1. ***Abstract***

With the surge of internet companies like Youtube, Yahoo,Amazon, eBay and a rapidly growing internet savvy population, today's advanced systems and enterprises are generating data in a very huge volume with great velocity and in a multi structured formats including videos, images, weblogs etc. from different sources. This has given birth to a new type of data called Big Data.

The big data marketplace is growing big every other day. Open source technologies like Hadoop, Spark finds valuable use cases to top the big data market place.

Analysis of unstructured data has seen tremendous success in the present times using such open source systems.

YouTube a Google company, has over a billion users and generates billions of views. Since YouTube data is getting created in a very huge amount and with an equally great speed, there is a huge demand to store, process and carefully study this large amount of data to make it usable.

A new approach to tackle problems is always needed and this is what Apache Spark is all about! Spark proves to be of great use in analysing the kind of dataset we have for Youtube.

The main objective of this project is to demonstrate how we can analyse this huge data set using Big Data concepts learnt in the class and mainly focuses on how data generated from YouTube can be analysed using Spark framework.

# **2.1 *Introduction***

YouTube, a Google company, has millions of unique visitors every day. These visitors upload or view millions of videos per minute. With that amount of activity comes an equally large amount of data about viewing patterns. Processing this data is of huge importance to YouTube. Done right, analysis of this data could improve visitor engagement, thus leading to an increase in ad revenue for creators and YouTube alike.

The unstructured data that is generated cannot be handled by traditional relational databases like SQL. So to handle such a huge amount of data we can use distributed frameworks like Hadoop or Apache Spark which are built specially for handling unstructured data. This project is about analyzing the Youtube dataset using spark framework.

Just as Google’s search algorithms rank websites to improve search results, YouTube videos can be ranked to improve engagement. In this project we will be analyzing a sample YouTube dataset provided by researchers at Simon Fraser University and was obtained through scraping in the early days of YouTube, 2007 and 2008. This dataset includes quantitative data about each video as well as a list of related videos. Through analysis it should be possible to improve recommendations both at the home screen (highly ranked videos overall) and at the video player screen (highest ranked related video). Further work is done to analyze other aspects of video metadata.

***2.2 Dataset Description:***

YouTube data we used is publicly available. We can download the dataset from the link below.

The dataset includes a series of scrapes of YouTube data from 2007/2008. It was compiled by researchers at Simon Fraser University. It can be found at <http://netsg.cs.sfu.ca/youtubedata/>

Dataset Description :

Column no. 1: Video id.

Column no. 2: Video uploader.

Column no. 3: Interval between the day of establishment of YouTube and the date of uploading of the video.

Column no. 4: Category of the video.

Column no. 5: Length of the video.

Column no. 6: Number of views for the video.

Column no. 7: Rating on the video.

Column no. 8: Number of ratings given for the video

Column no. 9: Number of comments on the videos.

Column no. 10: Related video ids of the uploaded video.

Using this dataset, we have performed following analysis

1. Find out what are the Highest Ranked videos using the PageRank algorithm
2. Find out what are the top video categories with the maximum number of videos uploaded
3. Find out what are the top video categories with the maximum number of views uploaded
4. Sort videos with decreasing order of their ratings.
5. Find the top commented videos on YouTube.
6. Find the top users based on number of videos uploaded.

3. ***Related Work***

Similar research is being performed on various datasets. For instance, the Stock Market dataset. Since Stock Markets generate wide variety of unstructured data, this type of data can be analyzed using Hadoop framework. From the results, stock brokers provided key recommendations including the possibility of stock prices moving in the upward direction or inverse direction

Other works include, the Airline data analysis, in which the project focused on possible delays and provided the output based upon information derived from the analysis

There has been work on the weather data as well, The weather data analysis focused on providing information, which was used to plan any outdoor events.

In all of these cases, improvements to distributed processing from MapReduce, Hadoop, Spark, etc. has advanced the field of data science tremendously.

4. ***Our Approach and DataFlow***

This project is about analyzing the Youtube dataset using spark framework. Steps we followed are described below.

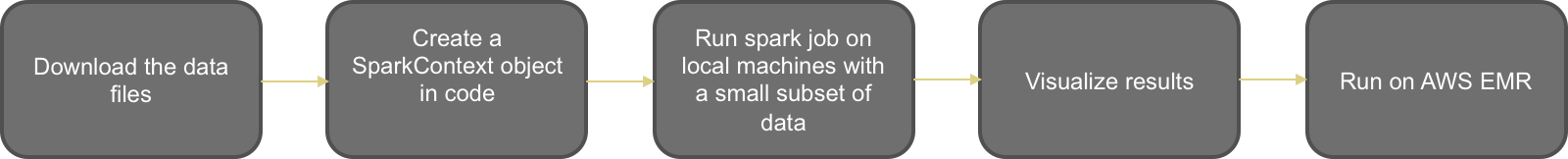


Figure 1: A flow chart outlining the data pipeline for analysis.

4.1 ***Spark Framework Overview***

Apache Spark has its own wonderful advantages which always helped in attracting users. The speed and suitability for handling iterative computations as compared to Hadoop are far better. Iterative computations are especially used for PageRank algorithms. Since finding highest ranked video using pagerank is the key objective of our project, spark serves to be the best fit as our choice of tool

Many Modern world problems related to scientific research are being solved using Spark tools and libraries. It is also a top-level Apache project focused on processing data in parallel across a cluster, but the biggest difference is that it works in-memory.

Apache Spark is written in Scala programming language. To support Python with Spark, Apache Spark community released a tool, PySpark. Using PySpark, we can work with RDDs in Python programming language.

***Design Comparisons:***

To understand the trade off between MapReduce and Spark framework, we performed the comparisons between the two approaches

### ***Hadoop MapReduce***

### To start with, The MapReduce algorithm sits on top of HDFS and consists of Map and a Reduce job. Once an application is written in one of the languages, Hadoop accepts the JobTracker, picks it up, and allocates the work which could include anything from counting words and cleaning log files to TaskTrackers listening on other nodes. YARN allocates resources that the JobTracker spins up and monitors them, moving the processes around for more efficiency. All the results from the MapReduce stage are then aggregated and written back to disk in HDFS.

1. ***Apache Spark***

Spark handles work in a similar way to Hadoop, except that computations are carried out in memory and stored there, until the user actively persists them. Initially, Spark reads from a file on HDFS, S3, or another filestore on local drive, into an established mechanism called the SparkContext. Out of that context, Spark creates a structure called an RDD, or Resilient Distributed Dataset, which represents an immutable collection of elements that can be operated on in parallel.

As the RDD and related actions are being created, Spark also creates a DAG, or Directed Acyclic Graph, to visualize the order of operations and the relationship between the operations in the DAG. Each DAG has stages and steps; in this way, it’s similar to an explain plan in SQL.

You can perform transformations, intermediate steps, actions, or final steps on RDDs. The result of a given transformation goes into the DAG but does not persist to disk, but the result of an action persists all the data in memory to disk.

As stated in the research paper on Spark, it has been found to outperform Hadoop by 10X in iterative jobs and can be used to query the dataset interactively. This encouraged us to choose Spark for our implementation.

***Evaluation :***

We present experimental results demonstrating how we have utilized Spark framework and Pagerank algorithm with respect to ranking the videos and analysing the other trends in Youtube. Specifically, we focus on the following investigation while are analysing the dataset.

1. Accurately identify the highest ranked videos using the PageRank algorithm,
2. Identify the top categories by number of videos uploaded and by number of views,
3. Find the highest rated videos,
4. List the top videos by number of comments
5. Identify the top users by number of videos uploaded
6. ***Experimental setup***

***System overview****:* We have implemented our approach with the help of Apache Spark framework. Apache Spark has its own wonderful advantages which always helped in attracting users. Since finding highest ranked video using pagerank is the key objective of our project, spark serves to be the best fit as our choice of tool as stated in the design comparisons section.

**Workload**: We evaluate our script by running it locally on our machines using a very small subset of data provided.

**Hardware configurations**: Experiments are run on local machines with intel processor of 5 cores running at 2.3 GHz on each core.

**Cluster configurations**: EMR cluster config details

***B. Experimental Results***

We could add some of the screenshots here.

### 

***Conclusion/ Future Work.***

The future work would include extending the feature that is not added in this project to represent the output in a Graphical User Interface (GUI).The current project displays a very simplistic Output which does not warrant a GUI interface.

***Acknowledgement***:

Sowmya would like to thank her family who has supported her to pursue the higher studies and her team member Evan who has been a great guide in building a workable product in this project.