BIG DATA HADOOP & SPARK TRAINING

Case study I for session 7

Rashmi Krishna

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?

****************************RATING MAPPER***

```
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();
                  }

          }
}
```

Explanation:
This code is for mapping the rating:

- Here we are checking the input received from input and files and bifurcating them accordingly
- Input values are LongWritable and text formats while outputs are in Text formats.
- We are taking only *UserID & rating* from this file.
- We are checking if key and values are null, then return. If not split the inputs by **","** and parts[1] in the parts array is UserID and parts[2] is movierating.
- This UserID i.e. **Key** and rating i.e. **Value** is sent as output to the reducer from this mapper.

****

**********************************MOVIE MAPPER*************************************

```
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();
    }
            }
}
```

Explanation:
This code is for mapping the rating:

- Here we are checking the input received from input and files and bifurcating them accordingly
- Input values are LongWritable and text formats while outputs are in Text formats.
- We are taking only *movieID & moviename* from this file.
- We are checking if key and values are null, then return. If not split the inputs by **","** and parts[0] in the parts array is movieID and parts[1] is moviename.
- This movieID i.e. **Key** and moviename i.e. **Value** is sent as output to the reducer from this mapper.

**************************************REDUCER**

```
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(",");
                                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));
                }
        }
```

Explanation:

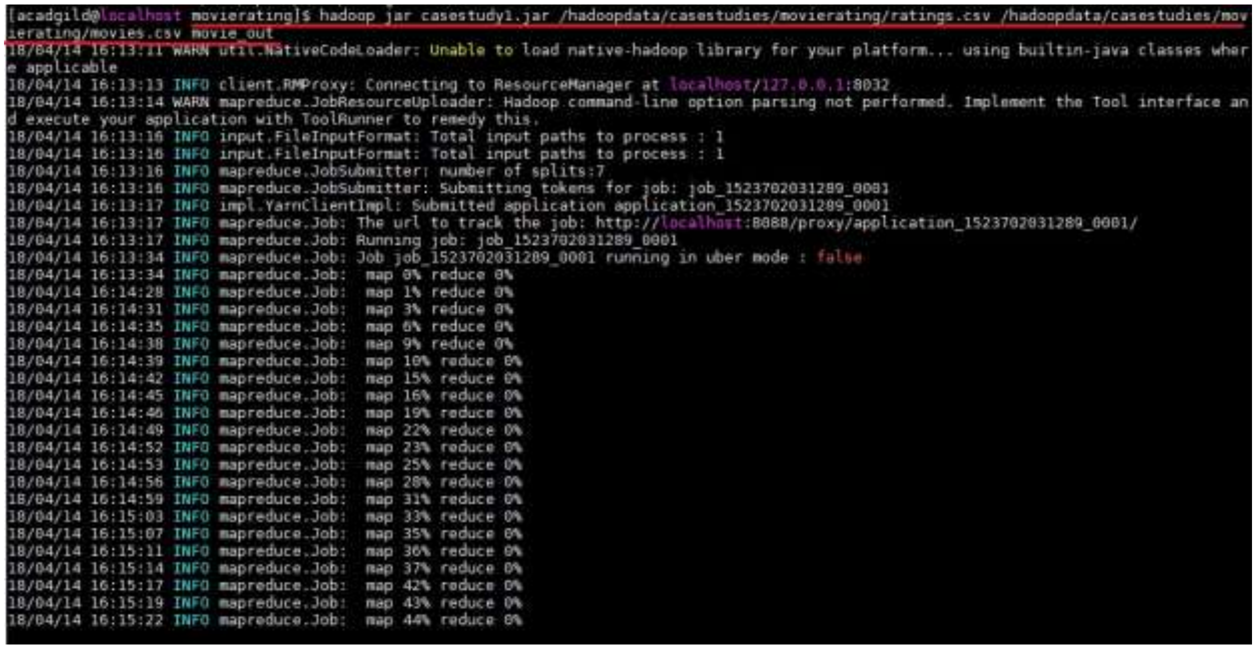
- Here outputs of two mappers are inputs to this reducer.
- Both input and outputs are Text format.
- Now we check all the inputs and bifurcate them accordingly.
- UserID and MovieID are the keys, we split the input by "," and check if the part is "*rating*" or not.
    - If the part is rating then we print the rating and calculate the total.
    - If the part is not rating then it must moviename, then we pring the moviename and save it in the variable "title"
    - We calculate the average of the rating for a particular movie title.
- We print the number of times the movie was rating by the user and the average rating.

****

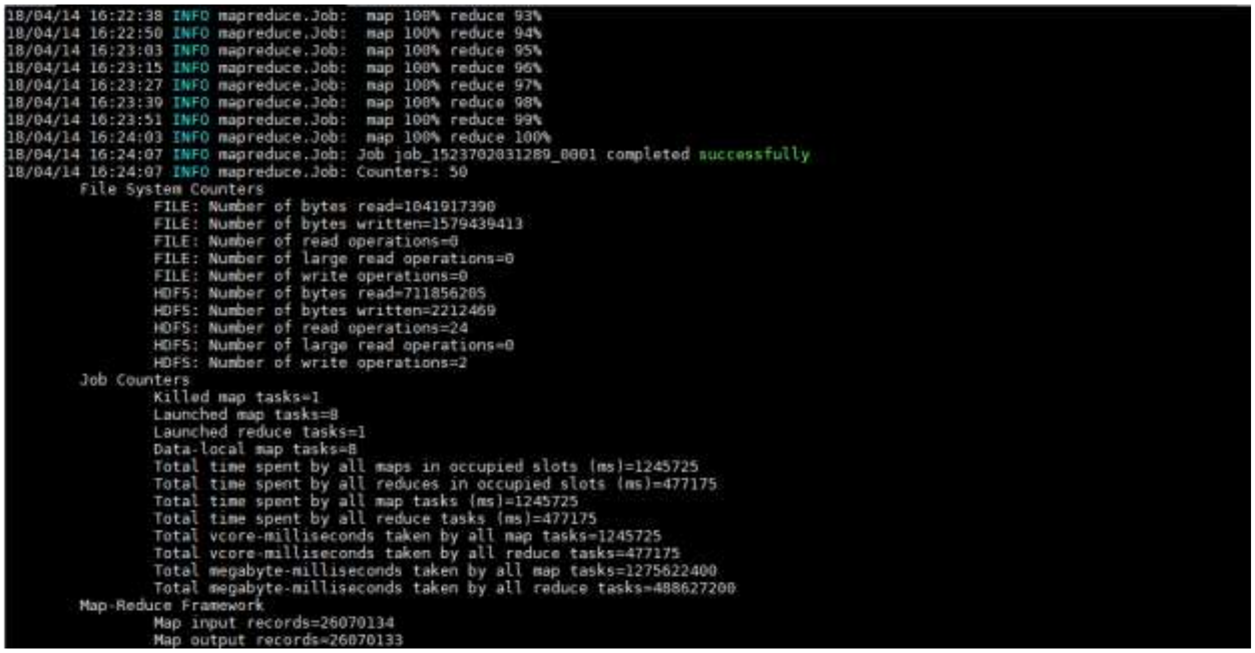
```
******************************DRIVER *******************************************

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);
 }
}
```

Explanation:

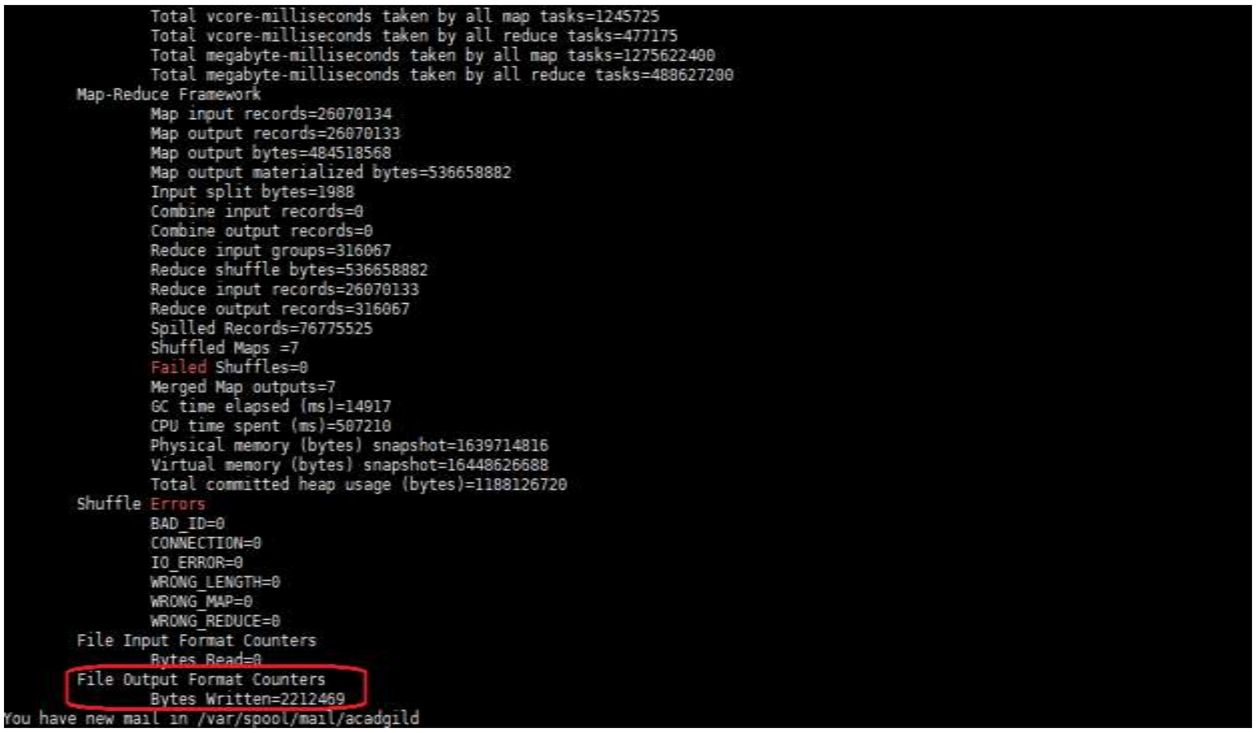
- Here there are 2 input paths and 1 output path, thereby, we check if all the 3 parameters are entered by the user, if not an error is given saying user has to enter 3 parameters and exits.
- Job configuration instance is created and driverclass is set jar by class.
- Multiple input path are defined under args[0] and args[1], as we have two csv files. So each csv file is given in two different paths
- Output path is defined and also output key and value class



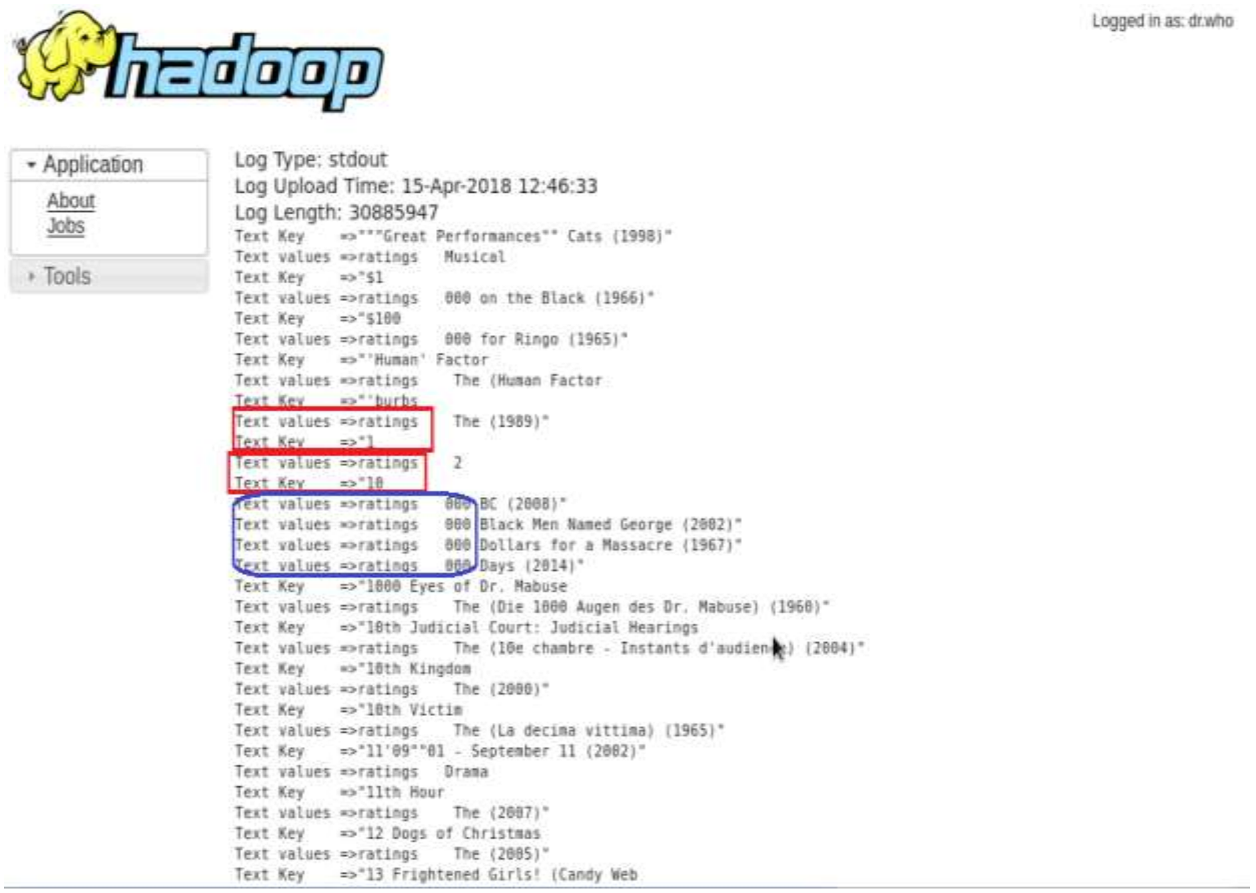
```
[acadgild@localhost movierating]$ hadoop jar casestudy1.jar /hadoopdata/casestudies/movierating/ratings.csv /hadoopdata/casestudies/mov
ierating/movies.csv movie_out
18/04/14 16:13:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes wher
e applicable
18/04/14 16:13:13 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:8032
18/04/14 16:13:14 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface an
d execute your application with ToolRunner to remedy this.
18/04/14 16:13:16 INFO input.FileInputFormat: Total input paths to process : 1
18/04/14 16:13:16 INFO input.FileInputFormat: Total input paths to process : 1
18/04/14 16:13:16 INFO mapreduce.JobSubmitter: number of splits:7
18/04/14 16:13:16 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1523702031289_0001
18/04/14 16:13:17 INFO impl.YarnClientImpl: Submitted application application_1523702031289_0001
18/04/14 16:13:17 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1523702031289_0001/
18/04/14 16:13:17 INFO mapreduce.Job: Running job: job_1523702031289_0001
18/04/14 16:13:34 INFO mapreduce.Job: Job job_1523702031289_0001 running in uber mode : false
18/04/14 16:13:34 INFO mapreduce.Job:  map 0% reduce 0%
18/04/14 16:14:28 INFO mapreduce.Job:  map 1% reduce 0%
18/04/14 16:14:31 INFO mapreduce.Job:  map 3% reduce 0%
18/04/14 16:14:35 INFO mapreduce.Job:  map 6% reduce 0%
18/04/14 16:14:38 INFO mapreduce.Job:  map 9% reduce 0%
18/04/14 16:14:39 INFO mapreduce.Job:  map 10% reduce 0%
18/04/14 16:14:42 INFO mapreduce.Job:  map 15% reduce 0%
18/04/14 16:14:45 INFO mapreduce.Job:  map 16% reduce 0%
18/04/14 16:14:46 INFO mapreduce.Job:  map 19% reduce 0%
18/04/14 16:14:49 INFO mapreduce.Job:  map 22% reduce 0%
18/04/14 16:14:52 INFO mapreduce.Job:  map 23% reduce 0%
18/04/14 16:14:53 INFO mapreduce.Job:  map 25% reduce 0%
18/04/14 16:14:56 INFO mapreduce.Job:  map 28% reduce 0%
18/04/14 16:14:59 INFO mapreduce.Job:  map 31% reduce 0%
18/04/14 16:15:03 INFO mapreduce.Job:  map 33% reduce 0%
18/04/14 16:15:07 INFO mapreduce.Job:  map 35% reduce 0%
18/04/14 16:15:11 INFO mapreduce.Job:  map 36% reduce 0%
18/04/14 16:15:14 INFO mapreduce.Job:  map 37% reduce 0%
18/04/14 16:15:17 INFO mapreduce.Job:  map 42% reduce 0%
18/04/14 16:15:19 INFO mapreduce.Job:  map 43% reduce 0%
18/04/14 16:15:22 INFO mapreduce.Job:  map 44% reduce 0%
```



```
18/04/14 16:22:38 INFO mapreduce.Job:  map 100% reduce 93%
18/04/14 16:22:50 INFO mapreduce.Job:  map 100% reduce 94%
18/04/14 16:23:03 INFO mapreduce.Job:  map 100% reduce 95%
18/04/14 16:23:15 INFO mapreduce.Job:  map 100% reduce 96%
18/04/14 16:23:27 INFO mapreduce.Job:  map 100% reduce 97%
18/04/14 16:23:39 INFO mapreduce.Job:  map 100% reduce 98%
18/04/14 16:23:51 INFO mapreduce.Job:  map 100% reduce 99%
18/04/14 16:24:03 INFO mapreduce.Job:  map 100% reduce 100%
18/04/14 16:24:07 INFO mapreduce.Job: Job job_1523702031289_0001 completed successfully
18/04/14 16:24:07 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=1041917390
                FILE: Number of bytes written=1579439413
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=711856205
                HDFS: Number of bytes written=2212469
                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)=1245725
                Total time spent by all reduces in occupied slots (ms)=477175
                Total time spent by all map tasks (ms)=1245725
                Total time spent by all reduce tasks (ms)=477175
                Total vcore-milliseconds taken by all map tasks=1245725
                Total vcore-milliseconds taken by all reduce tasks=477175
                Total megabyte-milliseconds taken by all map tasks=1275622400
                Total megabyte-milliseconds taken by all reduce tasks=488627200
        Map-Reduce Framework
                Map input records=26070134
                Map output records=26070133
```



```
		Total vcore-milliseconds taken by all map tasks=1245725
		Total vcore-milliseconds taken by all reduce tasks=477175
		Total megabyte-milliseconds taken by all map tasks=1275622400
		Total megabyte-milliseconds taken by all reduce tasks=488627200
	Map-Reduce Framework
		Map input records=26070134
		Map output records=26070133
		Map output bytes=484518568
		Map output materialized bytes=536658882
		Input split bytes=1988
		Combine input records=0
		Combine output records=0
		Reduce input groups=316067
		Reduce shuffle bytes=536658882
		Reduce input records=26070133
		Reduce output records=316067
		Spilled Records=76775525
		Shuffled Maps =7
		Failed Shuffles=0
		Merged Map outputs=7
		GC time elapsed (ms)=14917
		CPU time spent (ms)=507210
		Physical memory (bytes) snapshot=1639714816
		Virtual memory (bytes) snapshot=16448626688
		Total committed heap usage (bytes)=1188126720
	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=2212469
You have new mail in /var/spool/mail/acadgild
```



Log Type: stdout

Log Upload Time: 15-Apr-2018 12:46:33

Log Length: 30885947

```
Text Key     =>"""Great Performances"" Cats (1998)"
Text values =>ratings   Musical
Text Key     =>"$1
Text values =>ratings   000 on the Black (1966)"
Text Key     =>"$100
Text values =>ratings   000 for Ringo (1965)"
Text Key     =>"'Human' Factor
Text values =>ratings    The (Human Factor
Text Key     =>"'burbs
Text values =>ratings    The (1989)"
Text Key     =>"1
Text values =>ratings    2
Text Key     =>"10
Text values =>ratings   000 BC (2008)"
Text values =>ratings   000 Black Men Named George (2002)"
Text values =>ratings   000 Dollars for a Massacre (1967)"
Text values =>ratings   000 Days (2014)"
Text Key     =>"1000 Eyes of Dr. Mabuse
Text values =>ratings    The (Die 1000 Augen des Dr. Mabuse) (1960)"
Text Key     =>"10th Judicial Court: Judicial Hearings
Text values =>ratings    The (10e chambre - Instants d'audience) (2004)"
Text Key     =>"10th Kingdom
Text values =>ratings    The (2000)"
Text Key     =>"10th Victim
Text values =>ratings    The (La decima vittima) (1965)"
Text Key     =>"11'09""01 - September 11 (2002)"
Text values =>ratings   Drama
Text Key     =>"11th Hour
Text values =>ratings    The (2007)"
Text Key     =>"12 Dogs of Christmas
Text values =>ratings    The (2005)"
Text Key     =>"13 Frightened Girls! (Candy Web
```