Case Study 1

Problem Statement:

What are the movie titles that the user has rated?

How many times a movie has been rated by the user?

In question 2 above, what is the average rating given for a movie?

Solution:

Step 1:

Make Case study Directory

hadoop fs –mkdir case study

Step 2 :

Push movies.csv into case study directory

hadoop fs –put movies.csv casestudy

Step 3:

Push ratingss.csv into case study directory

hadoop fs –put ratings.csv casestudy

acadgild@localhost:~

```
[acadgild@localhost ~]$ hadoop fs -mkdir casestudy
18/05/10 14:00:54 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$ hadoop fs -put movies.csv casestudy
18/05/10 14:01:24 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$ hadoop fs -put ratings.csv casestudy
18/05/10 14:01:56 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$
```

Step 4:

hadoop fs –ls casestudy

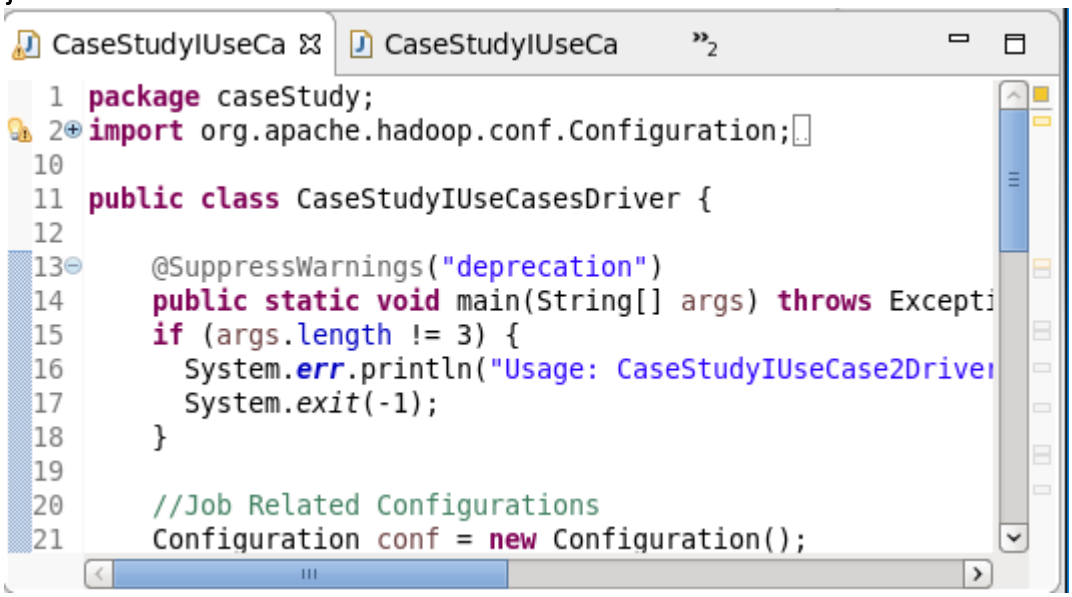
```
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop fs -ls casestudy
18/05/10 14:03:00 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--   1 acadgild supergroup    2283410 2018-05-10 14:01 casestudy/movies.csv
-rw-r--r--   1 acadgild supergroup   25389657 2018-05-10 14:01 casestudy/ratings.csv
[acadgild@localhost ~]$
```

Step 5:

`Driver Code :` CaseStudyIUseCasesDriver

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.MultipleInputs;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class CaseStudyIUseCasesDriver {

        @SuppressWarnings("deprecation")
        public static void main(String[] args) throws Exception {
  if (args.length != 3) {
   System.err.println("Usage: CaseStudyIUseCase2Driver <input path1> <input
path2> <output path>");
   System.exit(-1);
  }

        //Job Related Configurations
        Configuration conf = new Configuration();
```

```
        Job job = new Job(conf, "CaseStudyIUseCase2Driver");
        job.setJarByClass(CaseStudyIUseCasesDriver.class);

        //job.setNumReduceTasks(0);

        //Since there are multiple input, there is a slightly different way of
specifying input path, input format and mapper
        MultipleInputs.addInputPath(job, new
Path(args[0]),TextInputFormat.class, CaseStudyIUseCasesMoviesMapper.class);
        MultipleInputs.addInputPath(job, new
Path(args[1]),TextInputFormat.class, CaseStudyIUseCasesRatingsMapper.class);

        //Set the reducer
        job.setReducerClass(CaseStudyIUseCasesReducer.class);

  //set the out path
        Path outputPath = new Path(args[2]);
        FileOutputFormat.setOutputPath(job, outputPath);
        outputPath.getFileSystem(conf).delete(outputPath, true);

  //set up the output key and value classes
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(Text.class);

  //execute the job
  System.exit(job.waitForCompletion(true) ? 0 : 1);
 }
}
```



```
CaseStudyIUseCa   CaseStudyIUseCa
20      //Job Related Configurations
21      Configuration conf = new Configuration();
22      Job job = new Job(conf, "CaseStudyIUseCase2Driver");
23      job.setJarByClass(CaseStudyIUseCasesDriver.class);
24
25      //job.setNumReduceTasks(0);
26
27      //Since there are multiple input, there is a slightly
28      MultipleInputs.addInputPath(job, new Path(args[0]),Te
29      MultipleInputs.addInputPath(job, new Path(args[1]),Te
30
31      //Set the reducer
32      job.setReducerClass(CaseStudyIUseCasesReducer.class);
```

```
CaseStudyIUseCa   CaseStudyIUseCa
34      //set the out path
35      Path outputPath = new Path(args[2]);
36      FileOutputFormat.setOutputPath(job, outputPath);
37      outputPath.getFileSystem(conf).delete(outputPath, tru
38
39      //set up the output key and value classes
40      job.setOutputKeyClass(Text.class);
41      job.setOutputValueClass(Text.class);
42
43      //execute the job
44      System.exit(job.waitForCompletion(true) ? 0 : 1);
45    }
46  }
```

Step 6 :
Movies Mapper : CaseStudyIUseCasesMoviesMapper

```
import java.io.IOException;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class CaseStudyIUseCasesMoviesMapper extends
Mapper<LongWritable, Text, Text, Text> {
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
```

```
try {
if (key.get() == 0 && value.toString().contains("movieId")){
 return;
 } else {
 String record = value.toString();
 String[] parts = record.split(",");
  context.write(new Text(parts[0]), new Text("movies\t" + parts[1]));
          }
      } catch (Exception e) {
          e.printStackTrace();
      }
           }
}
```

CaseStudyIUseCa | CaseStudyIUseCa

```
1  package caseStudy;
2  import java.io.IOException;
3
4  import org.apache.hadoop.io.LongWritable;
5  import org.apache.hadoop.io.Text;
6  import org.apache.hadoop.mapreduce.Mapper;
7
8  public class CaseStudyIUseCasesMoviesMapper extends
9              Mapper<LongWritable, Text, Text, Text> {
10
11
12     public void map(LongWritable key, Text value, Context
13                 throws IOException, InterruptedException
14
```

```
CaseStudyIUseCa    CaseStudyIUseCa
14
15  try {
16      if (key.get() == 0 && value.toString().contains("mo
17          return;
18      } else {
19          String record = value.toString();
20          String[] parts = record.split(",");
21          context.write(new Text(parts[0]), new Text("mov
22      }
23  } catch (Exception e) {
24      e.printStackTrace();
25  }
26
27
```

Step 7:

Rating Mapper : CaseStudyIUseCasesRatingsMapper

```
import java.io.IOException;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class CaseStudyIUseCasesRatingsMapper extends
Mapper<LongWritable, Text, Text, Text> {
public void map(LongWritable key, Text value, Context context)
throws IOException, InterruptedException {
try {
if (key.get() == 0 && value.toString().contains("userId")){
return;
} else {
String record = value.toString();
String[] parts = record.split(",");
context.write(new Text(parts[1]), new Text("ratings\t" + parts[2]));
 }
} catch (Exception e) {
e.printStackTrace();
}
```

```
}
}
```

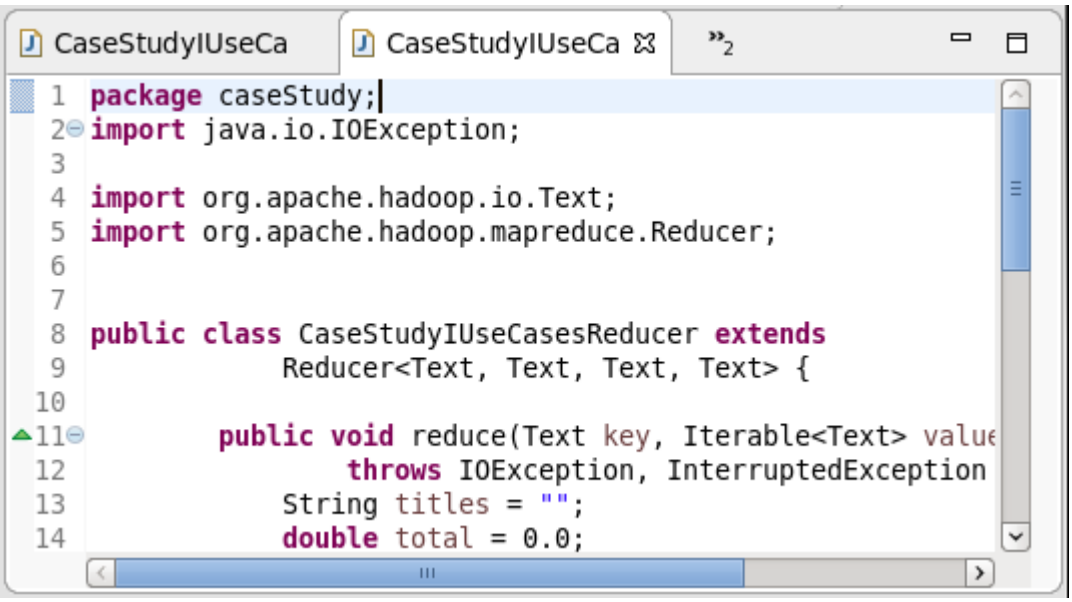
```
package caseStudy;
import java.io.IOException;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;

public class CaseStudyIUseCasesRatingsMapper extends
            Mapper<LongWritable, Text, Text, Text> {

    public void map(LongWritable key, Text value, Context
                throws IOException, InterruptedException

            try {
```

```

    try {
        if (key.get() == 0 && value.toString().contains("us
            return;
        } else {
            String record = value.toString();
            String[] parts = record.split(",");
            context.write(new Text(parts[1]), new Text("rat
        }
    } catch (Exception e) {
        e.printStackTrace();
    }


```

Step 8 :
Reducer : CaseStudyIUseCasesReducer

```
import java.io.IOException;

import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
```

```
public class CaseStudyIUseCasesReducer extends
Reducer<Text, Text, Text, Text> {
public void reduce(Text key, Iterable<Text> values, Context context)
throws IOException, InterruptedException {
String titles = "";
double total = 0.0;
int count = 0;
System.out.println("Text Key    =>"+key.toString());
for (Text t : values) {
String parts[] = t.toString().split("\t");
System.out.println("Text values =>"+t.toString());
if (parts[0].equals("ratings")) {
count++;
String rating = parts[1].trim();
System.out.println("Rating is =>"+rating);
total += Double.parseDouble(rating);
} else if (parts[0].equals("movies")) {
titles = parts[1];
            }
}
double average = total / count;
String str = String.format("%d\t%f", count, average);
context.write(new Text(titles), new Text(str));
}
}
```



```java
        throws IOException, InterruptedException {
    String titles = "";
    double total = 0.0;
    int count = 0;
    System.out.println("Text Key    =>"+key.toString());
    for (Text t : values) {
        String parts[] = t.toString().split("\t");
        System.out.println("Text values =>"+t.toString());
        if (parts[0].equals("ratings")) {
            count++;
            String rating = parts[1].trim();
            System.out.println("Rating is =>"+rating);
            total += Double.parseDouble(rating);
        } else if (parts[0].equals("movies")) {
```

```java
            String rating = parts[1].trim();
            System.out.println("Rating is =>"+rating);
            total += Double.parseDouble(rating);
        } else if (parts[0].equals("movies")) {
            titles = parts[1];
        }
    }

    double average = total / count;
    String str = String.format("%d\t%f", count, average);
    context.write(new Text(titles), new Text(str));
```

Step 9: Run the command

hadoop jar casestudy.jar caseStudy.CaseStudyIUseCasesDriver casestudy/movies.csv casestudy/ratings.csv casestudy/out

```
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop jar casestudy.jar caseStudy.CaseStudyIUseCasesDriver casestudy/movies.csv casestudy/ratings.csv casestudy/out
18/05/10 14:17:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/05/10 14:17:55 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/05/10 14:17:56 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with T
oolRunner to remedy this.
18/05/10 14:17:57 INFO input.FileInputFormat: Total input files to process : 1
18/05/10 14:17:57 INFO input.FileInputFormat: Total input files to process : 1
18/05/10 14:17:57 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1252)
        at java.lang.Thread.join(Thread.java:1326)
        at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:980)
        at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:630)
        at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:807)
18/05/10 14:17:57 INFO mapreduce.JobSubmitter: number of splits:2
18/05/10 14:17:57 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enable
d
18/05/10 14:17:58 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1525940637390_0002
18/05/10 14:18:00 INFO impl.YarnClientImpl: Submitted application application_1525940637390_0002
18/05/10 14:18:00 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1525940637390_0002/
18/05/10 14:18:00 INFO mapreduce.Job: Running job: job_1525940637390_0002
18/05/10 14:18:23 INFO mapreduce.Job: Job job_1525940637390_0002 running in uber mode : false
18/05/10 14:18:23 INFO mapreduce.Job:  map 0% reduce 0%
18/05/10 14:18:46 INFO mapreduce.Job:  map 50% reduce 0%
18/05/10 14:18:52 INFO mapreduce.Job:  map 83% reduce 0%
18/05/10 14:18:54 INFO mapreduce.Job:  map 100% reduce 0%
18/05/10 14:19:10 INFO mapreduce.Job:  map 100% reduce 76%
18/05/10 14:19:16 INFO mapreduce.Job:  map 100% reduce 87%
18/05/10 14:19:22 INFO mapreduce.Job:  map 100% reduce 99%
18/05/10 14:19:23 INFO mapreduce.Job:  map 100% reduce 100%
18/05/10 14:19:24 INFO mapreduce.Job: Job job_1525940637390_0002 completed successfully
18/05/10 14:19:25 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=20203379
                FILE: Number of bytes written=41013414
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=27673611
                HDFS: Number of bytes written=1520050
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
```

```
acadgild@localhost:~
                Launched map tasks=3
                Launched reduce tasks=1
                Data-local map tasks=3
                Total time spent by all maps in occupied slots (ms)=52172
                Total time spent by all reduces in occupied slots (ms)=35338
                Total time spent by all map tasks (ms)=52172
                Total time spent by all reduce tasks (ms)=35338
                Total vcore-milliseconds taken by all map tasks=52172
                Total vcore-milliseconds taken by all reduce tasks=35338
                Total megabyte-milliseconds taken by all map tasks=53424128
                Total megabyte-milliseconds taken by all reduce tasks=36186112
        Map-Reduce Framework
                Map input records=1094420
                Map output records=1094418
                Map output bytes=18014530
                Map output materialized bytes=20203385
                Input split bytes=544
                Combine input records=0
                Combine output records=0
                Reduce input groups=45843
                Reduce shuffle bytes=20203385
                Reduce input records=1094418
                Reduce output records=45843
                Spilled Records=2188836
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=921
                CPU time spent (ms)=29690
                Physical memory (bytes) snapshot=594665472
                Virtual memory (bytes) snapshot=6186311680
                Total committed heap usage (bytes)=381878272
        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=0
        File Output Format Counters
                Bytes Written=1520050
[acadgild@localhost ~]$
```

Step 10:
List the casestudy/out
Hadoop fs –ls casestudy/out

```
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop fs -ls casestudy/out
18/05/10 14:20:49 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--   1 acadgild supergroup          0 2018-05-10 14:19 casestudy/out/_SUCCESS
-rw-r--r--   1 acadgild supergroup    1520050 2018-05-10 14:19 casestudy/out/part-r-00000
[acadgild@localhost ~]$
```

Step 11 : Output
hadoop fs –cat casestudy/out/part-r-00000 | head

```
acadgild@localhost:~
[acadgild@localhost ~]$ hadoop fs -cat casestudy/out/part-r-00000 | head
18/05/10 14:21:38 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Toy Story (1995)        2681    3.897799
GoldenEye (1995)        1300    3.428077
City Hall (1996)        204     3.235294
Curdled (1996)  14      3.250000
"Comic  0       NaN
Up in Smoke (1957)      0       NaN
First Daughter (1999)   0       NaN
"Flaw   1       3.500000
Battle of Los Angeles (2011)    3       3.000000
Jason Becker: Not Dead Yet (2012)       0       NaN
cat: Unable to write to output stream.
[acadgild@localhost ~]$
```