# BIG DATA HADOOP AND SPARK DEVLOPMENT

# CASE STUDY I

Table of Contents:

# BIG DATA HADOOPAND SPARK DEVELOPMENT

## 1. Introduction

In this case study, the given tasks are performed and Output of the tasks are recorded in the form of Screenshots.

## 2. Objective

This case study consolidates the deeper understanding of the Sessions

## 3. Problem Statement

- Task 1

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?

## 4. Expected Output

- ### Task 1
  - 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?

Mapper class

```
package
MovieRatingPackage;

                      import java.io.IOException;

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

                      public class MovieMapper extends Mapper<Object, Text, Text,
                      Text> {
                             public void map(Object key, Text value, Context context)
                                           throws IOException, InterruptedException {
                                    int j = 0;
                                    String record = value.toString();
                                    String[] parts = record.split(",");

                                    if (parts[0].equals("movieId")) {
                                           j=1;
                                    }
                                    if(j != 1) {
                                           context.write(new Text(parts[0]), new
                      Text("movies\t" + parts[1]));
                                    }
                             }
                      }
```

## Driver class

```
package
MovieRatingPackage;

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


public class MovieRatingDriver {

        public static void main(String[] args) throws Exception
{
        // TODO Auto-generated method stub
        if (args.length != 3) {
        System.err.println("Usage: MovieRatingDriver <input
path1> <input path2> <output path>");
        System.exit(-1);
            }

        //Job Related Configurations
        Configuration conf = new Configuration();
        Job job = new Job(conf, "Movie-Rating Join");
        job.setJarByClass(MovieRatingDriver.class);

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

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

        //Set the output key and value class
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

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

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

        }

}
```

## Rating Mapper class

```
Package
MovieRatingPackage;

                      import java.io.IOException;


                      import org.apache.hadoop.io.Text;
                      import org.apache.hadoop.mapreduce.Mapper;
                      import org.apache.hadoop.mapreduce.Mapper.Context;
```

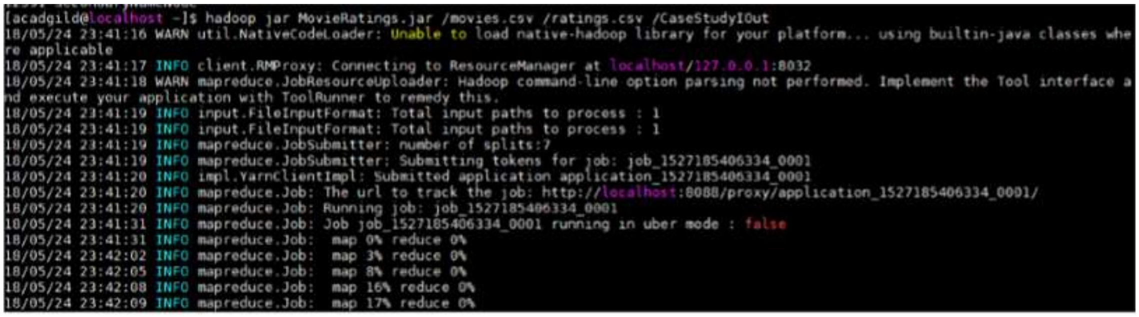
```
public class RatingsMapper extends
                      Mapper<Object, Text, Text, Text>{
       public void map(Object key, Text value, Context context)
                      throws IOException, InterruptedException {

              int j = 0;
              String record = value.toString();
              String[] parts = record.split(",");

              if (parts[1].equals("movieId")) {
                     j=1;
              }
              if(j != 1) {
              context.write(new Text(parts[1]), new
Text("ratings\t" + parts[2]));
              }
       }
}
```

Average rating of the movie

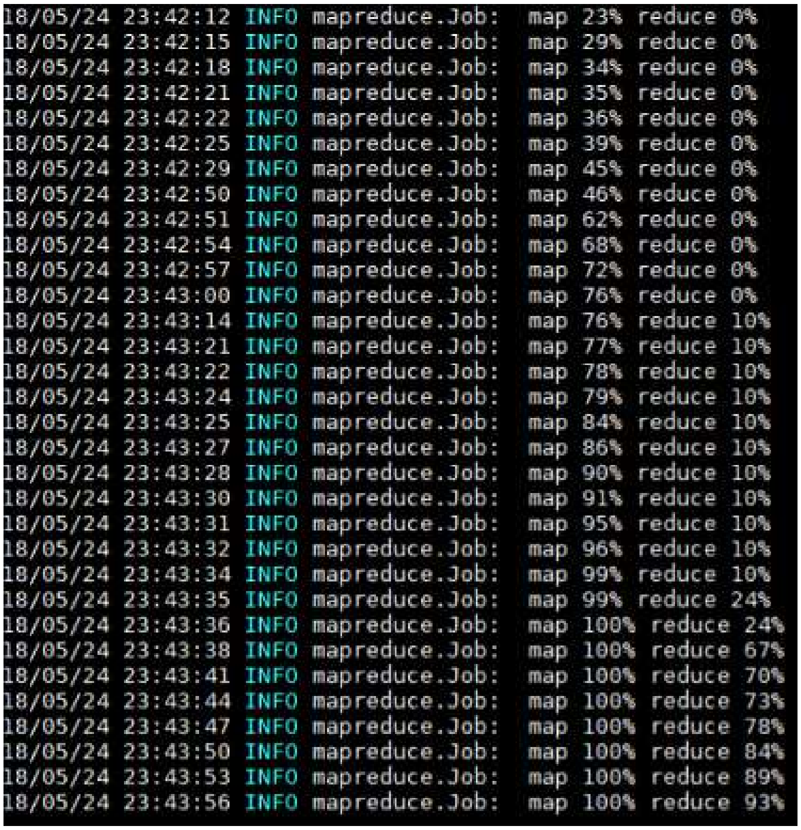
```
package
MovieRatingPackage;
                      import java.io.IOException;

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

                      public class MovieJoinReducer  extends
                      Reducer<Text, Text, Text, Text>{
                      public void reduce(Text key, Iterable<Text> values, Context
                      context)
                                    throws IOException, InterruptedException {
                                    String movieName = "";
                                    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("movies")) {
                                    movieName = parts[1];
                                           }else if (parts[0].equals("ratings")) {
                             count ++;
                             String movieRating = parts[1].trim();
                             total += Double.parseDouble(movieRating);
                             }
```

```
            }

    double avgRating = total / count;// average rating of the movie
    String str = String.format("%d\t%f", count, avgRating);
    context.write(new Text(movieName), new Text(str));
    }
    }
```

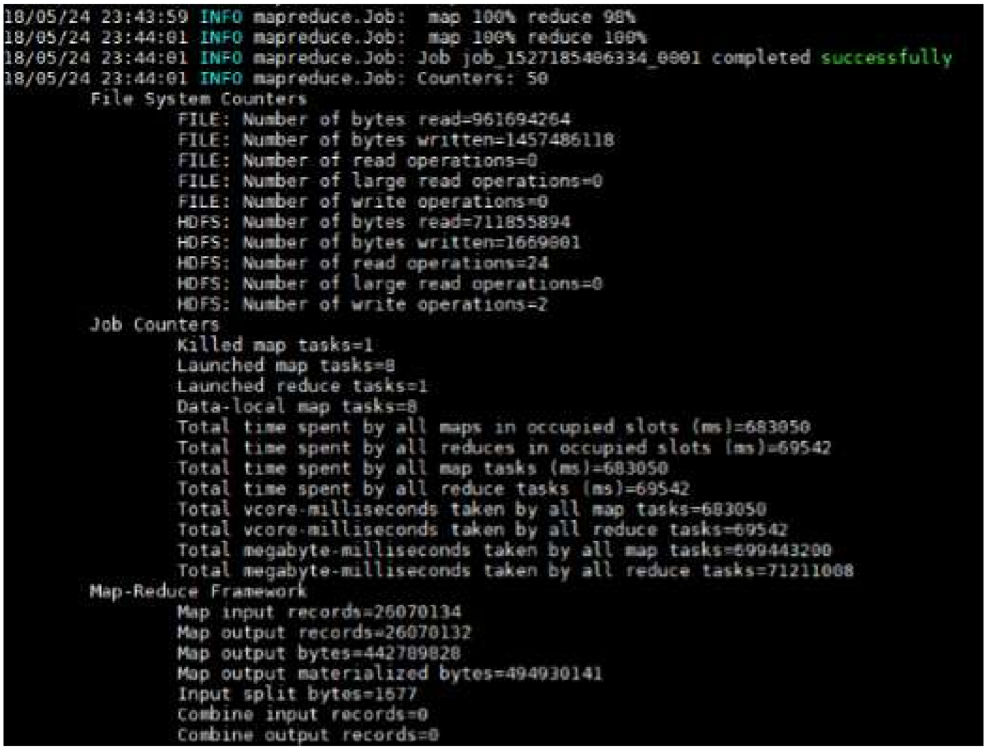
Output



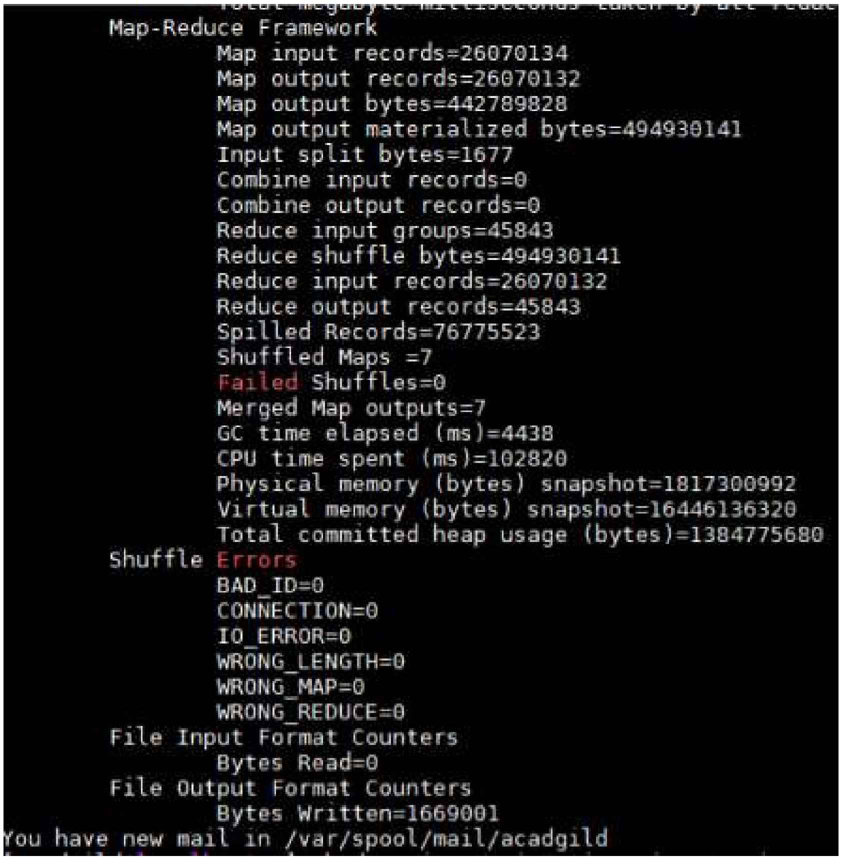
```
[acadgild@localhost ~]$ hadoop jar MovieRatings.jar /movies.csv /ratings.csv /CaseStudyIOut
18/05/24 23:41:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes whe
re applicable
18/05/24 23:41:17 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/05/24 23:41:18 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface a
nd execute your application with ToolRunner to remedy this.
18/05/24 23:41:19 INFO input.FileInputFormat: Total input paths to process : 1
18/05/24 23:41:19 INFO input.FileInputFormat: Total input paths to process : 1
18/05/24 23:41:19 INFO mapreduce.JobSubmitter: number of splits:7
18/05/24 23:41:19 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1527185406334_0001
18/05/24 23:41:20 INFO impl.YarnClientImpl: Submitted application application_1527185406334_0001
18/05/24 23:41:20 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1527185406334_0001/
18/05/24 23:41:20 INFO mapreduce.Job: Running job: job_1527185406334_0001
18/05/24 23:41:31 INFO mapreduce.Job: Job job_1527185406334_0001 running in uber mode : false
18/05/24 23:41:31 INFO mapreduce.Job:  map 0% reduce 0%
18/05/24 23:42:02 INFO mapreduce.Job:  map 3% reduce 0%
18/05/24 23:42:05 INFO mapreduce.Job:  map 8% reduce 0%
18/05/24 23:42:08 INFO mapreduce.Job:  map 16% reduce 0%
18/05/24 23:42:09 INFO mapreduce.Job:  map 17% reduce 0%
```



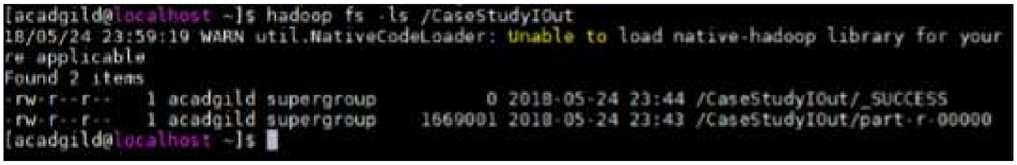
```
18/05/24 23:42:12 INFO mapreduce.Job:  map 23% reduce 0%
18/05/24 23:42:15 INFO mapreduce.Job:  map 29% reduce 0%
18/05/24 23:42:18 INFO mapreduce.Job:  map 34% reduce 0%
18/05/24 23:42:21 INFO mapreduce.Job:  map 35% reduce 0%
18/05/24 23:42:22 INFO mapreduce.Job:  map 36% reduce 0%
18/05/24 23:42:25 INFO mapreduce.Job:  map 39% reduce 0%
18/05/24 23:42:29 INFO mapreduce.Job:  map 45% reduce 0%
18/05/24 23:42:50 INFO mapreduce.Job:  map 46% reduce 0%
18/05/24 23:42:51 INFO mapreduce.Job:  map 62% reduce 0%
18/05/24 23:42:54 INFO mapreduce.Job:  map 68% reduce 0%
18/05/24 23:42:57 INFO mapreduce.Job:  map 72% reduce 0%
18/05/24 23:43:00 INFO mapreduce.Job:  map 76% reduce 0%
18/05/24 23:43:14 INFO mapreduce.Job:  map 76% reduce 10%
18/05/24 23:43:21 INFO mapreduce.Job:  map 77% reduce 10%
18/05/24 23:43:22 INFO mapreduce.Job:  map 78% reduce 10%
18/05/24 23:43:24 INFO mapreduce.Job:  map 79% reduce 10%
18/05/24 23:43:25 INFO mapreduce.Job:  map 84% reduce 10%
18/05/24 23:43:27 INFO mapreduce.Job:  map 86% reduce 10%
18/05/24 23:43:28 INFO mapreduce.Job:  map 90% reduce 10%
18/05/24 23:43:30 INFO mapreduce.Job:  map 91% reduce 10%
18/05/24 23:43:31 INFO mapreduce.Job:  map 95% reduce 10%
18/05/24 23:43:32 INFO mapreduce.Job:  map 96% reduce 10%
18/05/24 23:43:34 INFO mapreduce.Job:  map 99% reduce 10%
18/05/24 23:43:35 INFO mapreduce.Job:  map 99% reduce 24%
18/05/24 23:43:36 INFO mapreduce.Job:  map 100% reduce 24%
18/05/24 23:43:38 INFO mapreduce.Job:  map 100% reduce 67%
18/05/24 23:43:41 INFO mapreduce.Job:  map 100% reduce 70%
18/05/24 23:43:44 INFO mapreduce.Job:  map 100% reduce 73%
18/05/24 23:43:47 INFO mapreduce.Job:  map 100% reduce 78%
18/05/24 23:43:50 INFO mapreduce.Job:  map 100% reduce 84%
18/05/24 23:43:53 INFO mapreduce.Job:  map 100% reduce 89%
18/05/24 23:43:56 INFO mapreduce.Job:  map 100% reduce 93%
```



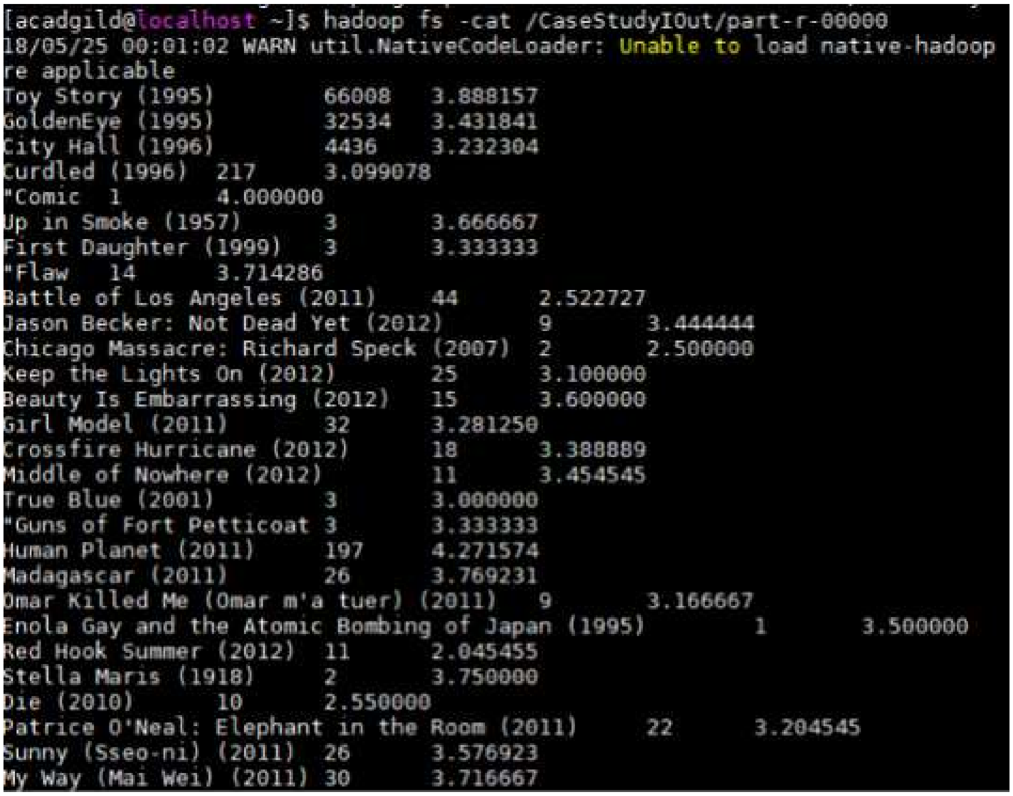
```
18/05/24 23:43:59 INFO mapreduce.Job:  map 100% reduce 98%
18/05/24 23:44:01 INFO mapreduce.Job:  map 100% reduce 100%
18/05/24 23:44:01 INFO mapreduce.Job: Job job_1527185406334_0001 completed successfully
18/05/24 23:44:01 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=961694264
                FILE: Number of bytes written=1457486118
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=711855894
                HDFS: Number of bytes written=1669001
                HDFS: Number of read operations=24
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=1
                Launched map tasks=8
                Launched reduce tasks=1
                Data-local map tasks=8
                Total time spent by all maps in occupied slots (ms)=683050
                Total time spent by all reduces in occupied slots (ms)=69542
                Total time spent by all map tasks (ms)=683050
                Total time spent by all reduce tasks (ms)=69542
                Total vcore-milliseconds taken by all map tasks=683050
                Total vcore-milliseconds taken by all reduce tasks=69542
                Total megabyte-milliseconds taken by all map tasks=699443200
                Total megabyte-milliseconds taken by all reduce tasks=71211008
        Map-Reduce Framework
                Map input records=26070134
                Map output records=26070132
                Map output bytes=442789828
                Map output materialized bytes=494930141
                Input split bytes=1677
                Combine input records=0
                Combine output records=0
```



```
                Total megabyte-milliseconds taken by all reduc
        Map-Reduce Framework
                Map input records=26070134
                Map output records=26070132
                Map output bytes=442789828
                Map output materialized bytes=494930141
                Input split bytes=1677
                Combine input records=0
                Combine output records=0
                Reduce input groups=45843
                Reduce shuffle bytes=494930141
                Reduce input records=26070132
                Reduce output records=45843
                Spilled Records=76775523
                Shuffled Maps =7
                Failed Shuffles=0
                Merged Map outputs=7
                GC time elapsed (ms)=4438
                CPU time spent (ms)=102820
                Physical memory (bytes) snapshot=1817300992
                Virtual memory (bytes) snapshot=16446136320
                Total committed heap usage (bytes)=1384775680
        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=1669001
You have new mail in /var/spool/mail/acadgild
```



```
[acadgild@localhost ~]$ hadoop fs -ls /CaseStudyIOut
18/05/24 23:59:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
re applicable
Found 2 items
-rw-r--r--   1 acadgild supergroup          0 2018-05-24 23:44 /CaseStudyIOut/_SUCCESS
-rw-r--r--   1 acadgild supergroup    1669001 2018-05-24 23:43 /CaseStudyIOut/part-r-00000
[acadgild@localhost ~]$ 
```



```
[acadgild@localhost ~]$ hadoop fs -cat /CaseStudyIOut/part-r-00000
18/05/25 00:01:02 WARN util.NativeCodeLoader: Unable to load native-hadoop
re applicable
Toy Story (1995)        66008   3.888157
GoldenEye (1995)        32534   3.431841
City Hall (1996)        4436    3.232304
Curdled (1996)  217     3.099078
"Comic  1       4.000000
Up in Smoke (1957)      3       3.666667
First Daughter (1999)   3       3.333333
"Flaw   14      3.714286
Battle of Los Angeles (2011)    44      2.522727
Jason Becker: Not Dead Yet (2012)       9       3.444444
Chicago Massacre: Richard Speck (2007)  2       2.500000
Keep the Lights On (2012)       25      3.100000
Beauty Is Embarrassing (2012)   15      3.600000
Girl Model (2011)       32      3.281250
Crossfire Hurricane (2012)      18      3.388889
Middle of Nowhere (2012)        11      3.454545
True Blue (2001)        3       3.000000
"Guns of Fort Petticoat 3       3.333333
Human Planet (2011)     197     4.271574
Madagascar (2011)       26      3.769231
Omar Killed Me (Omar m'a tuer) (2011)   9       3.166667
Enola Gay and the Atomic Bombing of Japan (1995)        1       3.500000
Red Hook Summer (2012)  11      2.045455
Stella Maris (1918)     2       3.750000
Die (2010)      10      2.550000
Patrice O'Neal: Elephant in the Room (2011)     22      3.204545
Sunny (Sseo-ni) (2011)  26      3.576923
My Way (Mai Wei) (2011) 30      3.716667
```