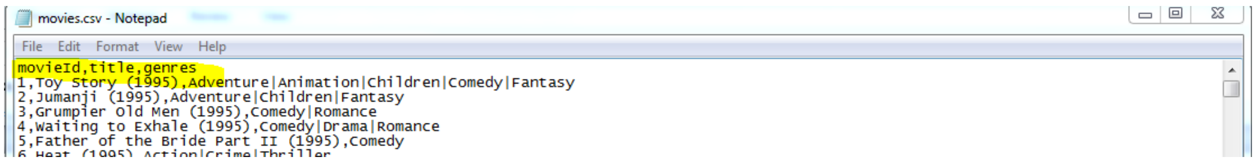
Case Study I for Session 7
(Movie Ratings Case Study)

- Prachi Mohite

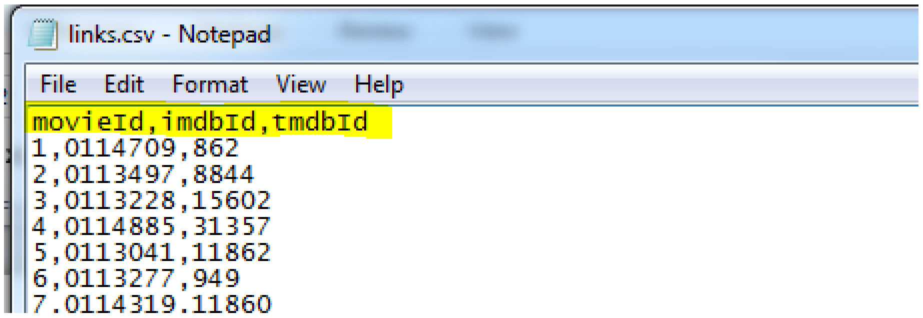
For this case Study we will be using below data sets

- **Movies.csv**
  - **Which has columns as**
    - **movieId,**
    - **title,**
    - **genres**



```
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
4,Waiting to Exhale (1995),Comedy|Drama|Romance
5,Father of the Bride Part II (1995),Comedy
6,Heat (1995),Action|Crime|Thriller
```

- **Ratings.csv**
  - **userId,**
  - **movieId,**
  - **rating,**
  - **timestamp**
- **Links.csv (has below columns)**
  - **movieId,**
  - **imdbId,**
  - **tmdbId**



```
movieId,imdbId,tmdbId
1,0114709,862
2,0113497,8844
3,0113228,15602
4,0114885,31357
5,0113041,11862
6,0113277,949
7,0114319,11860
```

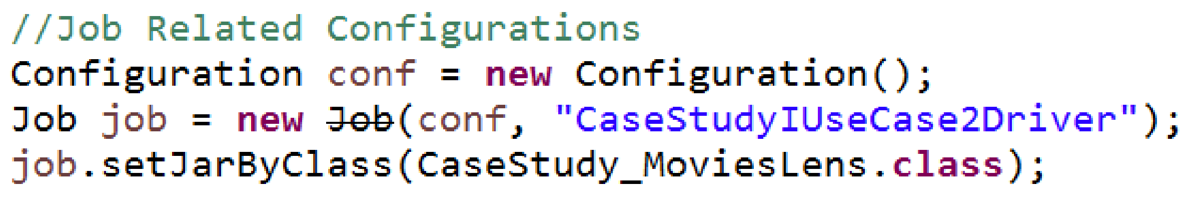
Task 1

1.1 What are the movie titles that the user has rated?

Driver Class.

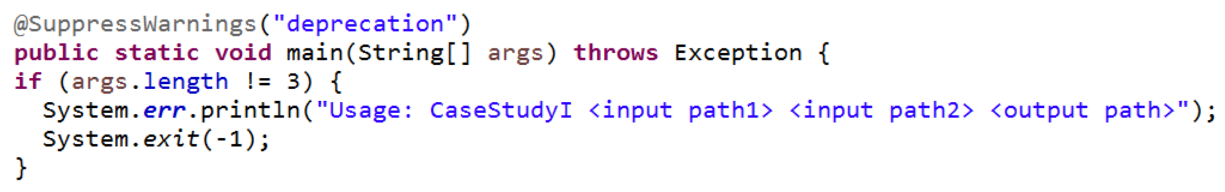
- The driver class which communicates with the Hadoop framework and specifies the configuration elements required to run a MapReduce job. This involves aspects such as telling

Hadoop which Mapper and Reducer classes to use, where to find the input data and in what format, and where to place the output data and how to format it. There is an additional variety of other configuration options that can be set here.
- There is no default parent Driver class as a subclass; the driver logic usually exists in the main method of the class written to encapsulate a MapReduce job.
- Getting the configuration from hadoop configuration



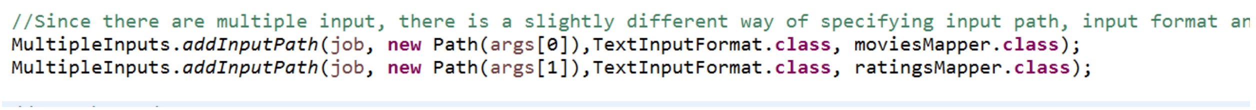
```
//Job Related Configurations
Configuration conf = new Configuration();
Job job = new ~~Job~~(conf, "CaseStudyIUseCase2Driver");
job.setJarByClass(CaseStudy_MoviesLens.class);
```

- Checking if input is given properly or not



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

- As we have multiple input files there is slightly different way to specify these file



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

As we have join movie ratings and movie details we will be using JOINs to get the desired output.

Below are some details for JOIN

Join

The join operation is used to combine two or more database tables based on foreign keys. In general, companies maintain separate tables for the customer and the transaction records in their database. And, many times these companies need to generate analytic reports using the data present in such separate tables. Therefore, they perform a join operation on these separate tables using a common column (foreign key), like movie id, etc., to generate a combined table. Then, they analyze this combined table to get the desired analytic reports.

In this Example we have two files ratings.csv and movies.csv with common (foreign key as ‘movieid’)

Joins in MapReduce

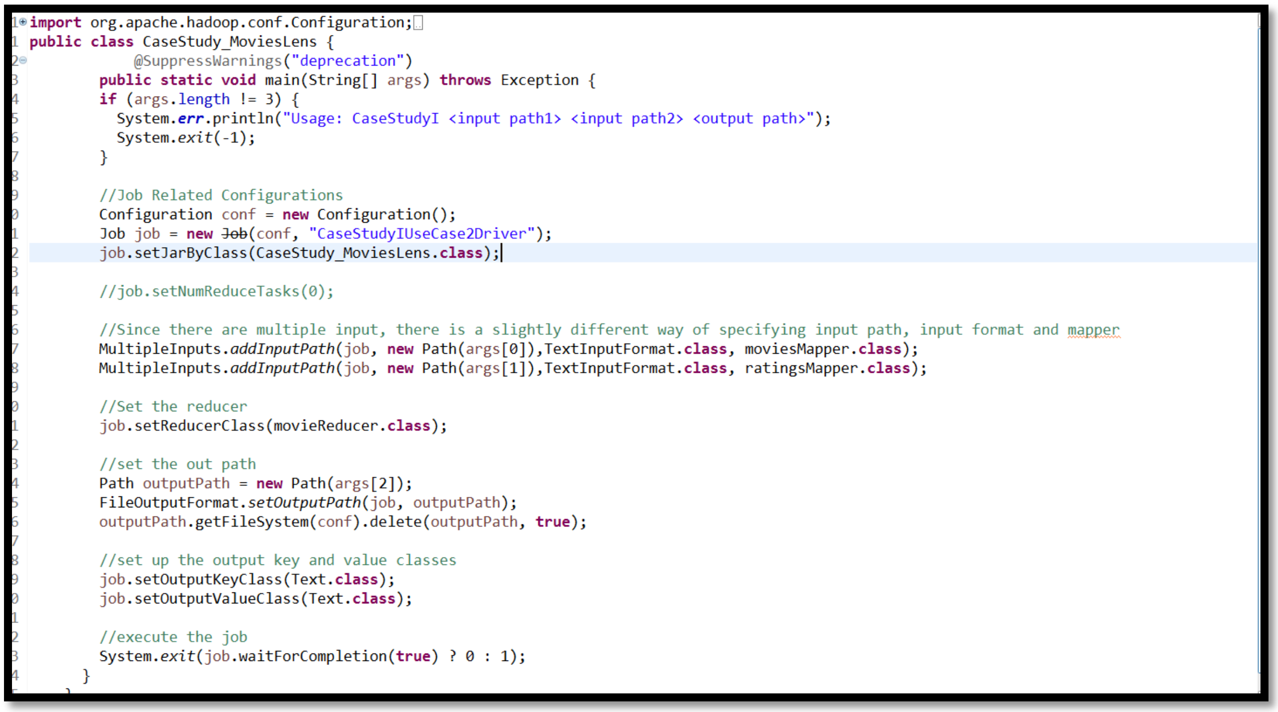
Just like SQL join, we can also perform join operations in MapReduce on different data sets. There are two types of join operations in MapReduce:

- **Map Side Join:** As the name implies, the join operation is performed in the map phase itself. Therefore, in the map side join, the mapper performs the join and it is mandatory that the input to each map is partitioned and sorted according to the keys.

- **Reduce Side Join:** As the name suggests, in the reduce side join, the reducer is responsible for performing the join operation. It is comparatively simple and easier to implement than the map side join as the sorting and shuffling phase sends the values having identical keys to the same reducer and therefore, by default, the data is organized for us.

We will be using Reduce Side Join in our Case Study. As mentioned above we have written code to load multiple input files.

Complete Code of Driver Class



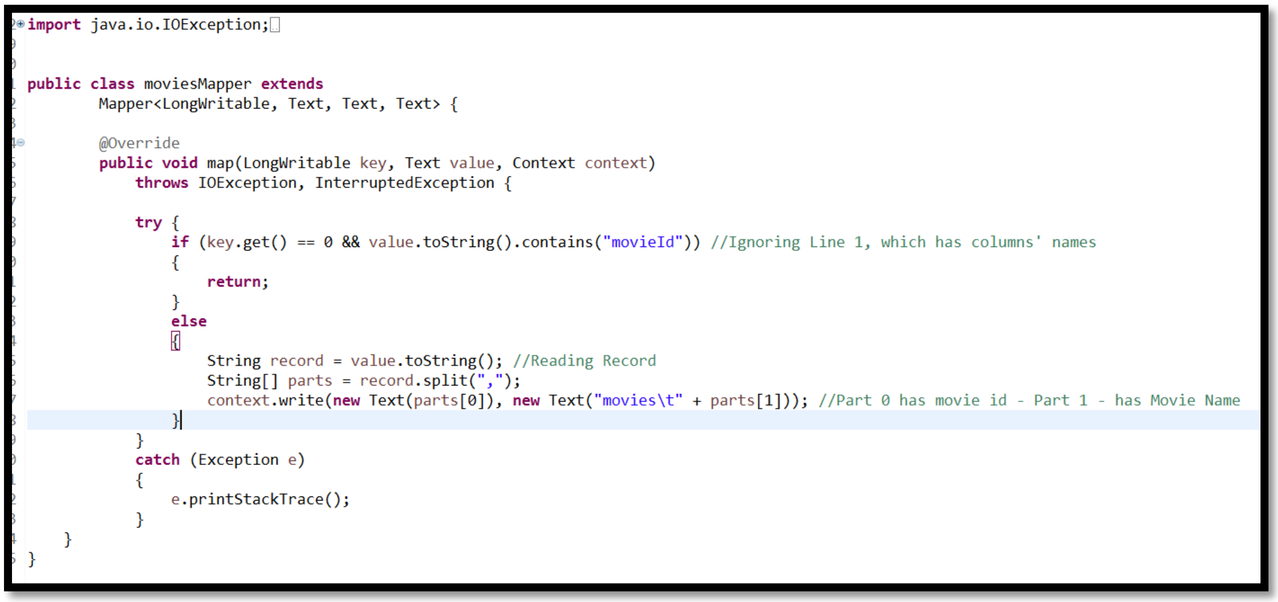
```java
import org.apache.hadoop.conf.Configuration;
public class CaseStudy_MoviesLens {
        @SuppressWarnings("deprecation")
        public static void main(String[] args) throws Exception {
        if (args.length != 3) {
          System.err.println("Usage: CaseStudyI <input path1> <input path2> <output path>");
          System.exit(-1);
        }

        //Job Related Configurations
        Configuration conf = new Configuration();
        Job job = new Job(conf, "CaseStudyIUseCase2Driver");
        job.setJarByClass(CaseStudy_MoviesLens.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, moviesMapper.class);
        MultipleInputs.addInputPath(job, new Path(args[1]),TextInputFormat.class, ratingsMapper.class);

        //Set the reducer
        job.setReducerClass(movieReducer.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);
      }
```

Now we have to write two mappers considering the same (foreign key) as movie ID

1. **moviesMapper –** This mapper will read the movies.csv and create the records with
2. movie names and movie id.

To write a mapper we need to extend mapper class and override the Map method

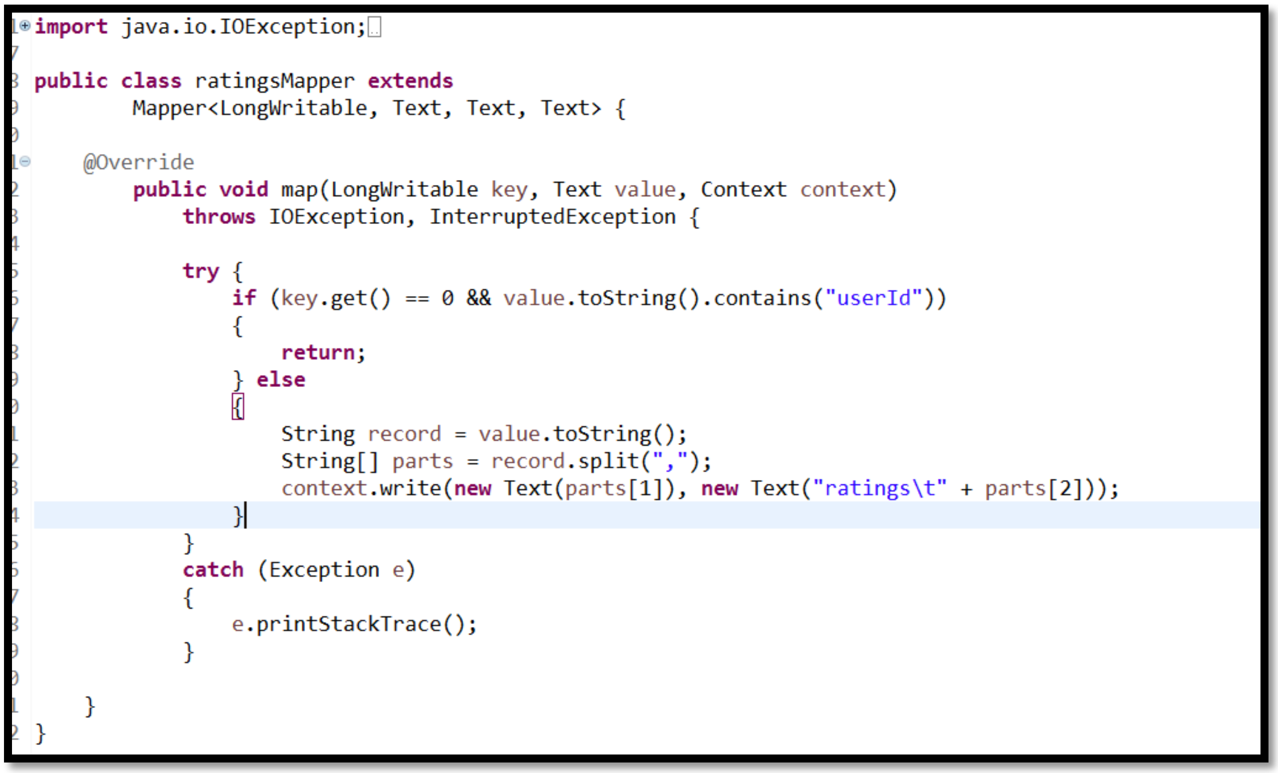
Complete Code of moviesMapper as below



```java
import java.io.IOException;

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

        @Override
        public void map(LongWritable key, Text value, Context context)
            throws IOException, InterruptedException {

            try {
                if (key.get() == 0 && value.toString().contains("movieId")) //Ignoring Line 1, which has columns' names
                {
                    return;
                }
                else
                {
                    String record = value.toString(); //Reading Record
                    String[] parts = record.split(",");
                    context.write(new Text(parts[0]), new Text("movies\t" + parts[1])); //Part 0 has movie id - Part 1 - has Movie Name
                }
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
    }
}
```

3. **ratingsMapper – This mapper will read the entries from ratings.csv and create the records with movieid and ratings**

Complete Code of ratingsMapper



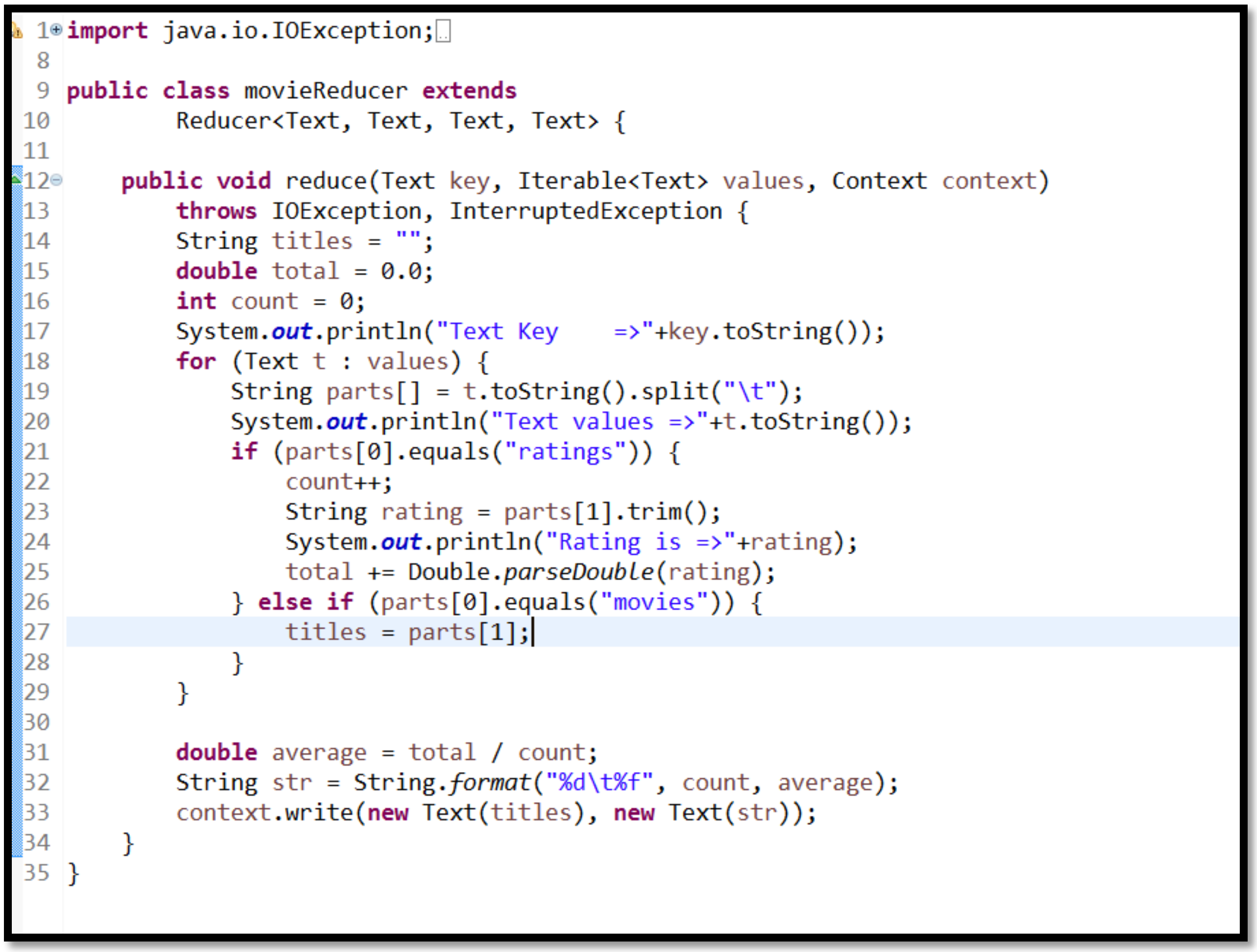
```java
import java.io.IOException;

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

    @Override
        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();
            }

    }
}
```

Reducer Class

- Reducer Class
- It reads the output generated by different mappers.
- The output of reducer is final output in HDFS
- Reducers run in parallel since they are independent of one another. The user decides the number of reducers. By default number of reducers is 1.
- Phases of Reducer
    - Shuffle
    - Sort
    - Reduce

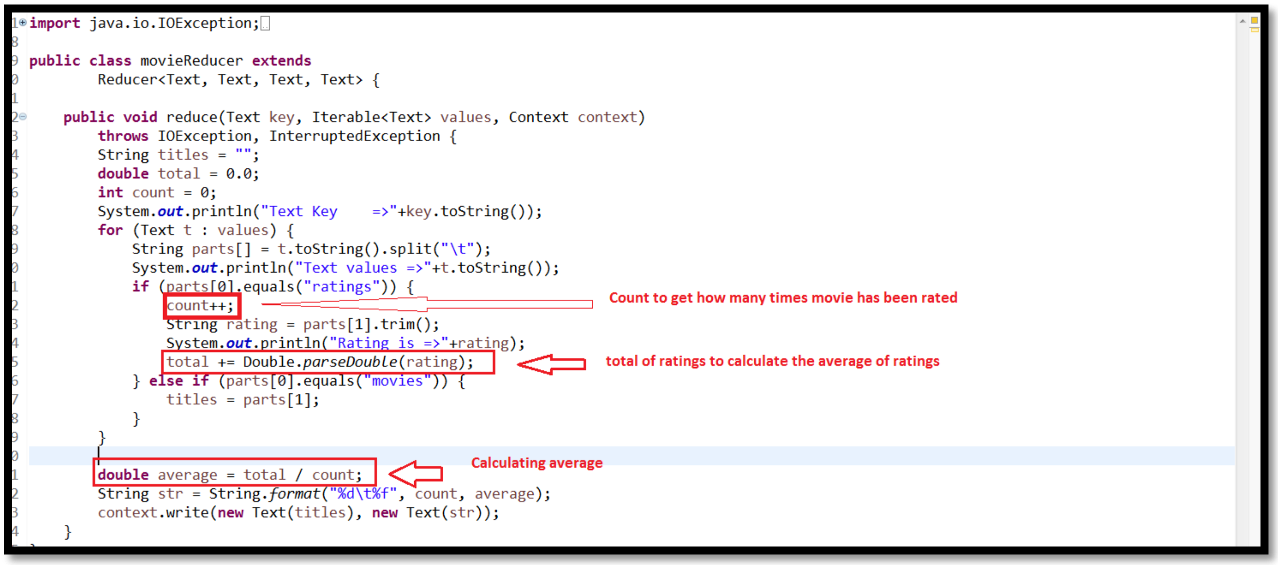
Written Reducer to get the output from above both the mapper and reduce the output based on the unique key i.e. ‘movieid’ and return the output where movies are rated and with their respective ratings

Complete Code of Reducer



```
import java.io.IOException;

public class movieReducer 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));
    }
}
```

1.2 How many times a movie has been rated by the user?
1.3 In question 2 above, what is the average rating given for a movie?
- Above tasks will require same two mappers (one for movies and other for ratings) in reducer we have to add code to make count of when movie is rated and to calculate the average. Below are snap shot for the same from reducer code

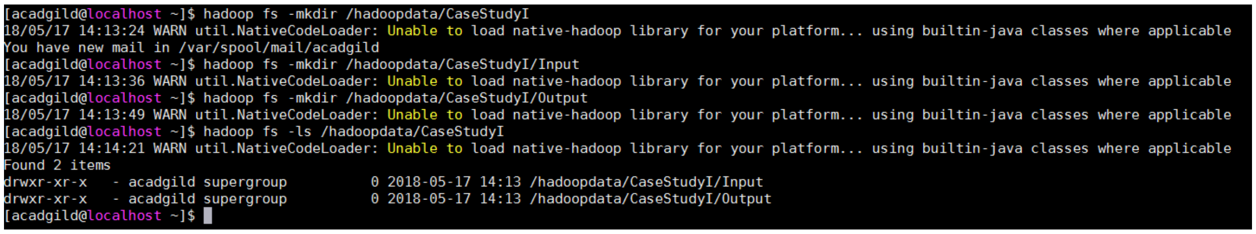


```java
import java.io.IOException;

public class movieReducer 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));
    }
```

Count to get how many times movie has been rated
total of ratings to calculate the average of ratings
Calculating average

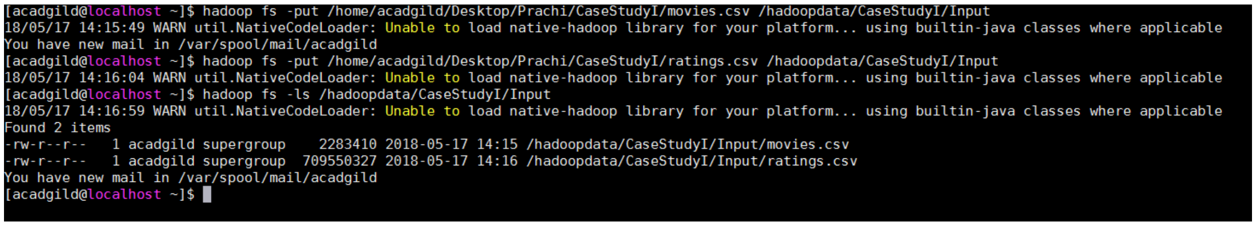
Now we will execute this code. Make sure your hadoop is running and required data sets are placed

Created Directory in hadoop as /hadoopdata/CaseStudyI/Input and /hadoopdata/CaseStudyI/output



```
[acadgild@localhost ~]$ hadoop fs -mkdir /hadoopdata/CaseStudyI
18/05/17 14:13:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ hadoop fs -mkdir /hadoopdata/CaseStudyI/Input
18/05/17 14:13:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$ hadoop fs -mkdir /hadoopdata/CaseStudyI/Output
18/05/17 14:13:49 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$ hadoop fs -ls /hadoopdata/CaseStudyI
18/05/17 14:14:21 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x   - acadgild supergroup          0 2018-05-17 14:13 /hadoopdata/CaseStudyI/Input
drwxr-xr-x   - acadgild supergroup          0 2018-05-17 14:13 /hadoopdata/CaseStudyI/Output
[acadgild@localhost ~]$
```

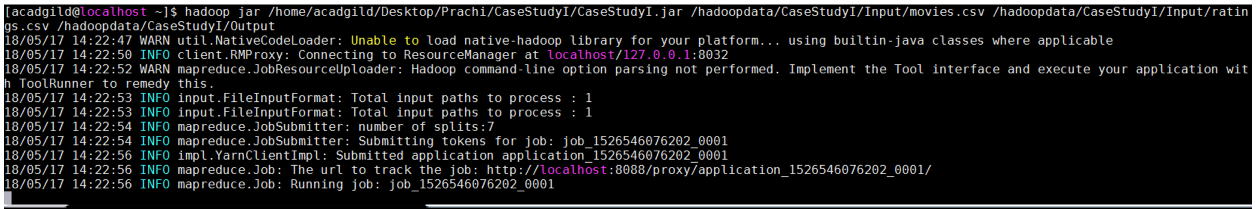
Put the inputfiles movies.csv and ratings.csv to the Input folder



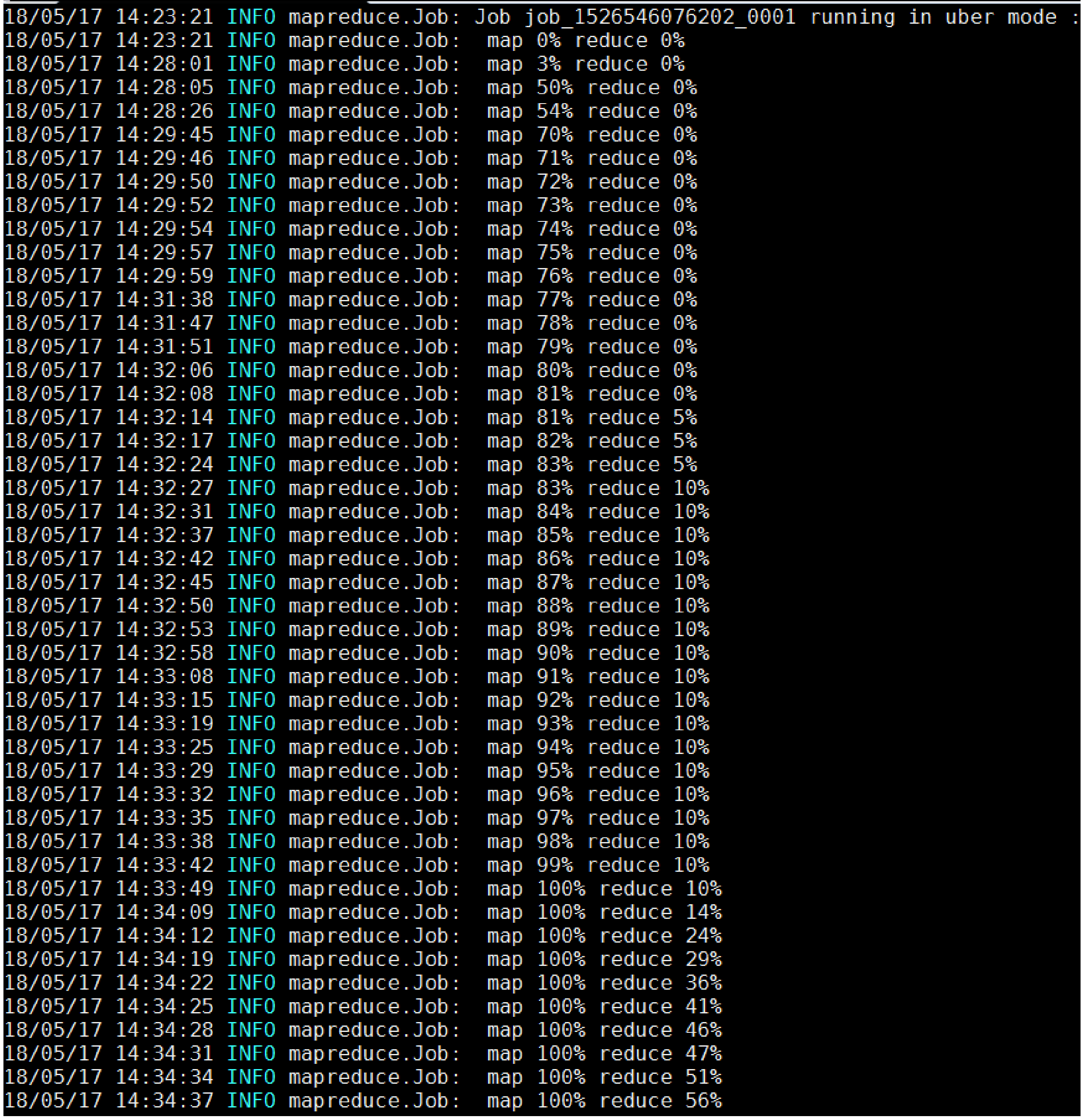
```
[acadgild@localhost ~]$ hadoop fs -put /home/acadgild/Desktop/Prachi/CaseStudyI/movies.csv /hadoopdata/CaseStudyI/Input
18/05/17 14:15:49 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ hadoop fs -put /home/acadgild/Desktop/Prachi/CaseStudyI/ratings.csv /hadoopdata/CaseStudyI/Input
18/05/17 14:16:04 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
[acadgild@localhost ~]$ hadoop fs -ls /hadoopdata/CaseStudyI/Input
18/05/17 14:16:59 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 2018-05-17 14:15 /hadoopdata/CaseStudyI/Input/movies.csv
-rw-r--r--   1 acadgild supergroup  709550327 2018-05-17 14:16 /hadoopdata/CaseStudyI/Input/ratings.csv
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$
```

Execute the Jar file

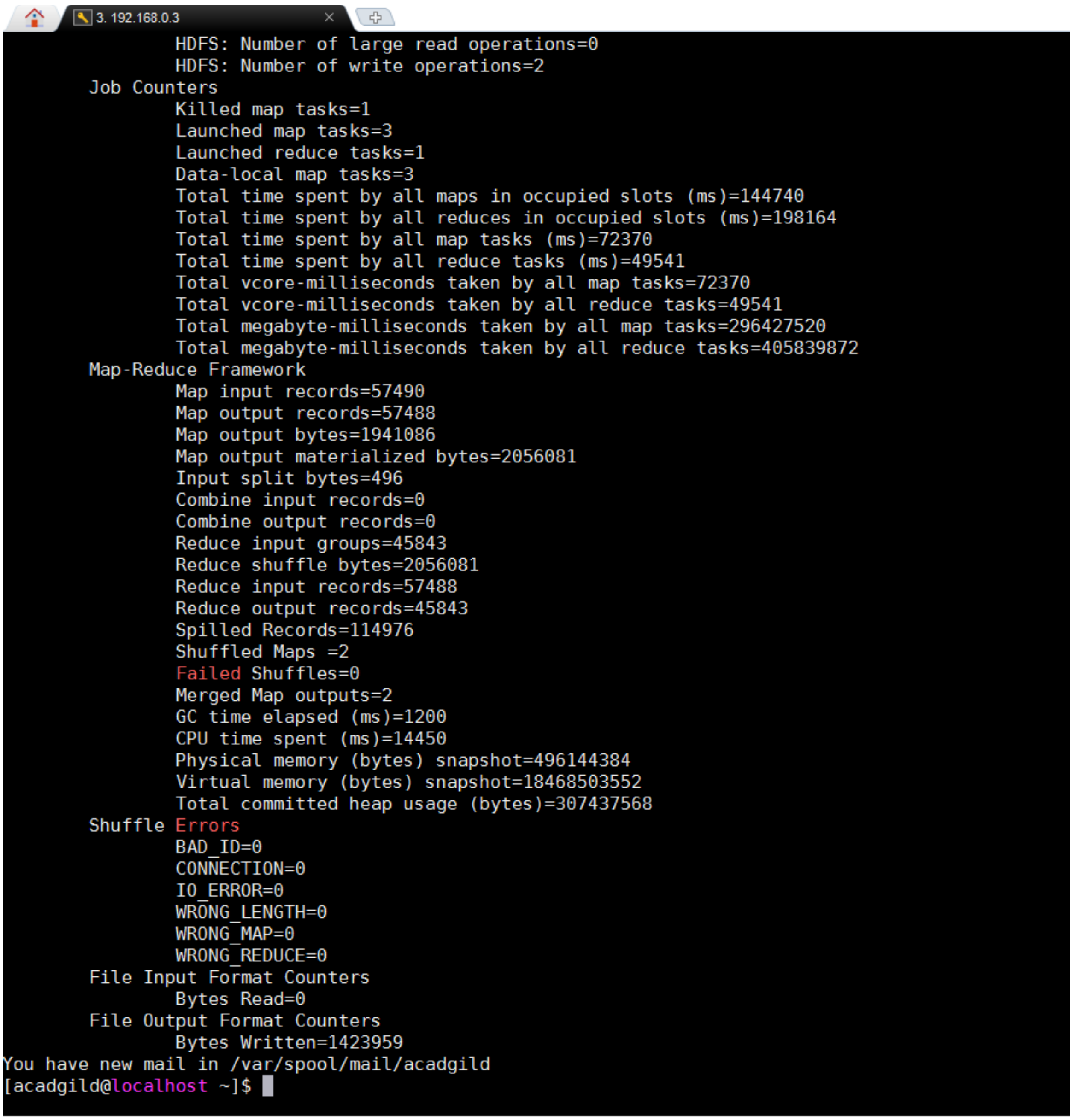
Command - hadoop jar /home/acadgild/Desktop/Prachi/CaseStudyI/CaseStudyI.jar /hadoopdata/CaseStudyI/Input/movies.csv /hadoopdata/CaseStudyI/Input/ratings.csv / hadoopdata/CaseStudyI/Output



```
[acadgild@localhost ~]$ hadoop jar /home/acadgild/Desktop/Prachi/CaseStudyI/CaseStudyI.jar /hadoopdata/CaseStudyI/Input/movies.csv /hadoopdata/CaseStudyI/Input/ratin
gs.csv /hadoopdata/CaseStudyI/Output
18/05/17 14:22:47 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/05/17 14:22:50 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/05/17 14:22:52 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application wit
h ToolRunner to remedy this.
18/05/17 14:22:53 INFO input.FileInputFormat: Total input paths to process : 1
18/05/17 14:22:53 INFO input.FileInputFormat: Total input paths to process : 1
18/05/17 14:22:54 INFO mapreduce.JobSubmitter: number of splits:7
18/05/17 14:22:54 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1526546076202_0001
18/05/17 14:22:56 INFO impl.YarnClientImpl: Submitted application application_1526546076202_0001
18/05/17 14:22:56 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1526546076202_0001/
18/05/17 14:22:56 INFO mapreduce.Job: Running job: job_1526546076202_0001
```

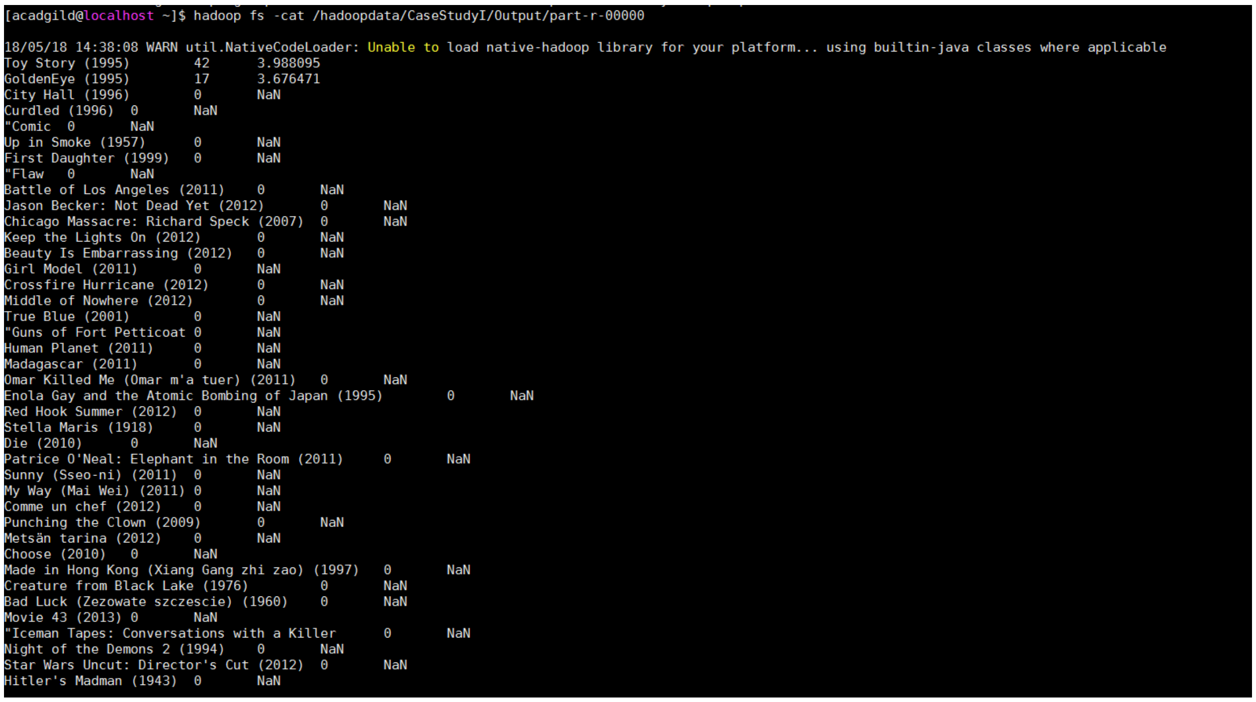


```
18/05/17 14:23:21 INFO mapreduce.Job: Job job_1526546076202_0001 running in uber mode :
18/05/17 14:23:21 INFO mapreduce.Job:  map 0% reduce 0%
18/05/17 14:28:01 INFO mapreduce.Job:  map 3% reduce 0%
18/05/17 14:28:05 INFO mapreduce.Job:  map 50% reduce 0%
18/05/17 14:28:26 INFO mapreduce.Job:  map 54% reduce 0%
18/05/17 14:29:45 INFO mapreduce.Job:  map 70% reduce 0%
18/05/17 14:29:46 INFO mapreduce.Job:  map 71% reduce 0%
18/05/17 14:29:50 INFO mapreduce.Job:  map 72% reduce 0%
18/05/17 14:29:52 INFO mapreduce.Job:  map 73% reduce 0%
18/05/17 14:29:54 INFO mapreduce.Job:  map 74% reduce 0%
18/05/17 14:29:57 INFO mapreduce.Job:  map 75% reduce 0%
18/05/17 14:29:59 INFO mapreduce.Job:  map 76% reduce 0%
18/05/17 14:31:38 INFO mapreduce.Job:  map 77% reduce 0%
18/05/17 14:31:47 INFO mapreduce.Job:  map 78% reduce 0%
18/05/17 14:31:51 INFO mapreduce.Job:  map 79% reduce 0%
18/05/17 14:32:06 INFO mapreduce.Job:  map 80% reduce 0%
18/05/17 14:32:08 INFO mapreduce.Job:  map 81% reduce 0%
18/05/17 14:32:14 INFO mapreduce.Job:  map 81% reduce 5%
18/05/17 14:32:17 INFO mapreduce.Job:  map 82% reduce 5%
18/05/17 14:32:24 INFO mapreduce.Job:  map 83% reduce 5%
18/05/17 14:32:27 INFO mapreduce.Job:  map 83% reduce 10%
18/05/17 14:32:31 INFO mapreduce.Job:  map 84% reduce 10%
18/05/17 14:32:37 INFO mapreduce.Job:  map 85% reduce 10%
18/05/17 14:32:42 INFO mapreduce.Job:  map 86% reduce 10%
18/05/17 14:32:45 INFO mapreduce.Job:  map 87% reduce 10%
18/05/17 14:32:50 INFO mapreduce.Job:  map 88% reduce 10%
18/05/17 14:32:53 INFO mapreduce.Job:  map 89% reduce 10%
18/05/17 14:32:58 INFO mapreduce.Job:  map 90% reduce 10%
18/05/17 14:33:08 INFO mapreduce.Job:  map 91% reduce 10%
18/05/17 14:33:15 INFO mapreduce.Job:  map 92% reduce 10%
18/05/17 14:33:19 INFO mapreduce.Job:  map 93% reduce 10%
18/05/17 14:33:25 INFO mapreduce.Job:  map 94% reduce 10%
18/05/17 14:33:29 INFO mapreduce.Job:  map 95% reduce 10%
18/05/17 14:33:32 INFO mapreduce.Job:  map 96% reduce 10%
18/05/17 14:33:35 INFO mapreduce.Job:  map 97% reduce 10%
18/05/17 14:33:38 INFO mapreduce.Job:  map 98% reduce 10%
18/05/17 14:33:42 INFO mapreduce.Job:  map 99% reduce 10%
18/05/17 14:33:49 INFO mapreduce.Job:  map 100% reduce 10%
18/05/17 14:34:09 INFO mapreduce.Job:  map 100% reduce 14%
18/05/17 14:34:12 INFO mapreduce.Job:  map 100% reduce 24%
18/05/17 14:34:19 INFO mapreduce.Job:  map 100% reduce 29%
18/05/17 14:34:22 INFO mapreduce.Job:  map 100% reduce 36%
18/05/17 14:34:25 INFO mapreduce.Job:  map 100% reduce 41%
18/05/17 14:34:28 INFO mapreduce.Job:  map 100% reduce 46%
18/05/17 14:34:31 INFO mapreduce.Job:  map 100% reduce 47%
18/05/17 14:34:34 INFO mapreduce.Job:  map 100% reduce 51%
18/05/17 14:34:37 INFO mapreduce.Job:  map 100% reduce 56%
```

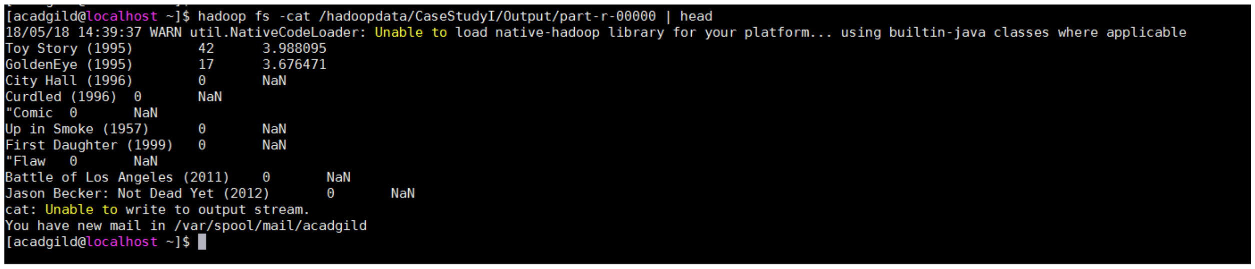


```
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=1
                Launched map tasks=3
                Launched reduce tasks=1
                Data-local map tasks=3
                Total time spent by all maps in occupied slots (ms)=144740
                Total time spent by all reduces in occupied slots (ms)=198164
                Total time spent by all map tasks (ms)=72370
                Total time spent by all reduce tasks (ms)=49541
                Total vcore-milliseconds taken by all map tasks=72370
                Total vcore-milliseconds taken by all reduce tasks=49541
                Total megabyte-milliseconds taken by all map tasks=296427520
                Total megabyte-milliseconds taken by all reduce tasks=405839872
        Map-Reduce Framework
                Map input records=57490
                Map output records=57488
                Map output bytes=1941086
                Map output materialized bytes=2056081
                Input split bytes=496
                Combine input records=0
                Combine output records=0
                Reduce input groups=45843
                Reduce shuffle bytes=2056081
                Reduce input records=57488
                Reduce output records=45843
                Spilled Records=114976
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=1200
                CPU time spent (ms)=14450
                Physical memory (bytes) snapshot=496144384
                Virtual memory (bytes) snapshot=18468503552
                Total committed heap usage (bytes)=307437568
        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=1423959
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$
```

Output after execution



```
[acadgild@localhost ~]$ hadoop fs -cat /hadoopdata/CaseStudyI/Output/part-r-00000

18/05/18 14:38:08 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Toy Story (1995)        42      3.988095
GoldenEye (1995)        17      3.676471
City Hall (1996)        0       NaN
Curdled (1996)  0       NaN
"Comic  0       NaN
Up in Smoke (1957)      0       NaN
First Daughter (1999)   0       NaN
"Flaw   0       NaN
Battle of Los Angeles (2011)    0       NaN
Jason Becker: Not Dead Yet (2012)       0       NaN
Chicago Massacre: Richard Speck (2007)  0       NaN
Keep the Lights On (2012)       0       NaN
Beauty Is Embarrassing (2012)   0       NaN
Girl Model (2011)       0       NaN
Crossfire Hurricane (2012)      0       NaN
Middle of Nowhere (2012)        0       NaN
True Blue (2001)        0       NaN
"Guns of Fort Petticoat 0       NaN
Human Planet (2011)     0       NaN
Madagascar (2011)       0       NaN
Omar Killed Me (Omar m'a tuer) (2011)   0       NaN
Enola Gay and the Atomic Bombing of Japan (1995)        0       NaN
Red Hook Summer (2012)  0       NaN
Stella Maris (1918)     0       NaN
Die (2010)      0       NaN
Patrice O'Neal: Elephant in the Room (2011)     0       NaN
Sunny (Sseo-ni) (2011)  0       NaN
My Way (Mai Wei) (2011) 0       NaN
Comme un chef (2012)    0       NaN
Punching the Clown (2009)       0       NaN
Metsän tarina (2012)    0       NaN
Choose (2010)   0       NaN
Made in Hong Kong (Xiang Gang zhi zao) (1997)   0       NaN
Creature from Black Lake (1976)         0       NaN
Bad Luck (Zezowate szczescie) (1960)    0       NaN
Movie 43 (2013) 0       NaN
"Iceman Tapes: Conversations with a Killer      0       NaN
Night of the Demons 2 (1994)    0       NaN
Star Wars Uncut: Director's Cut (2012)  0       NaN
Hitler's Madman (1943)  0       NaN
```



```
[acadgild@localhost ~]$ hadoop fs -cat /hadoopdata/CaseStudyI/Output/part-r-00000 | head
18/05/18 14:39:37 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Toy Story (1995)        42      3.988095
GoldenEye (1995)        17      3.676471
City Hall (1996)        0       NaN
Curdled (1996)  0       NaN
"Comic  0       NaN
Up in Smoke (1957)      0       NaN
First Daughter (1999)   0       NaN
"Flaw   0       NaN
Battle of Los Angeles (2011)    0       NaN
Jason Becker: Not Dead Yet (2012)       0       NaN
cat: Unable to write to output stream.
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost ~]$ 
```