Assignment 5 Solution

STUDENT NAME:

1) Mapper class source code.

```
package kochinting.hw5;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileSplit;
import org.apache.hadoop.mapreduce.Counter;
import org.apache.log4j.Logger;

/**
 * Created by hdadmin on 10/22/16.
 */
public class MovieRatingsMapper extends Mapper<LongWritable, Text,
CompositeKeyWritable, Text>{

    Logger logger = Logger.getLogger(MovieRatingsMapper.class);
    //IntWritable one = new IntWritable(1);
    Text txtValue = new Text("");
    int intSrcIndex=0;
    StringBuilder strMapValueBuilder = new StringBuilder("");
    List<Integer> lstRequiredAttribList = new ArrayList<Integer>();
    CompositeKeyWritable cKey = new CompositeKeyWritable();

    @Override
    protected void setup(Context context) throws IOException, InterruptedException {
        super.setup(context);
        logger.info("in setup of " + context.getTaskAttemptID().toString());
        String fileName = ((FileSplit) context.getInputSplit()).getPath() + "";
        System.out.println ("in stdout"+ context.getTaskAttemptID().toString() + " " +
fileName);
        System.err.println ("in stderr"+ context.getTaskAttemptID().toString());

        FileSplit fsFileSplit = (FileSplit) context.getInputSplit();
    if (fsFileSplit.getPath().getName().equals("u1.data")
            || fsFileSplit.getPath().getName().equals("u2.data")
            || fsFileSplit.getPath().getName().equals("u3.data")
            || fsFileSplit.getPath().getName().equals("u4.data")
            || fsFileSplit.getPath().getName().equals("u5.data")){
        intSrcIndex =1;
    }
    if (fsFileSplit.getPath().getName().equals("u.item") ){
        intSrcIndex =2;
    }
    // Initialize the list of fields to emit as output based on
    // intSrcIndex (1: u.data, 2=u.item)
    if (intSrcIndex == 1 ) // movie-rating
    {
        //lstRequiredAttribList.add(1); // movie id
        lstRequiredAttribList.add(2); // movie rating
        //lstRequiredAttribList.add(3); // movie rating

    } else if (intSrcIndex == 2 ) // movie information
    {
        lstRequiredAttribList.add(1); // Title
        lstRequiredAttribList.add(2); // release date
        lstRequiredAttribList.add(4); // URL
    }
}

private String buildMapValue(String arrEntityAttributesList[]) {

    strMapValueBuilder.setLength(0);// Initialize
    // Build list of attributes to output based on source
    for (int i = 1; i < arrEntityAttributesList.length; i++) {
        // If the field is in the list of required output
        // append to stringbuilder
        if (lstRequiredAttribList.contains(i)) {
            strMapValueBuilder.append(arrEntityAttributesList[i]).append("\t");
        }
    }
    if (strMapValueBuilder.length() > 0) {
        // Drop last comma
        strMapValueBuilder.setLength(strMapValueBuilder.length() - 1);
        }
        return strMapValueBuilder.toString();
    }

    @Override
    protected void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {
        if (value.toString().length() > 0 && intSrcIndex==1) {
            String arrEntityAttributes[] = value.toString().split("\t");
            cKey.setjoinKey(arrEntityAttributes[1]);
            cKey.setsourceIndex(intSrcIndex);
            txtValue.set(buildMapValue(arrEntityAttributes));
            context.write(cKey, txtValue);
            Counter MapperCounter = context.getCounter(MyCounter.TOTAL_RECORDS);
            MapperCounter.increment(1);
        }
        if (value.toString().length() > 0 && intSrcIndex==2) {
            String arrEntityAttributes[] = value.toString().split("\\|");
            cKey.setjoinKey(arrEntityAttributes[0]);
            cKey.setsourceIndex(intSrcIndex);
            txtValue.set(buildMapValue(arrEntityAttributes));
            context.write(cKey, txtValue);
        }
    }
}
```

2)Reducer class source code

```
package kochinting.hw5;

import java.io.IOException;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.Counter;
/**
 * Created by hdadmin on 10/22/16.
 */

public class MovieRatingsReducer extends Reducer<CompositeKeyWritable, Text, Text,
Text> {

    Text movieID = new Text("");
    Text tab = new Text("\t");
    StringBuilder reduceValueBuilder = new StringBuilder("");
    Text reduceOutputValue = new Text("");
    String strSeparator = "\t";

    int count = 0;
    int ratingSUM = 0;
    float sum = 0;

    @Override
    protected void reduce(CompositeKeyWritable ckey, Iterable<Text> values, Context
context) throws IOException, InterruptedException {
        movieID.set(ckey.getjoinKey());
        for (Text value : values) {

            if (ckey.getsourceIndex() == 2) {
                reduceValueBuilder.append(value.toString()).append(strSeparator);
                reduceOutputValue.set(reduceValueBuilder.toString());
                context.write(movieID, reduceOutputValue);

                Counter ReducerCounter =
context.getCounter(MyCounter.NUMBER_MOVIES);
                ReducerCounter.increment(1);

            } else if (ckey.getsourceIndex() == 1) {
                // movie rating data
                ratingSUM = Integer.parseInt(value.toString());
        sum += ratingSUM;
        count++;
      }
    }
    if (count != 0) {
      reduceValueBuilder.append("Average Rating: " + Math.round(sum /
count)).append(strSeparator);
    }
    reduceValueBuilder.append("#of users: " + count).append(strSeparator).append("#of
ratings: " + count).append(strSeparator);
    //reduceValueBuilder.append("SUM: "+sum).append(strSeparator);

    if (reduceValueBuilder.length() > 1 && ckey.getsourceIndex()==1) {
      //reduceValueBuilder.setLength(reduceValueBuilder.length() - 1);
      // Emit output
      reduceOutputValue.set(reduceValueBuilder.toString());
      //context.write(tab, reduceOutputValue);
      count = 0;
      sum = 0;
    } else {
      {
        System.out.println("Key=" + ckey.getjoinKey() + "src="
            + ckey.getsourceIndex());
      }
      // Reset variables
      reduceValueBuilder.setLength(0);
      reduceOutputValue.set("");
    }
  }
}
```

3)Combiner class source code.

N/A

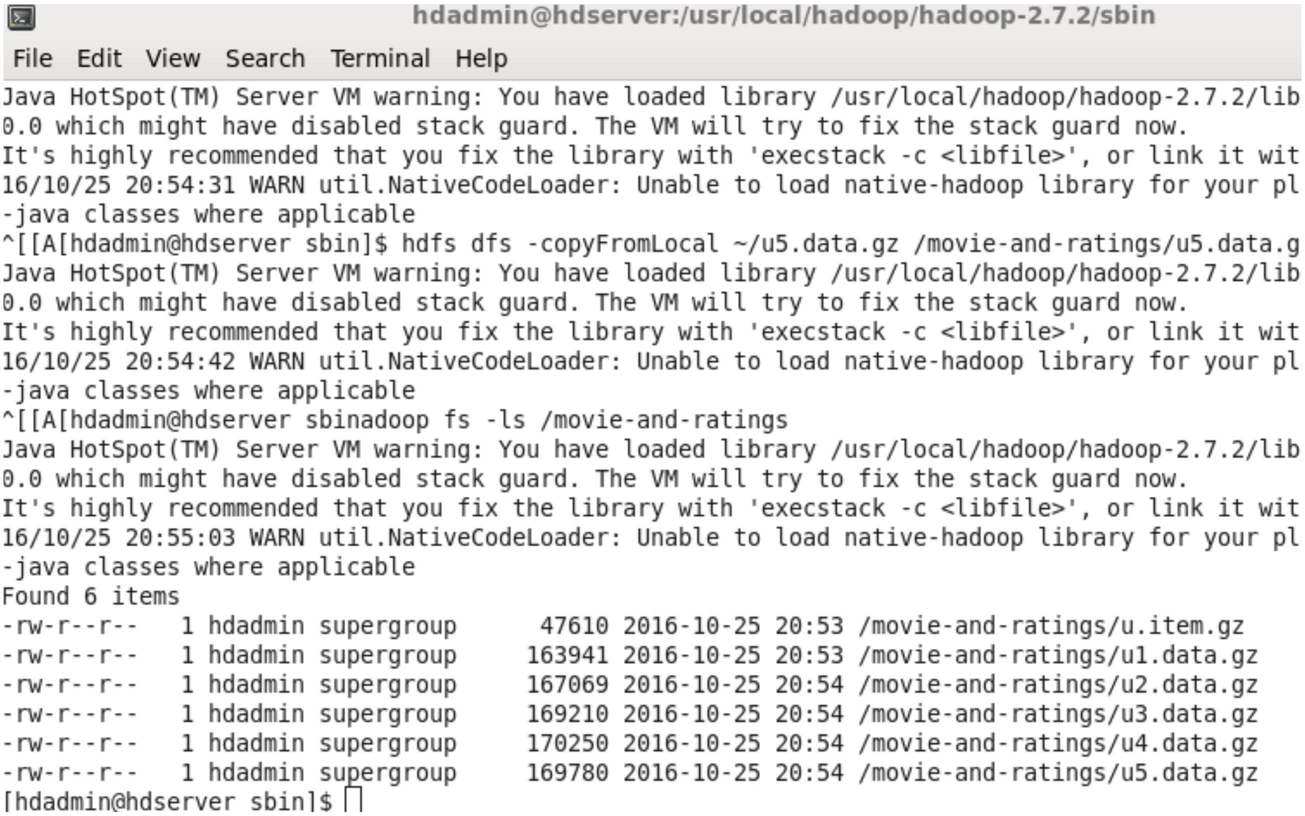
4)Main class source where you submit the jar.

```
package kochinting.hw5;

import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class Main {

    public static void main(String[] args) {
        // write your code here
        Job movieRatings = null;
        Configuration conf = new Configuration();

        conf.setBoolean("mapreduce.output.fileoutputformat.compress", true);
        conf.setStrings("mapreduce.output.fileoutputformat.compress.codec",
"org.apache.hadoop.io.compress.GzipCodec");

        //conf.setInt("mapred.reduce.tasks", 1);
        System.out.println ("======");
        try {
            movieRatings = new Job(conf, "MovieRatings");
        } catch (IOException ex) {
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
            return;
        }
        // Specify the Input path
        try {
            FileInputFormat.addInputPath(movieRatings, new Path(args[0]));
        } catch (IOException ex) {
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
            return;
        }
        // Set the Input Data Format
        movieRatings.setInputFormatClass(TextInputFormat.class);
        // Set the Mapper and Reducer Class
        movieRatings.setMapperClass(MovieRatingsMapper.class);
        movieRatings.setMapOutputKeyClass(CompositeKeyWritable.class);
        movieRatings.setMapOutputValueClass(Text.class);

        //movieRatings.setCombinerClass(MovieRatingsCombiner.class);
        movieRatings.setReducerClass(MovieRatingsReducer.class);
        movieRatings.setOutputKeyClass(Text.class);
        movieRatings.setOutputValueClass(Text.class);
        // Set the Jar file
        movieRatings.setJarByClass(Main.class);

        // Set the Output path
        FileOutputFormat.setOutputPath(movieRatings, new Path(args[1]));

        // Set the Output Data Format
        movieRatings.setOutputFormatClass(TextOutputFormat.class);

        // Set the Output Key and Value Class
        movieRatings.setOutputKeyClass(Text.class);
        movieRatings.setOutputValueClass(IntWritable.class);

        movieRatings.setNumReduceTasks(2);

        try {
            movieRatings.waitForCompletion(true);
        } catch (IOException ex) {
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
        } catch (InterruptedException ex) {
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

5) Any other vital class that your job depends upon

```
******************************************************************************
//Class:    CompositeKeyWritable
//Purpose:  Custom Writable that serves as composite key
//          with attributes joinKey and sourceIndex
//Reference:   http://hadooped.blogspot.com/2013/09/reduce-side-joins-in-java-map-
reduce.html
//
*******************************************************************************
package kochinting.hw5;

/**
 * Created by hdadmin on 10/22/16.
 */
import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;

import org.apache.hadoop.io.Writable;
import org.apache.hadoop.io.WritableComparable;
import org.apache.hadoop.io.WritableUtils;

public class CompositeKeyWritable implements Writable,
        WritableComparable<CompositeKeyWritable> {

    // Data members
    private String joinKey;// MovieID
    private int sourceIndex;// 1=Movie Rating; 2=Movie information

    public CompositeKeyWritable() {
    }
    public CompositeKeyWritable(String joinKey, int sourceIndex) {
        this.joinKey = joinKey;
        this.sourceIndex = sourceIndex;
    }

    @Override
    public String toString() {

        return (new StringBuilder().append(joinKey).append("\t")
                .append(sourceIndex)).toString();
    }
    public void readFields(DataInput dataInput) throws IOException {
        joinKey = WritableUtils.readString(dataInput);
        sourceIndex = WritableUtils.readVInt(dataInput);
    }

    public void write(DataOutput dataOutput) throws IOException {
        WritableUtils.writeString(dataOutput, joinKey);
        WritableUtils.writeVInt(dataOutput, sourceIndex);
    }

    public int compareTo(CompositeKeyWritable objKeyPair) {

        int result = joinKey.compareTo(objKeyPair.joinKey);
        if (0 == result) {
            result = Double.compare(sourceIndex, objKeyPair.sourceIndex);
        }
        return result;
    }

    public String getjoinKey() {
        return joinKey;
    }

    public void setjoinKey(String joinKey) {
        this.joinKey = joinKey;
    }
    public int getsourceIndex() {
        return sourceIndex;
    }
    public void setsourceIndex(int sourceIndex) {
        this.sourceIndex = sourceIndex;
    }
}
package kochinting.hw5;

/**
 * Created by hdadmin on 10/25/16.
 */
public enum  MyCounter {
    TOTAL_RECORDS,
    NUMBER_MOVIES
}
```

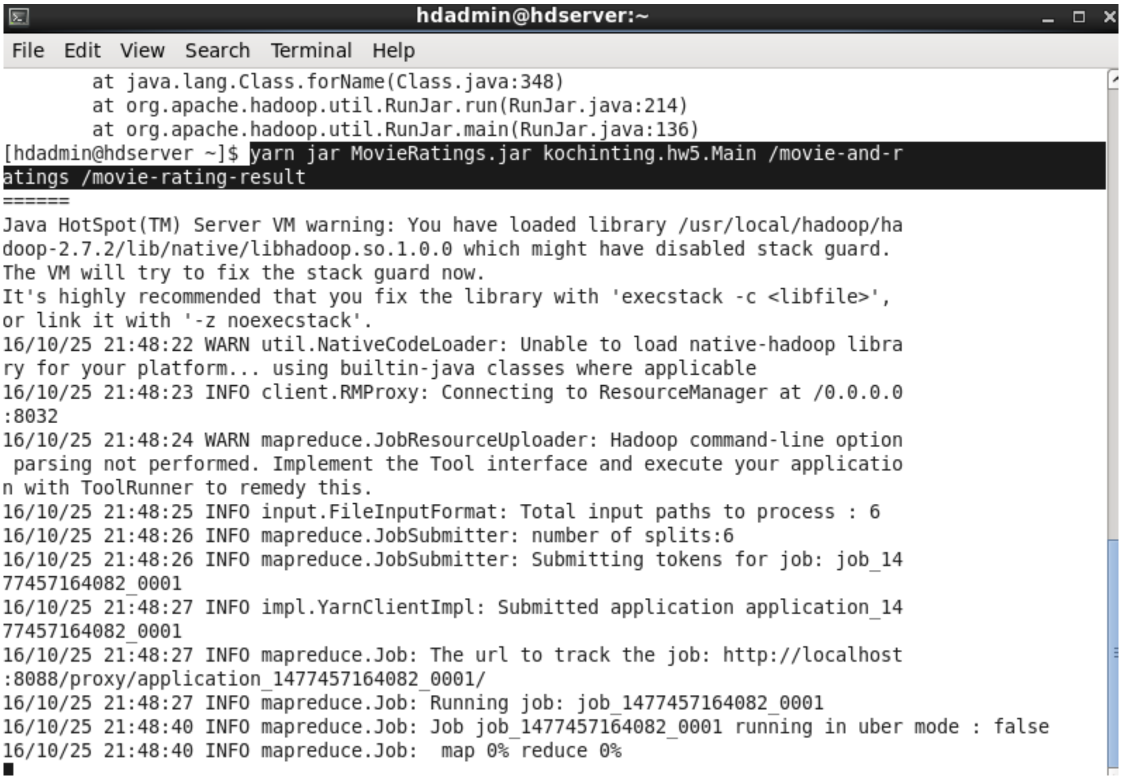
6)List the contents of program input directory in HDFS. That would be /movie-and-ratings .



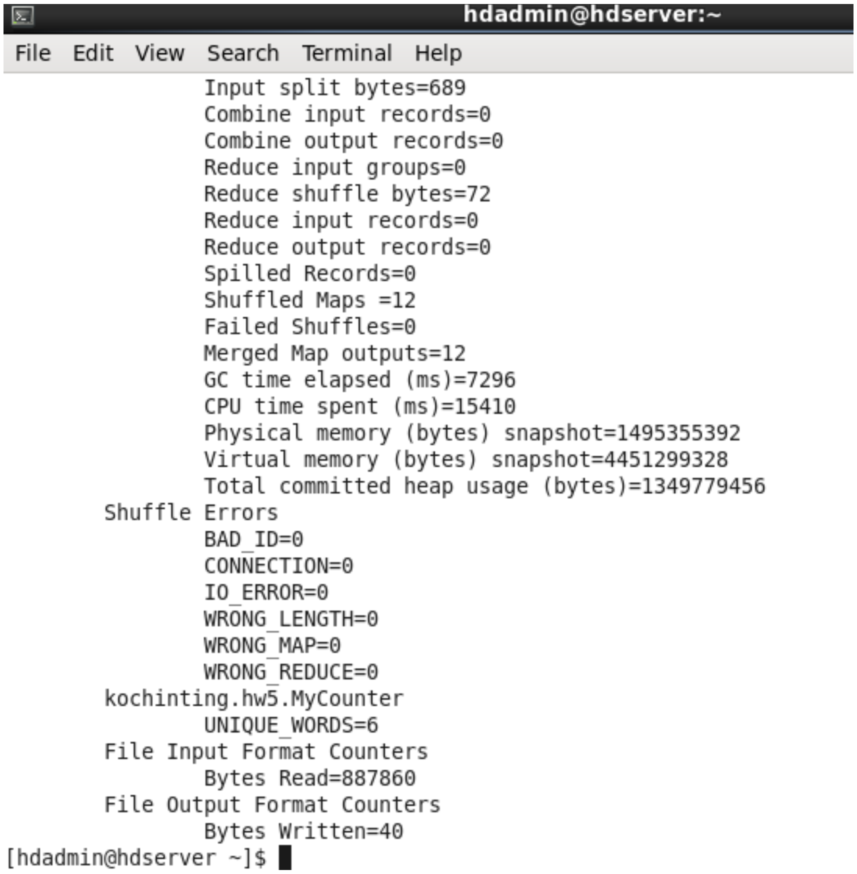
hdadmin@hdserver:/usr/local/hadoop/hadoop-2.7.2/sbin
File Edit View Search Terminal Help

```
Java HotSpot(TM) Server VM warning: You have loaded library /usr/local/hadoop/hadoop-2.7.2/lib
0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it wit
16/10/25 20:54:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pl
-java classes where applicable
^[[A[hdadmin@hdserver sbin]$ hdfs dfs -copyFromLocal ~/u5.data.gz /movie-and-ratings/u5.data.g
Java HotSpot(TM) Server VM warning: You have loaded library /usr/local/hadoop/hadoop-2.7.2/lib
0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it wit
16/10/25 20:54:42 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pl
-java classes where applicable
^[[A[hdadmin@hdserver sbinadoop fs -ls /movie-and-ratings
Java HotSpot(TM) Server VM warning: You have loaded library /usr/local/hadoop/hadoop-2.7.2/lib
0.0 which might have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it wit
16/10/25 20:55:03 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pl
-java classes where applicable
Found 6 items
-rw-r--r--   1 hdadmin supergroup      47610 2016-10-25 20:53 /movie-and-ratings/u.item.gz
-rw-r--r--   1 hdadmin supergroup     163941 2016-10-25 20:53 /movie-and-ratings/u1.data.gz
-rw-r--r--   1 hdadmin supergroup     167069 2016-10-25 20:54 /movie-and-ratings/u2.data.gz
-rw-r--r--   1 hdadmin supergroup     169210 2016-10-25 20:54 /movie-and-ratings/u3.data.gz
-rw-r--r--   1 hdadmin supergroup     170250 2016-10-25 20:54 /movie-and-ratings/u4.data.gz
-rw-r--r--   1 hdadmin supergroup     169780 2016-10-25 20:54 /movie-and-ratings/u5.data.gz
[hdadmin@hdserver sbin]$ 
```

7) Run the program on command line and place the screenshot of successful run right here.

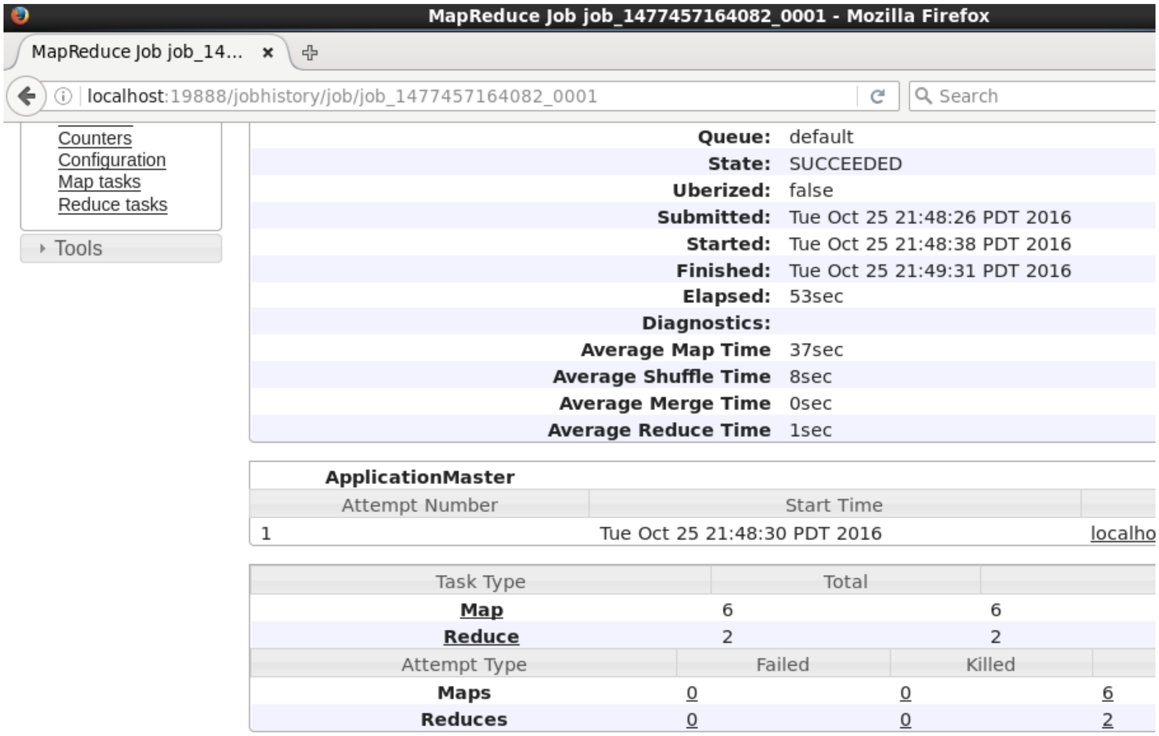


```
        at java.lang.Class.forName(Class.java:348)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:214)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
[hdadmin@hdserver ~]$ yarn jar MovieRatings.jar kochinting.hw5.Main /movie-and-r
atings /movie-rating-result
======
Java HotSpot(TM) Server VM warning: You have loaded library /usr/local/hadoop/ha
doop-2.7.2/lib/native/libhadoop.so.1.0.0 which might have disabled stack guard.
The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>',
or link it with '-z noexecstack'.
16/10/25 21:48:22 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
16/10/25 21:48:23 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
16/10/25 21:48:24 WARN mapreduce.JobResourceUploader: Hadoop command-line option
 parsing not performed. Implement the Tool interface and execute your applicatio
n with ToolRunner to remedy this.
16/10/25 21:48:25 INFO input.FileInputFormat: Total input paths to process : 6
16/10/25 21:48:26 INFO mapreduce.JobSubmitter: number of splits:6
16/10/25 21:48:26 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_14
77457164082_0001
16/10/25 21:48:27 INFO impl.YarnClientImpl: Submitted application application_14
77457164082_0001
16/10/25 21:48:27 INFO mapreduce.Job: The url to track the job: http://localhost
:8088/proxy/application_1477457164082_0001/
16/10/25 21:48:27 INFO mapreduce.Job: Running job: job_1477457164082_0001
16/10/25 21:48:40 INFO mapreduce.Job: Job job_1477457164082_0001 running in uber mode : false
16/10/25 21:48:40 INFO mapreduce.Job:  map 0% reduce 0%
```



```
                Input split bytes=689
                Combine input records=0
                Combine output records=0
                Reduce input groups=0
                Reduce shuffle bytes=72
                Reduce input records=0
                Reduce output records=0
                Spilled Records=0
                Shuffled Maps =12
                Failed Shuffles=0
                Merged Map outputs=12
                GC time elapsed (ms)=7296
                CPU time spent (ms)=15410
                Physical memory (bytes) snapshot=1495355392
                Virtual memory (bytes) snapshot=4451299328
                Total committed heap usage (bytes)=1349779456
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        kochinting.hw5.MyCounter
                UNIQUE_WORDS=6
        File Input Format Counters
                Bytes Read=887860
        File Output Format Counters
                Bytes Written=40
[hdadmin@hdserver ~]$
```

8)A screen shot of the JobHistoryServer when you ran the job. Click on the Job, get into the details, then take a screen shot.



MapReduce Job job_1477457164082_0001 - Mozilla Firefox

MapReduce Job job_14...

localhost:19888/jobhistory/job/job_1477457164082_0001

Queue:	default
State:	SUCCEEDED
Uberized:	false
Submitted:	Tue Oct 25 21:48:26 PDT 2016
Started:	Tue Oct 25 21:48:38 PDT 2016
Finished:	Tue Oct 25 21:49:31 PDT 2016
Elapsed:	53sec
Diagnostics:	
Average Map Time	37sec
Average Shuffle Time	8sec
Average Merge Time	0sec
Average Reduce Time	1sec

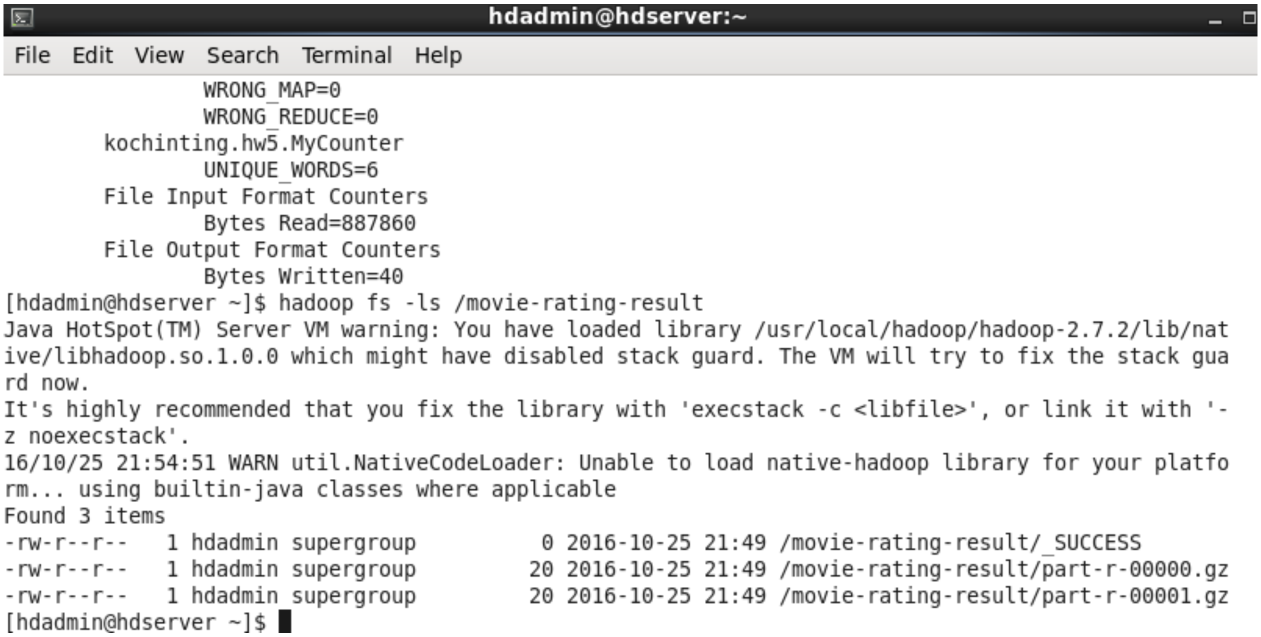
ApplicationMaster

Attempt Number	Start Time	
1	Tue Oct 25 21:48:30 PDT 2016	localho

Task Type	Total	
Map	6	6
Reduce	2	2

Attempt Type	Failed	Killed	
Maps	0	0	6
Reduces	0	0	2

9)List the contents of the job output directory in hdfs. That would be /movie-rating-result)



```
                WRONG_MAP=0
                WRONG_REDUCE=0
        kochinting.hw5.MyCounter
                UNIQUE_WORDS=6
        File Input Format Counters
                Bytes Read=887860
        File Output Format Counters
                Bytes Written=40
[hdadmin@hdserver ~]$ hadoop fs -ls /movie-rating-result
Java HotSpot(TM) Server VM warning: You have loaded library /usr/local/hadoop/hadoop-2.7.2/lib/nat
ive/libhadoop.so.1.0.0 which might have disabled stack guard. The VM will try to fix the stack gua
rd now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-
z noexecstack'.
16/10/25 21:54:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platfo
rm... using builtin-java classes where applicable
Found 3 items
-rw-r--r--   1 hdadmin supergroup          0 2016-10-25 21:49 /movie-rating-result/_SUCCESS
-rw-r--r--   1 hdadmin supergroup         20 2016-10-25 21:49 /movie-rating-result/part-r-00000.gz
-rw-r--r--   1 hdadmin supergroup         20 2016-10-25 21:49 /movie-rating-result/part-r-00001.gz
[hdadmin@hdserver ~]$ 
```

10) For the movie id 376 and 495, what is the movie title, release date, IMDB URL, average rating, total number of unique users rated, total number of ratings for the movie. Please so please fill the data grid below:

Movie id	Title	Release date	IMDB URL	Average rating	Total number of unique users rated	Total number of ratings for the movie
376	Houseguest	01-Jan-1994	http://us.imdb.com/M/title-exact?Houseguest%20(1994)	3	20	20
495	Around the world in 80 days	01-Jan-1956	http://us.imdb.com/M/title-exact?Around%20the%20World%20in%2080%20Days%20(1956)	4	49	49