**Youtude Data analysis using MapReduce**

**PROJECT REPORT**

**ITCS 6010 Big Data Cloud Deployment**

**Spring 2017**

**GROUP 5**

**Rucha Kshirsagar**

**Spoorthy Ravi**

**Table of Contents**

Abstract2

Hadoop Installation2

Dataset description2

Top 5 categories with Maximum Videos uploaded3

Source Code3

How to execute3

Screenshots3

Top 10 videos with Maximum Ratings4

Source Code4

How to execute4

Screenshots4

Top 10 videos with Maximum number of Views5

Source Code5

How to execute5

Screenshots5

Visual representations of the Results6

Conclusion7

Team Efforts 7

References8

# **Abstract**

****YouTube data is rapidly increasing and with such great speed there comes a demand to store and process these datasets. Hence, the idea behind this project is to convert the large amount of YouTube data into some valuable insights. We have implemented MapReduce for our Analysis. The intuition behind choosing MapReduce is its flexibility, high scalability, computing speed, parallel processing, availability and resilient nature. Also, the cost savings are massive and costs can reduce from thousands and figures to hundred figures for every terabyte of data.

# **Hadoop Installation**

Install VirtualBox and then Download the Cloudera quickstart vm for VirtualBox.

# **Data Set Description**

**Column 1:** Video id of 11 characters.

**Column 2:** Who has uploaded of the video  
**Column 3:** The interval between the date of YouTube establishment and the date the when video was uploaded.  
**Column 4:** Category of the video.  
**Column 5:** Duration of the video.

**Column 6:** Number of views for the video.  
**Column 7:** Rating on the video.  
**Column 8:** Number of ratings given for the video

**Column 9:** Number of comments on the videos.  
**Column 10:** Related video ids with the uploaded video.

# **ANALYSIS**

# **1: Top 5 categories with maximum number of videos uploaded.**

**Source Code:**

It contains the Mapper and Reducer Java Code.

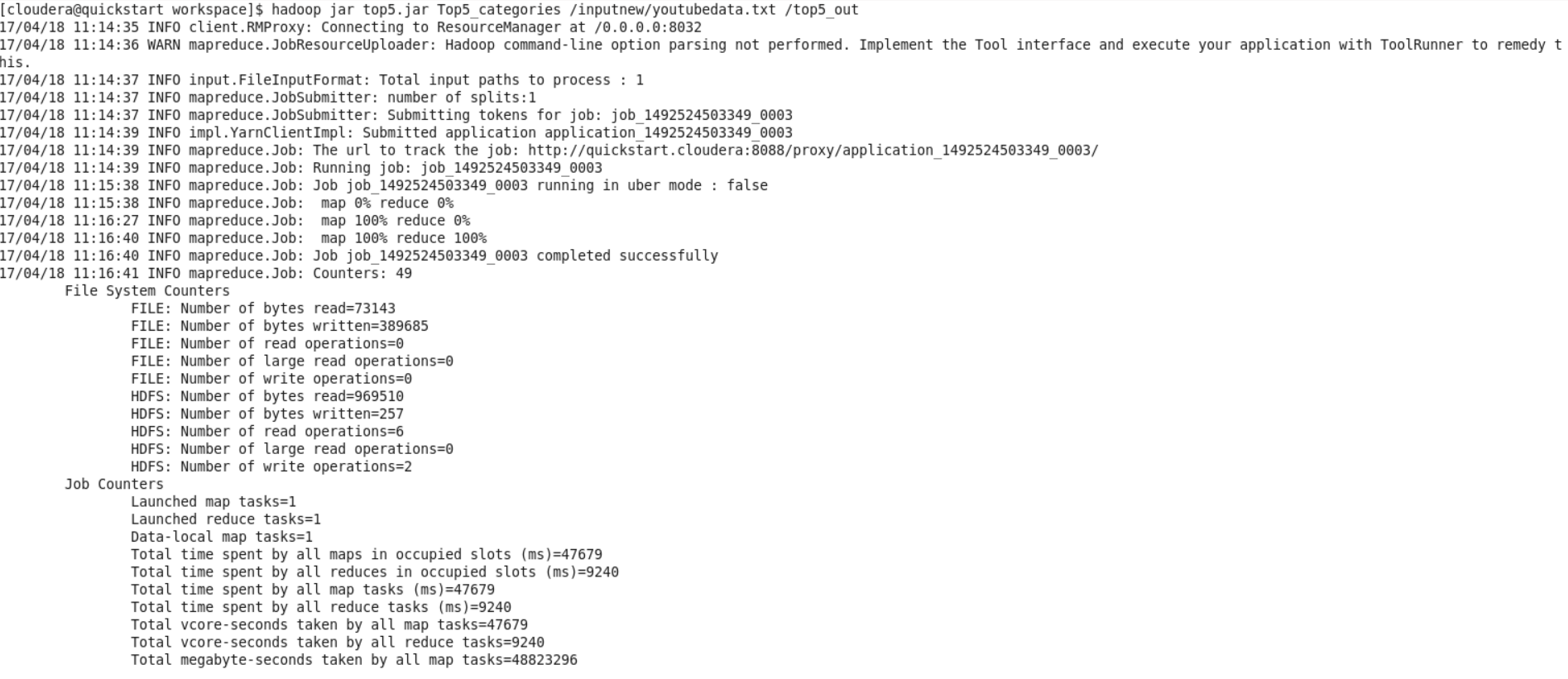
**How to execute:**

**$ hadoop jar top5.jar Top5\_categories /inputnew/youtubedata.txt /top5\_out**

Here *‘hadoop’* specifies we are running a Hadoop command and *jar* specifies which type of application we are running and*top5.jar* is the jar file which we have created consisting of the above source code.

The path of the Input file in our case is root directory of hdfs denoted by*/youtubedata.txt* and the output file location to store the output has been given as *top5\_out.*

**Cloudera VM terminal:**  
**Screenshots:**





**Output**

**Command to view the output**

**hadoop fs -cat /top5\_out/part-r-00000 | sort –n –k2 –r | head  –n5**

Here *‘hadoop’* stipulates that we are running a Hadoop command and *dfs* stipulates that we are performing an operation related to Hadoop Distributed File System and *‘- cat’* is used to view the contents of a file and *top5\_out/part-r-00000* is the file where output is stored.

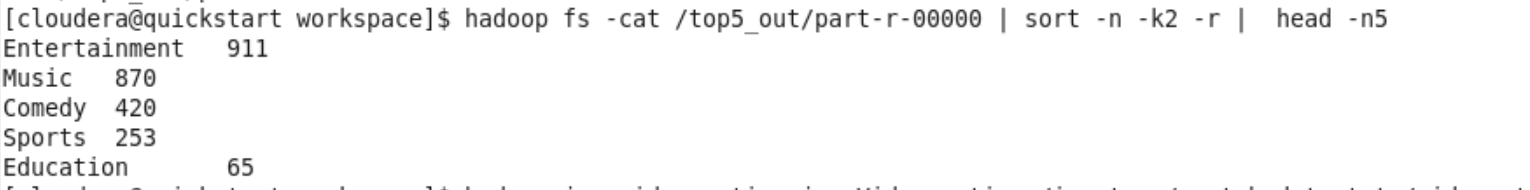
Part file holding the actual output is created by default by the *TextInputFormat* class of Hadoop.

Here ***sort –n –k2 –r | head –n5*** brings you the top 5 categories with maximum number of videos uploaded.

Rather than writing a*secondary sort* after reducer we can basically use this command to get the required output.

*Sort* will sort the data,*–n* means sorting numerically, *–k2* means second column, –*r* is for recursive operation and *head –n5* means to bring the first 5 values after sorting.

**Screenshot of the output:**



**2: Top 10 rated videos**

**Source Code:**

It contains the Mapper and Reducer Java Code.

**How to execute**

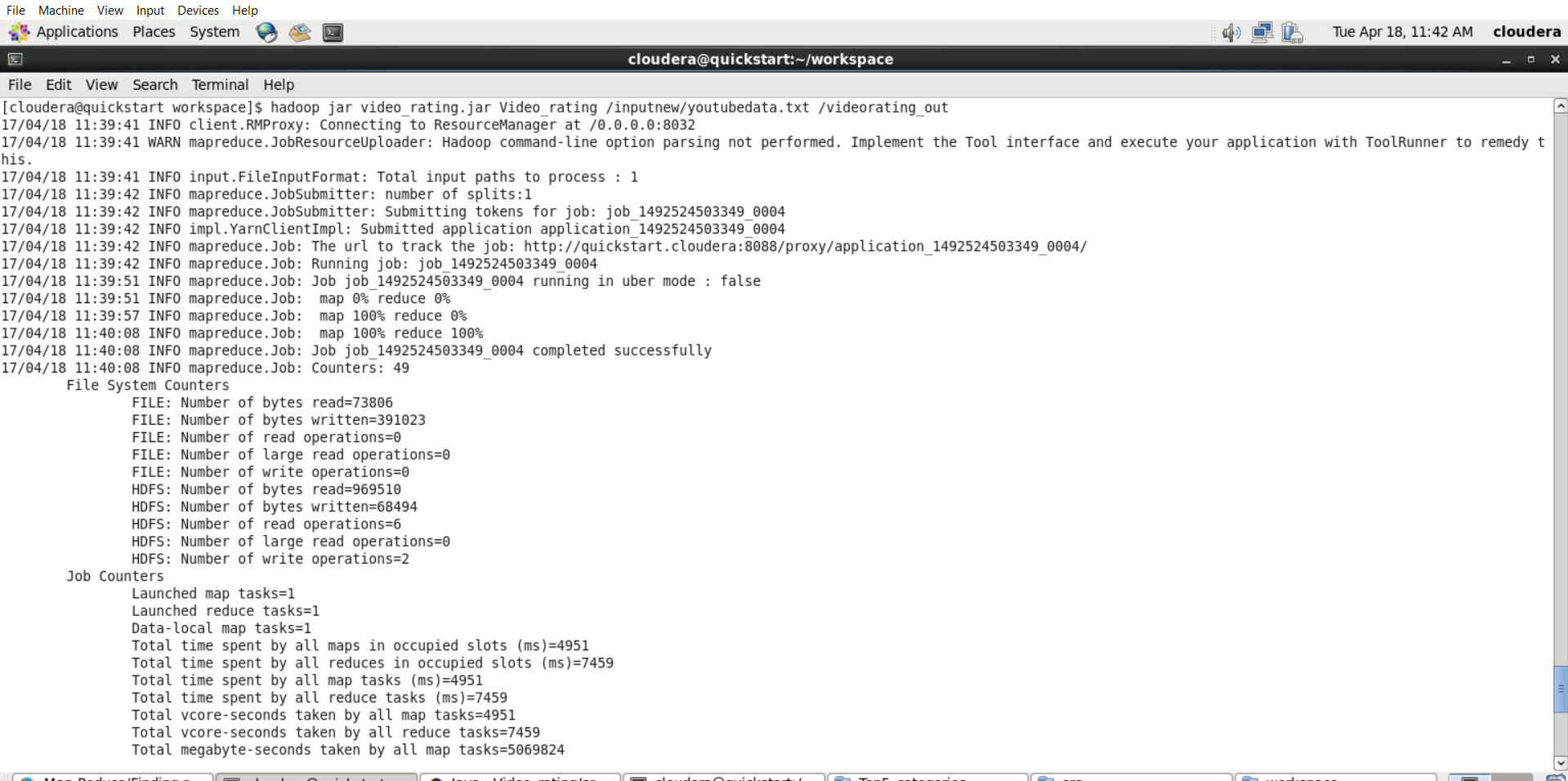
$ **hadoop jar video\_rating.jar Video\_rating/inputnew/youtubedata.txt /videorating\_out**

Here *‘hadoop’* stipulates we are running a Hadoop command and *jar* stipulates which type of application we are running and*video\_rating.jar* is the jar file which we have created consisting of the above source code.

The path of the Input file here is the root directory of hdfs denoted by*/youtubedata.txt* and the output file location to store the output has been given as *videorating\_out.*

**Cloudera VM terminal**

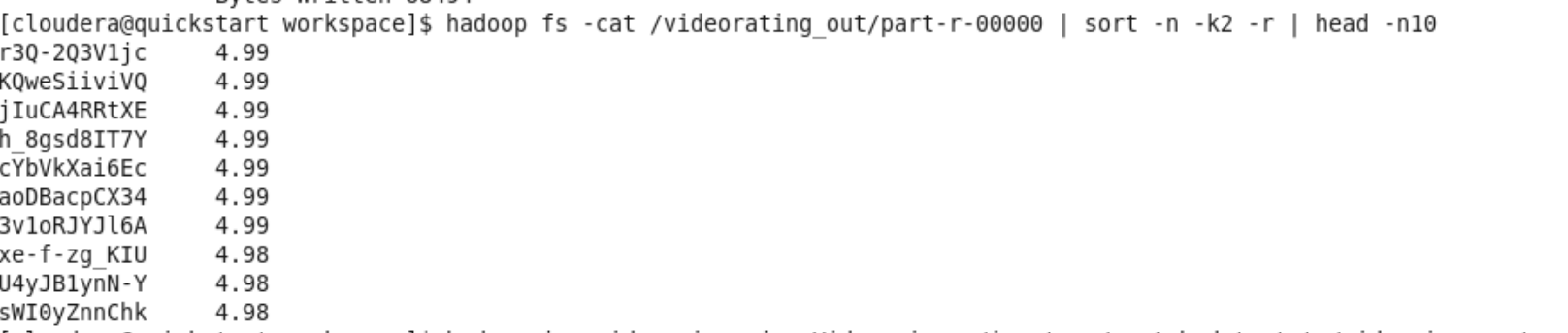
**Screenshots**





**Command to view output**

**hadoop fs -cat /videorating\_out/part-r-00000 | sort –n –k2 –r | head –n10**



# **3: Top 10 videos with Maximum number of views**



**Source Code:**

It contains the Mapper and Reducer Java Code.

**How to execute**

**hadoop jar video\_views.jar Video\_views /inputnew/youtubedata.txt /videoviews\_out**

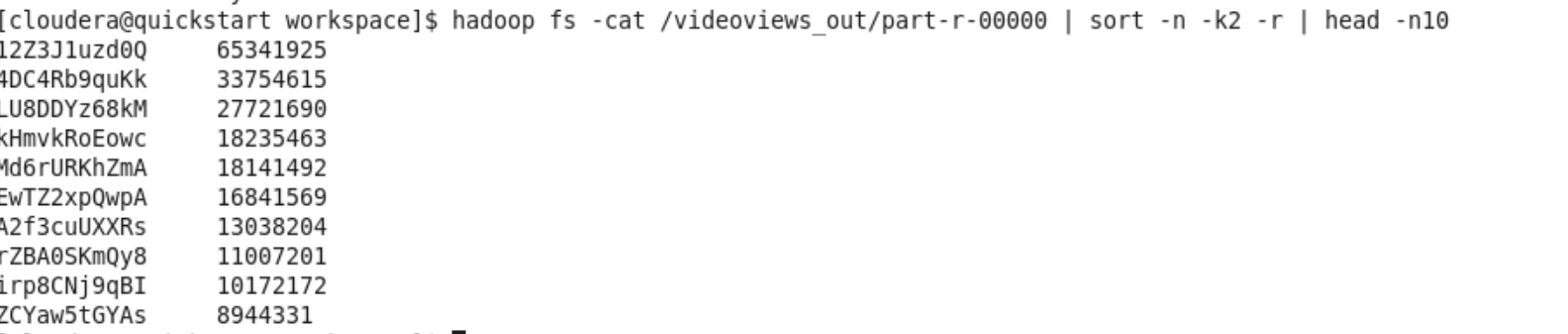
Here *‘hadoop’* specifies we are running a Hadoop command and *jar* specifies which type of application we are running and*video\_views.jar* is the jar file which we have created consisting of the above source code.

The path of the Input file in our case is root directory of hdfs denoted by*/youtubedata.txt* and the output file location to store the output has been given as *videoviews\_out.*

**Command to view the output**

**hadoop fs -cat /videoviews\_out/part-r-00000 | sort –n –k2 –r | head –n10**

**Output Screenshot**



**Visual Representation of the Results**

**Conclusion**

Thus we have analyzed the YouTube Big Data and come up with results that could not be determined otherwise.

This type of analysis can be used by multi-national companies who have strong YouTube footprint as YouTube has a very large viewer and subscriber base across the globe.

The Future work for the project relies on the Real time data extraction from YouTube Data API which permits applications or websites to incorporate functions that are used by YouTube to fetch and retrieve its information. It would include extending the analysis of YouTube data using other Big Data analysis Technologies like Pig and Spark and do a feature comparison analysis. It would be interesting to see which technology fares better as compared to the other technologies. Also, a sentiment Analysis can be conducted to determine the attitude of the people id positive, Negative or neutral towards a subset of videos.

**Team Efforts**

Rucha Kshirsagar

* Developed the JAVA code for Top 5 categories with maximum number of videos uploaded, Top 10 rated videos.
* Implemented the execution of the commands on Hadoop VM and performed Analysis.
* Created visual representation of the results

Spoorthy Ravi

* Installed Hadoop VM on the system.
* Developed JAVA code for top 10 videos with Maximum number of views
* Successfully implemented the commands to execute and view the output

**References**

* Statista.com. 2016. Statistics and facts about YouTube. https://www.statista.com/

[Online] August 2016. <https://www.statista.com/topics/2019/>

* Youtube.com. 2016. YouTube for media.

https://www.YouTube.com/yt/press/statistics.html. [Online] March 2016.

<https://www.YouTube.com/yt/press/statistics.html>

* Datanami.com. 2016. Mining for YouTube Gold with Hadoop and Friends

https://www.datanami.com [Online] July 2016.

https://www.datanami.com/2014/11/12/mining-YouTube-gold-hadoop-friends/