }
//
```

And the Reduce Function:

```
    }
static class TempReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

    @Override

        public void reduce(Text key, Iterable<IntWritable> values,

                Context context) throws IOException, InterruptedException {

           // System.out.println("Before Reduce:" + key + ",");
                int count = 0;
                for (IntWritable v : values) {
                    count = count + v.get();
                }
                try {
                    context.write(key, new IntWritable(count));
                    System.out.println(
                            ""
                             + "After Reduce:" + key + "," + count);

                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }



        }
```

The last stop, I read this file from HDFS, and then upload the result into HDFS too.

```
public static void main(String[] args) throws Exception {

        String dst = "hdfs://localhost:9000/Final/2008.csv";

String outFiles = "/Users/mumu/NetBeansProjects/FinalHBase/src/output";
     // String dstOut = "hdfs://localhost:9000/Final/Mapreduce3";

        Configuration hadoopConfig = new Configuration();

        hadoopConfig.set("fs.hdfs.impl",

            org.apache.hadoop.hdfs.DistributedFileSystem.class.getName()

        );

        hadoopConfig.set("fs.file.impl",

            org.apache.hadoop.fs.LocalFileSystem.class.getName()

        );

        Job job = new Job(hadoopConfig);

        FileInputFormat.addInputPath(job, new Path(dst));

        FileOutputFormat.setOutputPath(job, new Path(outFiles));
       job.setMapperClass(TempMapper.class);

        job.setReducerClass(TempReducer.class);

        job.setOutputKeyClass(Text.class);
```

Then use Hadoop Shell (-cat) to check if the result is upload and is successful.

```
AA      23855
AQ      2890
AS      6779
B6      9998
CO      13774
DL      33117
EV      16544
F9      6723
FL      21998
HA      7544
MQ      38524
NW      34035
OH      11158
OO      49786
UA      31388
US      41977
WN      63129
XE      35225
YV      19127
Zhuowens-MacBook-Pro:bin mumu$
```

B. Using Mapreduce chain job to calculate the top 10 total numbers of flights each carriers take.

Map1 Function

```
    static class TempMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
        @Override

        public void map(LongWritable key, Text value, Mapper.Context context)

                throws IOException, InterruptedException {

            // 打印样本: Before Mapper: 0, 2000010115
//System.out.println("Before Mapper: " + key + "." + value);

            String line = value.toString();
            try {
                String[] lineSplit = line.split(",");
                 String company = lineSplit[8];
               // requestUrl = requestUrl.substring(0,requestUrl.indexOf(' ')+1);
                // Text out = new Text(requestUrl);
                System.out.println(
                        ""
                        + "After Mapper:"+ new Text(company) + "," + new IntWritable(1));
context.write(new Text(company), new IntWritable(1));
            } // context.write(out,one);
            catch (java.lang.ArrayIndexOutOfBoundsException e) {
                //  context.getCounter(Counter.LINESKIP).increment(1);

            }



        }
```

Reduce1 Function

```
    static class TempReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

      @Override

       public void reduce(Text key, Iterable<IntWritable> values,

               Context context) throws IOException, InterruptedException {

          // System.out.println("Before Reduce:" + key + ",");
               int count = 0;
               for (IntWritable v : values) {
                   count = count + v.get();
               }
               try {
                   context.write(key, new IntWritable(count));
                   System.out.println(
                           ""
                            + "After Reduce:" + key + "," + count);

               } catch (InterruptedException e) {
                   e.printStackTrace();
               }
           }

    }
```

Map2 Function:

```
    static class TempMapper2 extends Mapper<LongWritable, Text, IntWritable, Text> {

        public void map(LongWritable key, Text value, Mapper.Context context)
                throws IOException, InterruptedException {
            String line = value.toString();
            try {
                String[] lineSplit = line.split("\t");
                // String requestUrl = line.substring(0, 10);
                String requestUrl = lineSplit[0];
                int val = Integer.parseInt(lineSplit[1]);

                // requestUrl = requestUrl.substring(0,requestUrl.indexOf(' ')+1);
                // Text out = new Text(requestUrl);
                System.out.println("" + "After Mapper2:" + line + ",    |" + val);
                context.write(new IntWritable(val), new Text(requestUrl));
            } // context.write(out,one);
            catch (java.lang.ArrayIndexOutOfBoundsException e) {
                //  context.getCounter(Counter.LINESKIP).increment(1);
            }
        }
    }
```

Reduce2 Fucntion: This function implement sort

```
    static class TempReduce2 extends Reducer<IntWritable, Text, Text, IntWritable> {
        LinkedHashMap<String, Integer> tm = new LinkedHashMap<String, Integer>();
        int count = 0;
       @Override
        public void reduce(IntWritable key, Iterable<Text> values, Context context)
                throws IOException, InterruptedException {
            for (Text val : values) {
                count++;
                String a = val.toString();
                int b = key.get();
                tm.put(a, b);
                System.out.println("" + "reduce2:" + a + " || " + tm.get(val));
                // context.write(result, key);
            }


        }

        @Override
                protected void cleanup(Reducer.Context context) throws IOException,
                                    InterruptedException {
                    ArrayList<LinkedHashMap<String,Integer>> a = new ArrayList();
          int i=0;
           for (Map.Entry<String,Integer> entry : tm.entrySet()) {
                i++;
                if(i>count-5){
                      LinkedHashMap<String,Integer> t = new LinkedHashMap<String,Integer>();
                      t.put(entry.getKey(), entry.getValue());
                     a.add(t);
                }

            }
            for(int i=a.size()-1;i>=0;i--)
```

```
            }
            for(int j=a.size()-1;j>=0;j--)

            {
                    Text result = new Text();

                    for (Map.Entry<String,Integer> entry : a.get(j).entrySet()) {
                            result.set(entry.getKey());
                        context.write(result,new IntWritable(entry.getValue()));
                    }

            }
        }


    public static void main(String[] args) throws Exception {
```

I read the dataset from the HDFS, then upload the result to the HDFS.

This time have two jobs so there are two jobs configuration.

```
public static void main(String[] args) throws Exception {

    //输入路径
   String dst = "hdfs://localhost:9000/Final/2008.csv";

    //输出路径，必须是不存在的，空文件加也不行。
    String dstOut = "hdfs://localhost:9000/Final/MaprReduceChainJob2";
   String outFiles = "/Users/mumu/NetBeansProjects/FinalHBase/src/output";
    Configuration hadoopConfig = new Configuration();

    hadoopConfig.set("fs.hdfs.impl",org.apache.hadoop.hdfs.DistributedFileSystem.class.getName());

    hadoopConfig.set("fs.file.impl",org.apache.hadoop.fs.LocalFileSystem.class.getName());

    Job job = new Job(hadoopConfig);
    Job job2 = new Job(hadoopConfig);
    //如果需要打成jar运行，需要下面这句
    //job.setJarByClass(NewMaxTemperature.class);
    //job执行作业时输入和输出文件的路径
    FileInputFormat.addInputPath(job, new Path(dst));
    FileOutputFormat.setOutputPath(job, new Path(dstOut));
    FileInputFormat.addInputPath(job2, new Path(dstOut));
  FileOutputFormat.setOutputPath(job2, new Path(outFiles));

    JobConf map1Conf = new JobConf(false);
    ChainMapper.addMapper(job,TempMapper.class,LongWritable.class,Text.class,Text.class,IntWritable.cla
    JobConf reduceConf = new JobConf(false);
     ChainReducer.setReducer(job,TempReducer.class,Text.class,IntWritable.class,Text.class,IntWritable.

     JobConf map2Conf = new JobConf(false);
    ChainMapper.addMapper(job2,TempMapper2.class,Text.class,IntWritable.class,IntWritable.class,Text.c
   JobConf map3Conf = new JobConf(false);
```

```
        ChainReducer.setReducer(job,TempReducer.class,Text.class,IntWritable.class,Text.class,IntWritable

        JobConf map2Conf = new JobConf(false);
       ChainMapper.addMapper(job2,TempMapper2.class,Text.class,IntWritable.class,IntWritable.class,Text.c
      JobConf map3Conf = new JobConf(false);
       ChainReducer.setReducer(job2,TempReduce2.class,IntWritable.class,Text.class,Text.class,IntWritable
       //  JobClient.runJob(job);

       //指定自定义的Mapper和Reducer作为两个阶段的任务处理类
         job.setMapperClass(TempMapper.class);

         job.setReducerClass(TempReducer.class);
       //设置最后输出结果的Key和Value的类型
       job.setOutputKeyClass(Text.class);

       job.setOutputValueClass(IntWritable.class);

       job2.setOutputKeyClass(Text.class);

       job2.setOutputValueClass(IntWritable.class);

//  job2.setSortComparatorClass(LongWritable.DecreasingComparator.class);
       //执行job，直到完成
       job.waitForCompletion(true);
       System.out.println("Finished1");
       job2.waitForCompletion(true);
       System.out.println("Finished2");

   }
```

Then use Hadoop Shell (-cat) to check if the result is upload and is successful.

```
WN      195880
QQ      95132
MQ      83675
US      76041
UA      74790
```

2. Hbase---→Hive

I used Java to implement Hbase. Created a table named “FinaRoutes3”

and also created a column family named “Routes_info” that contains 29 columns named Year, Month, DOM, DepTim, CRSDepTim, ArrTime, CRSArrTime, UniqueCarrier, FlightNum, TailNum, ActualElapsed, CRSElapsedTime, Airtime, ArrDelay, DepDelay, Original, Dest, disntance, Taxiin, Taxiout, …

Then I loaded dataset from local:

```
    */
    public static void main(String[] args) throws IOException {

        Configuration con = HBaseConfiguration.create();

     // Instantiating HbaseAdmin class
     HBaseAdmin admin = new HBaseAdmin(con);

     // Instantiating table descriptor class
     HTableDescriptor tableDescriptor = new HTableDescriptor(TableName.valueOf("FinaRoutes3"));

     // Adding column families to table descriptor
     tableDescriptor.addFamily(new HColumnDescriptor("Routes_info"));
    //  tableDescriptor.addFamily(new HColumnDescriptor("rate"));

     // Execute the table through admin
     admin.createTable(tableDescriptor);
     System.out.println(" Table created ");


     // Instantiating Configuration class
     Configuration config = HBaseConfiguration.create();

     // Instantiating HTable class
     HTable hTable = new HTable(config, "FinaRoutes3");

     FileReader file=new FileReader("/Users/mumu/Downloads/2008.csv");
     BufferedReader b=new BufferedReader(file);
     int count = 0;
```

```
while((test = b.readLine())!=null){
    count++;
}
// Instantiating Put class
// accepts a row name.
FileReader fr=new FileReader("/Users/mumu/Downloads/2008.csv");
BufferedReader br=new BufferedReader(fr);
for(int i = 1;i<=count;i++){
    Put p = new Put(Bytes.toBytes("row"+i));


// adding values using add() method
// accepts column family name, qualifier/row name ,value

String s=null;

if((s=br.readLine())!=null){
    String[] insert = s.split(",");

    p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("Year"),Bytes.toBytes(insert[0]));

    p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("Month"),Bytes.toBytes(insert[1]));

    p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("DOM"),Bytes.toBytes(insert[2]));

   p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("DOW"),Bytes.toBytes(insert[3]));
    p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("DepTim"),Bytes.toBytes(insert[4]));
     p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("CRSDepTim"),Bytes.toBytes(insert[5]));
      p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("ArrTime"),Bytes.toBytes(insert[6]));
       p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("CRSArrTime"),Bytes.toBytes(insert[7]));
        p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("UniqueCarrier"),Bytes.toBytes(insert[8
         p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("FlightNum"),Bytes.toBytes(insert[9]))
         p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("ActualElapsedTime"),Bytes.toBytes(ins
p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("CRSElapsedTime"),Bytes.toBytes(insert[12]));
 p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("Airtime"),Bytes.toBytes(insert[13]));
  p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("ArrDelay"),Bytes.toBytes(insert[14]));
   p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("DepDelay"),Bytes.toBytes(insert[15]));
    p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("Origin"),Bytes.toBytes(insert[16]));
     p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("Dest"),Bytes.toBytes(insert[17]));
      p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("disntance"),Bytes.toBytes(insert[18]));
       p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("Taxiin"),Bytes.toBytes(insert[19]));
        p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("Taxiout"),Bytes.toBytes(insert[20]));
         p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("Cancelled"),Bytes.toBytes(insert[21]))
          p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("CancellationCode"),Bytes.toBytes(inse
           p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("Diverted"),Bytes.toBytes(insert[23])
    // Saving the put Instance to the HTable.
             p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("CarrierDelay"),Bytes.toBytes(insert
              p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("WeatherDelay"),Bytes.toBytes(inser
               p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("NASDelay"),Bytes.toBytes(insert[2
                p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("SecurityDelay"),Bytes.toBytes(ir
              //   p.add(Bytes.toBytes("Routes_info"),Bytes.toBytes("LateAircraftDelay"),Bytes.toB
    hTable.put(p);
}
}

// closing HTable
hTable.close();

// Instantiating Configuration class
```

Then go to the terminal to check if the table named “FinaRoutes3” and data are loaded into the Hbase.

Because the dataset is too large, I used search function (list&get)to check them.

```
hbase(main):006:0> list
TABLE
Airports
FinaRoutes
FinaRoutes2
FinaRoutes3
Route
hi
test
tt
8 row(s) in 0.0400 seconds

=> ["Airports", "FinaRoutes", "FinaRoutes2", "FinaRoutes3", "Route", "hi", "test", "tt"]
hbase(main):007:0> get 'FinaRoutes3', 'row1'
COLUMN                      CELL
 Routes_info:ActualElapsedT timestamp=1449959007537, value=128
 ime
 Routes_info:Airtime        timestamp=1449959007537, value=116
 Routes_info:ArrDelay       timestamp=1449959007537, value=-14
 Routes_info:ArrTime        timestamp=1449959007537, value=2211
 Routes_info:CRSArrTime     timestamp=1449959007537, value=2225
 Routes_info:CRSDepTim      timestamp=1449959007537, value=1955
 Routes_info:CRSElapsedTime timestamp=1449959007537, value=150
 Routes_info:CancellationCo timestamp=1449959007537, value=0
 de
 Routes_info:Cancelled      timestamp=1449959007537, value=0
 Routes_info:CarrierDelay   timestamp=1449959007537, value=NA
 Routes_info:DOM            timestamp=1449959007537, value=3
 Routes_info:DOW            timestamp=1449959007537, value=4
 Routes_info:DepDelay       timestamp=1449959007537, value=8
 Routes_info:DepTim         timestamp=1449959007537, value=2003
 Routes_info:Dest           timestamp=1449959007537, value=TPA
 Routes_info:Diverted       timestamp=1449959007537, value=NA
 Routes_info:FlightNum      timestamp=1449959007537, value=335
 Routes_info:Month          timestamp=1449959007537, value=1
 Routes_info:NASDelay       timestamp=1449959007537, value=NA
 Routes_info:Origin         timestamp=1449959007537, value=IAD
 Routes_info:SecurityDelay  timestamp=1449959007537, value=NA
 Routes_info:TailNum        timestamp=1449959007537, value=N712SW
 Routes_info:Taxiin         timestamp=1449959007537, value=4
 Routes_info:Taxiout        timestamp=1449959007537, value=8
 Routes_info:UniqueCarrier  timestamp=1449959007537, value=WN
 Routes_info:WeatherDelay   timestamp=1449959007537, value=NA
 Routes_info:Year           timestamp=1449959007537, value=2008
 Routes_info:disntance      timestamp=1449959007537, value=810
28 row(s) in 0.0820 seconds
```

Hive can use SQL to select join, update and so on to manage the data. So I need to mapping the hbase column to create a table in Hive. I use shell to implement it.

```
hive> CREATE EXTERNAL TABLE Routes2(key string, ActualElapsedTime string, AirTime string, ArrDelay
 string, ArrTime string, CRSArrTime string, CRSDepTim string,CRSElaosedTime string,CRSElapsedTime
string, Cancelled string, CarrierDelay string, DOM string,DOW string, DepDelay string, DepTime str
ing, Dest string, Diverted string, FlightNum string, Month string, NASDelay string, Origin string,
 Security string, TailNum string, Taxiin string, Taxiout string, UniqueCarrier string, WeatherDela
y string, Year string,disntance string)
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
                                                    > WITH SERDEPROPERTIES("hbase.columns.mapping
" = "Routes_info:ActualElapsedTime,Routes_info:Airtime,Routes_info:ArrDelay,Routes_info:ArrTime, R
outes_info:CRSArrTime,Routes_info:CRSDepTim,Routes_info:CRSElapsedTime,Routes_info:CancellationCod
e,Routes_info:Cancelled,Routes_info:CarrierDelay,Routes_info:DOM,Routes_info:DOW,Routes_info:DepDe
lay,Routes_info:DepTim,Routes_info:Dest,Routes_info:Diverted,Routes_info:FlightNum,Routes_info:Mon
th,Routes_info:NASDelay,Routes_info:Origin,Routes_info:SecurityDelay,Routes_info:TailNum,Routes_in
fo:Taxiin,Routes_info:Taxiout,Routes_info:UniqueCarrier,Routes_info:WeatherDelay,Routes_info:Year,
Routes_info:disntance")
    > TBLPROPERTIES("hbase.table.name" = "FinaRoutes3");
                                                OK

Time taken: 0.191 seconds
```

Using

`'org.apache.hadoop.hive.hbase.HBaseStorageHandler'`

to mapping the data to Hbase, TBLEROPERTIES to the table of hbase.

Then I use “Desc” to see the table’s column type;

```
                                    hive> desc Routes2;
                                                                                OK

key                     string                  from deserializer
actualelapsedtime       string                  from deserializer
airtime                 string                  from deserializer
arrdelay                string                  from deserializer
arrtime                 string                  from deserializer
crsarrtime              string                  from deserializer
crsdeptim               string                  from deserializer
crselaosedtime          string                  from deserializer
crselapsedtime          string                  from deserializer
cancelled               string                  from deserializer
carrierdelay            string                  from deserializer
dom                     string                  from deserializer
dow                     string                  from deserializer
depdelay                string                  from deserializer
deptime                 string                  from deserializer
dest                    string                  from deserializer
diverted                string                  from deserializer
flightnum               string                  from deserializer
month                   string                  from deserializer
nasdelay                string                  from deserializer
origin                  string                  from deserializer
security                string                  from deserializer
tailnum                 string                  from deserializer
taxiin                  string                  from deserializer
taxiout                 string                  from deserializer
uniquecarrier           string                  from deserializer
weatherdelay            string                  from deserializer
year                    string                  from deserializer
disntance               string                  from deserializer
```

Because the dataset is too large, So I also used select to check if it is correct.

```
hive> select * from Routes2
    > where UniqueCarrier='AA';
```

The result is as follows:

```
      NA   2493   1    NA   DFW   NA   N390AA  13   9    AA   NA   2008  1235
row529437  254  199   55   54  2359  2240  199   0    0    0    6    7    0    2240  LAX
      0    2493   1    0    DFW   0    N5DCAA  40   15   AA   55   2008  1235
row529438  193  175   14   13  2359  2240  199   0    0    NA   7    1    20   2300  LAX
      NA   2493   1    NA   DFW   NA   N5FJAA  10   8    AA   NA   2008  1235
row529439  203  180   -3  2356  2359  2240  199   0    0    NA   8    2    -7   2233  LAX
      NA   2493   1    NA   DFW   NA   N643AA  10   13   AA   NA   2008  1235
row529440  188  171  -11  2348  2359  2240  199   0    0    NA   9    3    0    2240  LAX
      NA   2493   1    NA   DFW   NA   N5CFAA  4    13   AA   NA   2008  1235
row529441  208  179   44   43  2359  2240  199   0    0    0    10   4    35   2315  LAX
      0    2493   1    0    DFW   35   N5DMAA  19   10   AA   9    2008  1235
row529442  197  170   -7  2352  2359  2240  199   0    0    NA   11   5    -5   2235  LAX
      NA   2493   1    NA   DFW   NA   N616AA  11   16   AA   NA   2008  1235
row529443  207  177   34   33  2359  2240  199   0    0    0    12   6    26   2306  LAX
      26   2493   1    0    DFW   0    N5EUAA  13   17   AA   8    2008  1235
row529444  191  165   26   25  2359  2240  199   0    0    0    13   7    34   2314  LAX
      0    2493   1    0    DFW   24   N5CVAA  9    17   AA   2    2008  1235
row529445   NA    NA   NA   NA  2359  2240  199   0    1    NA   14   1    NA   NA    LAX
      NA   2493   1    NA   DFW   NA        NA   NA   AA   NA   2008  1235
row529446   NA    NA   NA   NA  2359  2240  199   0    1    NA   15   2    NA   NA    LAX
      NA   2493   1    NA   DFW   NA        NA   NA   AA   NA   2008  1235
row529447  201  175   0   2359  2359  2240  199   0    0    NA   16   3    -2   2238  LAX
      NA   2493   1    NA   DFW   NA   N5BRAA  9    17   AA   NA   2008  1235
row529448  191  169   97  136  2359  2240  199   0    0    0    17   4    105  25    LAX
      20   2493   1    0    DFW   77   N5DMAA  7    15   AA   0    2008  1235
row529449  185  161   6    5   2359  2240  199   0    0    NA   18   5    20   2300  LAX
      NA   2493   1    NA   DFW   NA   N630AA  11   13   AA   NA   2008  1235
row529450   NA    NA   NA   NA  2359  2240  199   0    1    NA   19   6    NA   NA    LAX
      NA   2493   1    NA   DFW   NA        NA   NA   AA   NA   2008  1235
row529451  195  174   -8  2351  2359  2240  199   0    0    NA   20   7    -4   2236  LAX
      NA   2493   1    NA   DFW   NA   N5DBAA  5    16   AA   NA   2008  1235
row529452  203  178   3    2   2359  2240  199   0    0    NA   21   1    -1   2239  LAX
      NA   2493   1    NA   DFW   NA   N5CXAA  14   11   AA   NA   2008  1235
row529453  218  185   12   11  2359  2240  199   0    0    NA   22   2    -7   2233  LAX
      NA   2493   1    NA   DFW   NA   N5EHAA  16   17   AA   NA   2008  1235
row529454   NA    NA   NA   NA  2359  2240  199   0    1    NA   23   3    NA   NA    LAX
      NA   2493   1    NA   DFW   NA        NA   NA   AA   NA   2008  1235
row529455  212  189   63  102  2359  2240  199   0    0    0    24   4    50   2330  LAX
      44   2493   1    0    DFW   6    N626AA  12   11   AA   13   2008  1235
row529456  227  198   25   24  2359  2240  199   0    0    0    25   5    -3   2237  LAX
```

This result is too large and also has so many information.

Next, I change my idea, Hive is related to the HDFS, So I analysis the results that I did mapreduce before.

I used hive shell to create table. Using syntax ‘LOAD’ to load data into table I created.

```
hive> CREATE EXTERNAL TABLE Mapreduce3 (Carrier String, Number int)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION '/Mapreduce3 /Mapreduce3.csv';
OK
Time taken: 0.045 seconds
hive> show tables;
OK
a
b
c
mapreduce
mapreduce1
mapreduce3
routes
routes2
Time taken: 0.028 seconds, Fetched: 8 row(s)
```

Then load the data

```
hive> load data inpath '/Results' overwrite into table Mapreduce3;
Loading data to table default.mapreduce3
```

Then select all to check

```
hive> select * from Mapreduce3;
OK
9E      19469
AA      23855
AQ      2890
AS      6779
B6      9998
CO      13774
DL      33117
EV      16544
F9      6723
FL      21998
HA      7544
MQ      38524
NW      34035
OH      11158
OO      49786
UA      31388
US      41977
WN      63129
XE      35225
YV      19127
```

this result is be analysis in mapreduce I did before. The data is not large and useful I think. You can use such as select, join and so on to see what you like.

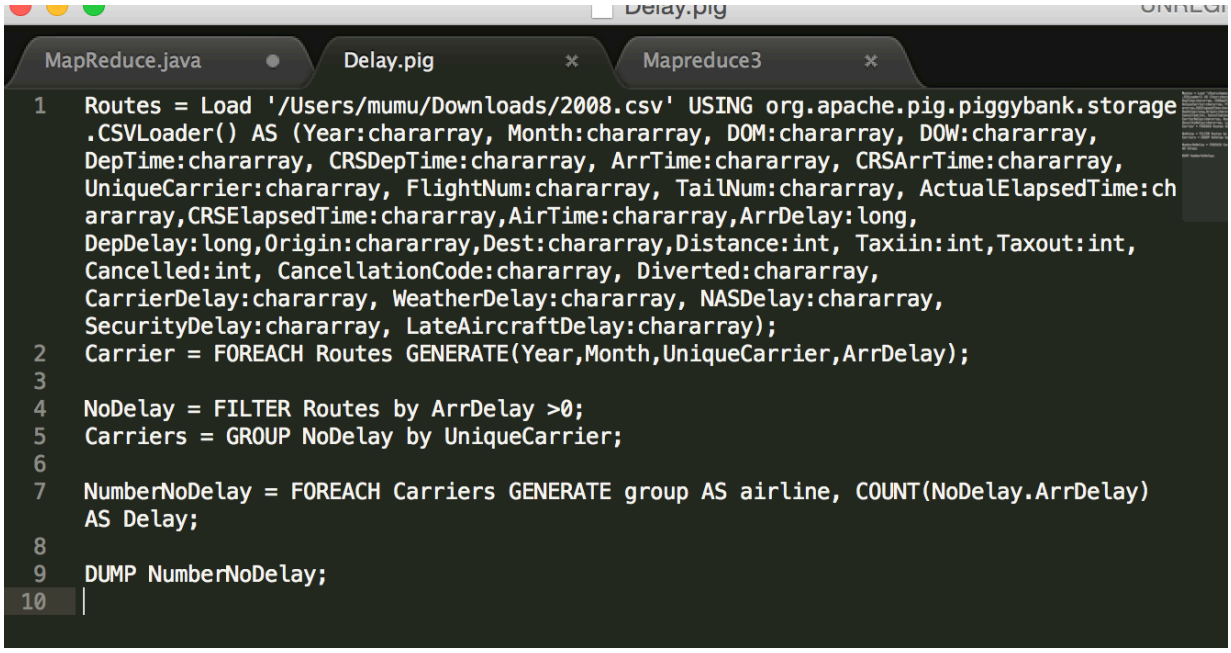
I want to see total flight numbers that a specific Carrier take. So I used select syntax .

```
hive> select * from Mapreduce3
    > where Carrier='AA';
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/Users/mumu/Downloads/hadoop-2.5.2/shar
taticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/Users/mumu/Downloads/apache-hive-0.13.
rBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an expla
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
15/12/15 21:33:51 WARN util.NativeCodeLoader: Unable to load native-hadoo
re applicable
15/12/15 21:33:51 WARN conf.Configuration: file:/tmp/mumu/hive_2015-12-15
n attempt to override final parameter: mapreduce.job.end-notification.max
15/12/15 21:33:51 WARN conf.Configuration: file:/tmp/mumu/hive_2015-12-15
n attempt to override final parameter: mapreduce.job.end-notification.max
15/12/15 21:33:51 WARN conf.HiveConf: DEPRECATED: Configuration property
rovide a valid value for hive.metastore.uris if you are connecting to a r
15/12/15 21:33:51 WARN conf.HiveConf: DEPRECATED: hive.metastore.ds.retry
tead
Execution log at: /var/folders/7y/gbtr3z3x6v52w4ct67hmlkg00000gn/T//mumu/
Job running in-process (local Hadoop)
Hadoop job information for null: number of mappers: 0; number of reducers
2015-12-15 21:33:54,640 null map = 100%,  reduce = 0%
Ended Job = job_local1233045640_0001
Execution completed successfully
MapredLocal task succeeded
OK
AA      23855
```

3. Pig (include UDF)

I wrote two pig script.

The first is to calculate how many flights that delay for each Carriers Company.

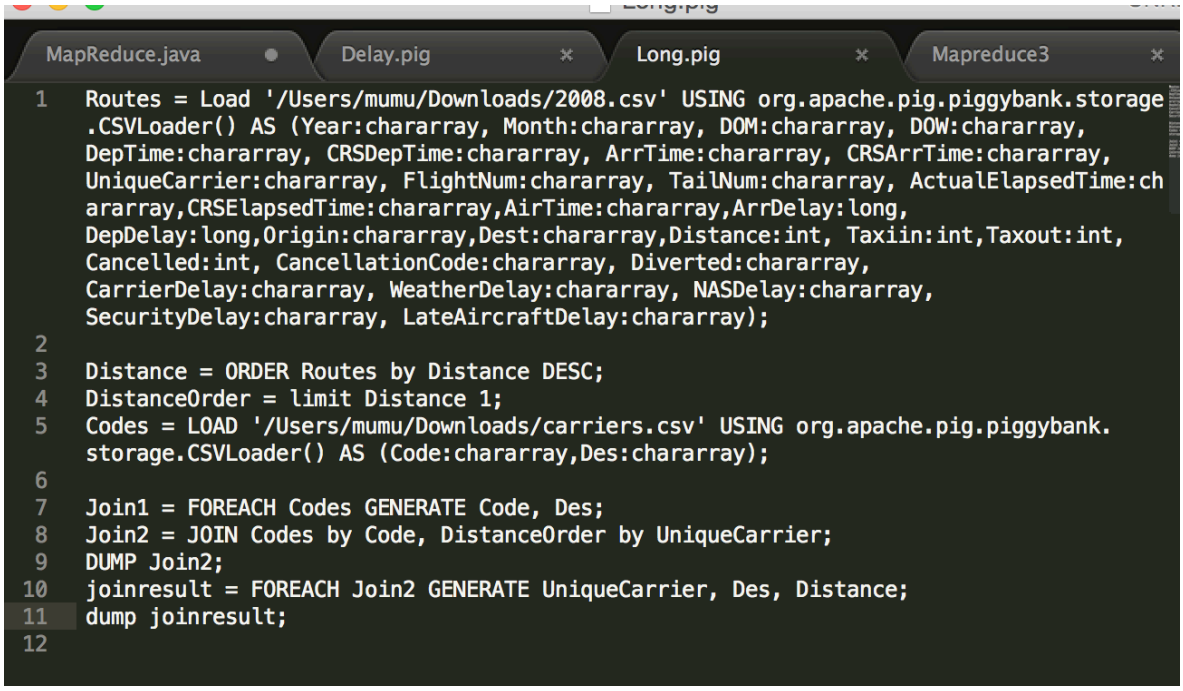


Then run in terminal. The results is as follows:

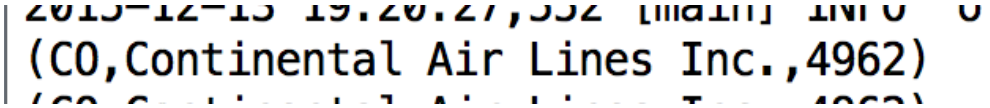
```
(AA,27112)
(AQ,963)
(AS,5662)
(B6,6603)
(CO,12630)
(DL,24282)
(EV,21069)
(F9,8006)
(FL,21934)
(HA,2193)
(MQ,40479)
(NW,33195)
(OH,18023)
(OO,46768)
(UA,39102)
(US,29484)
(WN,89390)
(XE,31837)
(YV,18500)
```

Second I download another dataset that contains the concrete description including the full name of each Airways companies.

I found the longest distance that one airways company hold and all output the full name and description of it.



And last step is to run in terminal and get the results.



I used User Define Function to realize the function that if the departure time more than zero, it will shows the string that “The flight is delay of departure ” else it will shows the string that ‘ The flight is not delay of departure’ I used Java to implement it.

```
public class UDF extends EvalFunc<String> {

    @Override
    public String exec(Tuple input) throws IOException {
        if (input==null||input.size()==0)
            return null;
        String result = " ";
        int delay=0;
        try{
             String depTime = (String) input.get(15);
             delay=Integer.parseInt(depTime);


        }catch(NumberFormatException ec){}
        if(delay<0){
            result= "The departure of flight is not delay";
        }
        else if(delay>0){
            result="The departure of flifght is delay";
        }


    return result;
}
}
```

And than complier it and run in terminal.

4.

This time, I tried to use AWS to configure the EC2 instance. And create master node, two slave nodes to create the Hadoop Cluster.

I use Linux Shell command to downloads resource from the website and also change JAVA_HOME HADOOP_HOME to export it the profile file.

I use manual configuration by myself not using EMR, During the process of completion, I have encounter some problems. When finish the configuration of hadoop, there is no enough space to configure hbase, and problem about zookeeper.

I have spent about one more week to configure them. But when

configure successfully, run mapreduce also appear the same problem about space. So I have to give up it.

But during the configuration of hadoop, hbase,hive. I have a deep understand about EC2 instance and hadoop cluster. And also learned a lot.