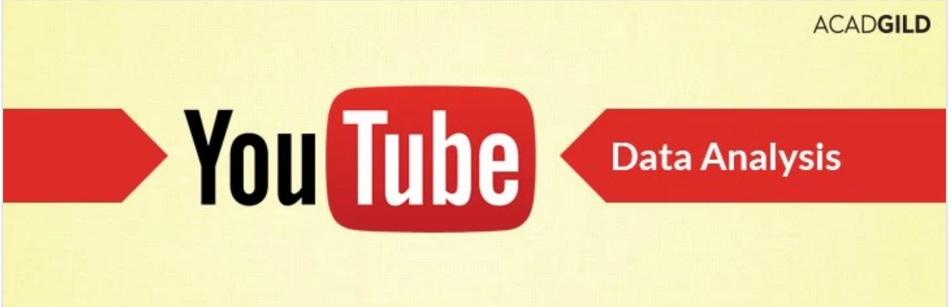
ACADGILD

Learn. Do. Earn.

CATEGORIES ▾

□ Home / Spark / Spark Use Case – Youtube Data Analysis



05
APRIL
2016

Spark Use Case – Youtube Data Analysis

This post is about analyzing the data of YouTube. This total analysis is performed using Apache Spark. This YouTube data is publicly available and the data set is described below under the heading Data Set Description.
Using that dataset, we will perform some analysis and will draw out some insights, like what are the top 10 rated videos in YouTube and who uploaded the most number of videos.

Before getting into the Use Case let's have a brief understanding of Spark with our Beginner's Guide

This post will help you to understand how to handle data sets that does not have proper structure and how to sort the output of reducer.

Data Set Description

Column 1: Video id of 11 characters.

Column 2: Uploader of the video.

Column 3: Interval between day of establishment of YouTube and the date of uploading of the video.

Column 4: Category of the video.

Column 5: Length 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.

You can download the data set from here

Problem Statement 1:

Here, we will find out what are the top five categories with maximum number of videos uploaded.

Source Code:

```
val textFile = sc.textFile("hdfs://localhost:9000/youtubedata.txt")
val counts = textFile.map(line=>{var YoutubeRecord = ""; val temp=line.split("\t"); ;if(temp.length >= 3) {YoutubeRecord=temp(3)};YoutubeRecord})
val test=counts.map ( x => (x,1) )
val res=test.reduceByKey(_+_).map(item => item.swap).sortByKey(false).take(5)
```

Walk Through of the Above Program:

SEARCH

fl Search Now

CATEGORIES

- AcadGild
- Android
- Android For Kids
- AngularJS
- Big Data and Hadoop
- Careers
- Cloud computing
- Database
- Digital Marketing
- Front End
- Full Stack
- Hadoop Administration
- IOS
- Java
- Kids
- Linux Administration
- NodeJS
- Others
- Python
- Quiz

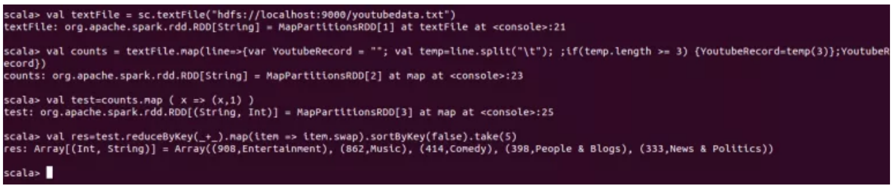
- In **line 1,** we are creating an RDD with the existing dataset, which is inside HDFS.
- In **line 2,** we are taking each record as input using the map method and extracting the 4th column, which is the category of the video.
- In **line 3,** we are creating a pair of category_name,1(count) which is used to calculate how many times that the category is present.

In **line 4,** we are using the reduceByKey method so that all the values of that key are aggregated. Then we are swapping the category_name and its count, and sorting the result with this we will get the sorted records of category_name and its count in descending order. Finally, we are taking the top five from the list.

Output:

(908,Entertainment), (862,Music), (414,Comedy), (398,People & Blogs), (333,News & Politics)

You can this result in the below screenshot.



```
scala> val textFile = sc.textFile("hdfs://localhost:9000/youtubedata.txt")
textFile: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at textFile at <console>:21

scala> val counts = textFile.map(line=>{var YoutubeRecord = ""; val temp=line.split("\t"); ;if(temp.length >= 3) {YoutubeRecord=temp(3)};YoutubeR
ecord})
counts: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[2] at map at <console>:23

scala> val test=counts.map ( x => (x,1) )
test: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[3] at map at <console>:25

scala> val res=test.reduceByKey(_+_).map(item => item.swap).sortByKey(false).take(5)
res: Array[(Int, String)] = Array((908,Entertainment), (862,Music), (414,Comedy), (398,People & Blogs), (333,News & Politics))

scala> 
```

Problem Statement 2:

In this problem statement, we will find the top 10 rated videos in YouTube.

```
val textFile = sc.textFile("hdfs://localhost:9000/youtubedata.txt")
val counts = textFile.filter { x => {if(x.toString().split("\t").length >= 6) true else false} }.map(line=>{line.toString().split("\t")})
val pairs = counts.map(x => {(x(0),x(6).toDouble)})
val res=pairs.reduceByKey(_+_).map(item => item.swap).sortByKey(false).take(10)
```

Walk Through of the Above Program:

- R & Machine Learning
- Scala
- Spark
- Uncategorized

GET SOCIAL

AcadGild
Local Business · Bangal
92,286 likes

Like Page

3 friends like this

AcadGild shared a li
1 hr

Spark Use Case – U
In this post, we will be perforn
ACADGILD.COM

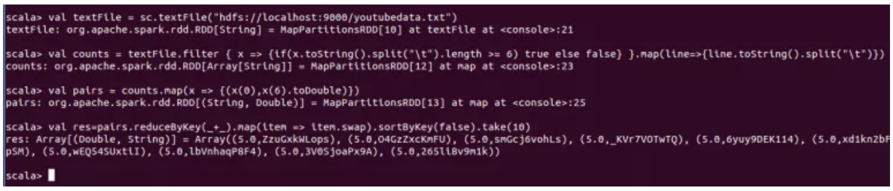
WHAT'S TRENDING

- In **line 1,** we are creating an RDD with the existing dataset, which is inside HDFS.
- In **line2,** we are first filtering the lines with more than six elements to avoid ***ArrayIndexOutOfBounds Exception*** and then we are using **map** method to pass the splitted line as output to the next RDD.
- In **line 3,** we are creating a pair of key and value by using the video id, which is the first column and 7th column 7 respectively.
- In **line 4,** we are using the reduceByKey method to find the ratings of the video, and to sort them by value, we are using the map method and swapping the key and value. Now, values become the keys and we are performing the ***sortByKey*** method and sorting the videos based on their rating and taking the top 10 videos.

Output:

(5.0,ZzuGxkWLops), (5.0,O4GzZxcKmFU), (5.0,smGcj6vohLs), (5.0,_KVr7VOTwTQ), (5.0,6yuy9DEK114), (5.0,xd1kn2bFpSM), (5.0,wEQ54SUxtiI), (5.0,lbVnhaqP8F4), (5.0,3V0SjoaPx9A), (5.0,265li8v9m1k)

This output can be seen in the below screenshot.



```
scala> val textFile = sc.textFile("hdfs://localhost:9000/youtubedata.txt")
textFile: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[10] at textFile at <console>:21

scala> val counts = textFile.filter { x => {if(x.toString().split("\t").length >= 6) true else false} }.map(line=>{line.toString().split("\t")})
counts: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[12] at map at <console>:23

scala> val pairs = counts.map(x => {(x(0),x(6).toDouble)})
pairs: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[13] at map at <console>:25

scala> val res=pairs.reduceByKey(_+_).map(item => item.swap).sortByKey(false).take(10)
res: Array[(Double, String)] = Array((5.0,ZzuGxkWLops), (5.0,O4GzZxcKmFU), (5.0,smGcj6vohLs), (5.0,_KVr7VOTwTQ), (5.0,6yuy9DEK114), (5.0,xd1kn2bF
pSM), (5.0,wEQ54SUxtiI), (5.0,lbVnhaqP8F4), (5.0,3V0SjoaPx9A), (5.0,265li8v9m1k))

scala> 
```

Hope this post has been helpful in understanding how to perform simple data analysis using Spark and Scala.

Keep visiting our website www.acadgild.com for more updates on Big Data and other technologies.

Tweets by @acadgild

ACADGILD
@acadgild

#Spark Use Case - Uber #DataAnalysis buff.ly/1NvpXDN

Spark…
In this …
acadg…

8s

ACADGILD
@acadgild

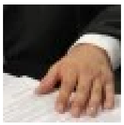
#TechTipTuesday:

Embed View on Twitter

RECENT POSTS

Spark Use Case – Uber Data Analysis
May 16, 2016

Why Learning MongoDB Will Boost Your Career
May 14, 2016



Job Responsibilities of Hadoop Professionals
May 13, 2016

Graphical Exploratory Data Analysis-II
May 13, 2016

Share this:

97 36

Related

MapReduce Use Case-Youtube Data Analysis
December 28, 2015
In "Big Data and Hadoop"

Spark Use Case - Titanic Data Analysis
April 6, 2016
In "Spark"

Spark Use Case - Olympics Data Analysis
April 7, 2016
In "Spark"

KIRAN KRISHNA

Kiran Krishna Innamuri is a Passionate Big Data enthusiast with 2 + years of experience in Hadoop and Spark Development. He is a passionate Java and scala programmer. AcadGild was founded with the vision of "Learn. Do. Earn". We provide skill development courses based on current industry needs. But what sets us apart is earning opportunities we provide after successful completion of course. We also provide live mentoring and 24x7 support. Our mentors are industry thought leaders in their respective fields. We provide courses for Android Programming, Big Data, Front End, Full Stack, AngularJS, NodeJS and Android Programming for children.

PREVIOUS ARTICLE
Testing your Scripts with PigUnit

NEXT ARTICLE
Cloud Computing in Organizations

ARCHIVES

- May 2016
- April 2016
- March 2016
- February 2016
- January 2016
- December 2015
- November 2015
- September 2015
- August 2015
- July 2015
- June 2015
- May 2015
- November 2014
- October 2014
- September 2014
- August 2014

RELATED POSTS

Spark Use Case – Uber Data Analysis

May 16, 2016

Integrating SparkSQL with MySQL

May 12, 2016

Spark Use Case – Travel Data Analysis

May 8, 2016

2 COMMENTS

RAJ □ REPLY TO RAJ

April 8, 2016 at 12:25 pm

Hi,

In the problem statement 1, line# 3, you have used ‘var’ to check the column counts and extract column#3. Instead of using ‘var’ you can write like this,

val counts = textFile.map(line => line.split(“\t”)).filter(columns => columns.length >=3).map(columns => columns(3))

or

val counts = textFile.map(_.split(“\t”)).filter(_.length >= 3).map(_(3))

SATYAM □ REPLY TO SATYAM

April 13, 2016 at 6:32 pm

Hi Raj,

Your approach is also correct, we can use filter also but we have followed a different approach using conditional statements and in our next use case in titanic and olympic we have

extracted the results using the filter approach.You can have a look at https://acadgild.com/blog/spark-use-case-titanic-data-analysis/
Please let us know with your feedbacks for the same.

LEAVE A REPLY

COMMENTS *

NAME * EMAIL *

WEBSITE

SUBMIT

□ NOTIFY ME OF FOLLOW-UP COMMENTS BY EMAIL.

□ NOTIFY ME OF NEW POSTS BY EMAIL.

CATEGORIES

- AcadGild
- Android
- Android For Kids

TAGS

ANDORID PROFILING TOOLS

ANDROID APP FOR SPEECH TO TEXT

LIKE WHAT YOU SEE? SUBSCRIBE TO OUR BLOG

We send only 1 email in a week

Enter your email

- AngularJS
- Big Data and Hadoop
- Careers
- Cloud computing
- Database
- Digital Marketing
- Front End
- Full Stack
- Hadoop Administration
- IOS
- Java
- Kids
- Linux Administration
- NodeJS
- Others
- Python
- Quiz
- R & Machine Learning
- Scala
- Spark
- Uncategorized

ANDROID APP FOR TEXT TO SPEECH

ANDROID DEVELOPMENT

ANDROID MEMORY ANALYZER

ANDROID MEMORY MANAGEMENT

BANGALORE SUMMER CAMP

BEST SUMMER CAMPS 2016

BIG DATA DEVELOPEMENT

COMMISSIONING AND DECOMMISSIONING OF DATANODE IN HADOOP

DEPENDENCY INJECTION

DIFFERENCE BETWEEN ANDROID VS IOS

FEATURES OF DDMS

FILE FORMATS

FILE FORMATS IN HADOOP

HADOOP

HADOOP ADMINISTRATION

HDFS

HIVE WITH MYSQL

INTRODUCTION TO SPARK

JAVASCRIPT MVC

Enter your email...

Subscribe

FRAMEWORK

JOB OPPORTUNITIES IN HADOOP

JOB TRENDS IN BIG DATA BLOG

JOB TRENDS IN HADOOP

LINUX

LINUX BASIC

LINUX BASIC COMMANDS

LINUX COMMANDS

MYSQL

MYSQL-CONNECTOR-JAVA-5.1.2.JAR

MYSQL-CONNECTOR-JAVA.JAR

MYSQL WITH HIVE

MYSQL WITH SQOOP

PYTHON

RACK AWARENESS

RECYCLE BIN

RESILIENT DISTRIBUTED DATASET (RDD)

SPARK

SQOOP

SQOOP WITH MYSQL

STYLING A RESPONSIVE WEB PAGE

SUMMER CAMP

SUMMER CAMP 2016

TOP 10 RECORDS IN MAPREDUCE

TRASH CONFIGURATION